Д. т. н. А. В. ПАЛАГИН, к. т. н. В. Н. ОПАНАСЕНКО, А. Н. ЛИСОВЫЙ

Украина, г. Киев, Институт кибернетики им. В. М. Глушкова
E-mail: vlopanas@ukr.net

Дата поступления в редакцию
09.01 2007 г.

Оппонент *к. т. н. В. Г. БРОВКОВ*
(ОНПУ, г. Одесса)

ПРОЕКТИРОВАНИЕ РЕКОНФИГУРИРУЕМЫХ СИСТЕМ НА ПЛИС

Логико-информационный подход к проектированию ЭВМ модифицируется применительно к реконфигурируемым устройствам на ПЛИС, что позволяет определить оптимальное количество уровней иерархической системы.

Разработка и освоение серийного производства СБИС требует больших затрат как на разработку, так и на оборудование для их изготовления. В схемы, выполненные методом печатного монтажа, изменения вносить довольно сложно, а в схемах, выполненных в виде БИС и СБИС, никакие заранее не предусмотренные изменения не возможны вообще. Это ограничивает не только возможности их специализации для конкретных применений, но и перспективы модернизации, расширения функций, внесения изменений в алгоритм функционирования. Поэтому одним из актуальных требований к современным устройствам вычислительной техники и систем управления является повышение их адаптируемости (гибкости).

Одним из путей реализации этих требований является создание устройств (систем, компонентов) с программируемой структурой.

Сегодня программируемые логические интегральные схемы — **ПЛИС** (Programmable Logic Devices — PLD) прочно вошли в практику современной компьютерной техники, заняв свою нишу прежде всего в качестве средства исследовательского проектирования и проблемной ориентации.

Широко известны работы [1] по организации реконфигурируемого компьютера, представленного двумя основными частями — постоянной (компьютер с жесткой структурой) и переменной (в виде набора вычислительных устройств), которые могут с помощью программы перестраивать свою структуру, а также работы [2] по формализованному проектированию процессоров с гибкой архитектурой. Они способствовали появлению нового направления в вычислительной технике по проектированию реконфигурируемых устройств (**РУ**) с гибкой (программируемой) архитектурой на базе ПЛИС — "Reconfigurable Computing" (что в общем случае обозначает двуединое понятие — как реконфигурируемой структуры компьютера (hardware), так и процесса обработки данных, выполняемого компьютером).

С появлением современных кристаллов ПЛИС [3] типа FPGA с большой логической емкостью (свыше 10 миллионов логических вентилей) стало возможным использовать полученные результаты для построения реконфигурируемых устройств и систем повышенной сложности на базе однородных сред с полностью программируемой архитектурой. При создании таких систем, которые должны обладать высокой надежностью и гибкостью приспособления к структуре решаемых задач, принцип реконфигурируемости структуры является определяющим. Из его реализации автоматически следует, в частности, возможность параллельного выполнения операций, естественно определяемого самой структурой алгоритма и реализуемого с помощью соответствующей настройки автомата. Применяя принцип реконфигурируемости, удается простым перепрограммированием структуры РУ настраивать его на эффективную реализацию заданного алгоритма, сохраняя функциональную универсальность устройства.

Определимся с понятиями. ПЛИС — программируемая логическая интегральная схема, сочетающая регулярность структуры полупроводникового запоминающего устройства с универсальностью микропроцессора, позволяющая программно формировать внутренний специализированный процессор. Конфигурация — определенная совокупность аппаратных средств и соединений между ними, реализующая заданный алгоритм функционирования в течение определенного периода времени. Реконфигурируемость — свойство системы переопределять совокупность аппаратных средств и соединений между ними в соответствии с требуемым алгоритмом функционирования. Файл конфигурации — программный файл, сформированный посредством САПР для конкретного типа кристалла ПЛИС, предназначенный для создания требуемой конфигурации в кристалле ПЛИС.

Постановка задачи

Реконфигурируемая система состоит из постоянной F (или «фиксированной») части (Host-компьютера) и переменной V части — РУ, которые можно объединять в различные конфигурации. Архитектура реконфигурируемых систем зависит от мощности множества алгоритмов (N_F), выполняемых на оборудовании F, и от мощности множества алгоритмов (N_V), выполняемых на оборудовании V. Соотношение этих величин определяет предлагаемую классификацию реконфигурируемых вычислительных систем:

— вычислительные системы, ориентированные на Host-компьютер, в котором сосредоточены основные вычислительные мощности, а реконфигурируемый компьютер обеспечивает повышение производительности только для узкого класса задач ($N_F \to N$, $N_V \to 0$, $N_F >> N_V$);

— реконфигурируемые вычислительные системы, ориентированные на РУ ($N_F \to 0$, $N_V \to N$, $N_V >> N_F$), в которых Host-компьютер используется в основном только для выполнения вспомогательных функций (сервис, ввод-вывод), а все алгоритмы выполняются преимущественно на РУ, которое может иметь собственное поле внешних устройств (через платы расширения) или общее поле внешних устройств с Host-компьютером, к которым РУ имеет непосредственный доступ. РУ является автономным устройством в случае когда $N_F = 0$, $N_V = N$, а Host-компьютер отсутствует;

— вычислительные системы, в которых Host-компьютер и РУ имеют приблизительно одинаковую сложность ($N_F \approx N_V$), при этом РУ ориентировано на решение трудоемких задач, а Host-компьютер обеспечивает мощную поддержку в части трансляции, ввода-вывода, сервиса и т. д.

Для определения оптимального количества уровней программируемых компонентов (в качестве которых будут использоваться ПЛИС) необходимо модифицировать известный *логико-информационный метод проектирования* (**ЛИМ**) [4], логическая концепция которого исходит из многоуровневой организации системы структурного программирования и представления процессора композицией операционного и управляющего автоматов. В соответствии с информационной концепцией процессор рассматривается как информационная система, вся информация в которой отнесена к трем «сферам» состояний: хранения, транспортировки и преобразования. Очевидно, что при определенных соотношениях между объектами информации в этих сферах можно получить оптимальные технические параметры ЭВМ. Оптимальной считается такая структурная реализация модели ЭВМ, для которой в соответствии с принятыми критериями найдены оптимальное количество уровней и оптимальные соотношения между обобщенными характеристиками компонент на каждом уровне, а также между соответствующими характеристиками компонент соседних уровней.

Методика проектирования реконфигурируемой системы

Сущность ЛИМ иллюстрируется схемой

$$\begin{array}{l} \underset{i}{\forall}\left(\exists\Omega : A_i \Rightarrow \Lambda_i \Rightarrow R_i\right) \\ \quad\downarrow \\ \quad A_{i-1} \Rightarrow \Lambda_{i-1} \Rightarrow R_{i-1} \\ \quad\quad \text{M} \\ \quad\quad\downarrow \\ \quad\quad A_0 \Rightarrow \Lambda_0 \Rightarrow R_0(\Theta = \Theta_{\text{extr}}(Q,t)), \end{array} \qquad (1)$$

где A_i, Λ_i, R_i ($i = \overline{1, N}$) — соответственно множества алгоритмов, операторов и их информационно-кодовые представления на i-м уровне программирования;

Θ — совокупность обобщенных характеристик.

Обобщенные характеристики формальной модели (1) легко пересчитываются в технические параметры ПЛИС, а также в информационные характеристики автоматных представлений каждого ($i = \overline{1, N}$) уровня иерархии программируемых автоматов. В [2, с. 135—136] предложена методика энтропийной оценки информационной сложности автомата через вероятности его переходов и состояний. Например, для построения матрицы вероятностей переходов определяется вероятность каждого перехода из состояния a_i в состояние a_j ($i, j = \overline{1, N}$):

$$p(a_j = \delta(a_i, x)) = p_{ij}. \qquad (2)$$

Зная матрицу переходов $\left\| p_{ij} \right\| (i, j = \overline{1, N})$, матрицу вероятности P_j ($j = \overline{1, N}$) пребывания автомата в каждом ($i, j = \overline{1, N}$)-состоянии можно определить в результате решения системы линейных уравнений:

$$P_j = \sum_{i=1}^{N} P_i p_{ij}, \ j = \overline{1, N}, \ \sum_{i=1}^{N} P_i = 1. \qquad (3)$$

Оценка суммарного объема оборудования автоматной композиции принимается пропорциональной сумме энтропийных оценок функций перехода и выхода, каждая из которых подсчитывается на основе известной формулы энтропии, определяющей среднюю длину i-го управляющего слова:

$$H_i = -\sum_{i=1}^{N} P_i \log_2 p_i. \qquad (4)$$

В выражениях (2)—(4) за величину вероятности принимается частота i-го состояния.

Для формализованного представления модели реконфигурируемых устройств выполнена модификация метода ЛИМ, которая иллюстрируется следующей схемой:

$$\begin{array}{l} \underset{i}{\forall}\left(\exists\Omega : A_i \Rightarrow \Lambda_i \Rightarrow R_i\right) \\ \quad\downarrow \\ \quad A_{i-1} \Rightarrow \Lambda_{i-1} \Rightarrow R_{i-1} \\ \quad\quad \text{M} \quad \text{M} \\ \quad\quad\downarrow \quad\downarrow \\ \quad\quad A_0 \Rightarrow \Lambda_0 \Rightarrow R_0(\Theta = \Theta_{\text{extr}}(Q,t)). \end{array} \qquad (5)$$

В соответствии с (1) в данном случае R_i — множество аппаратных реализаций операторов i-го уровня. Причем при синтезе оптимальной структуры устройства некоторые уровни могут быть исключены. В схеме (5) для классических архитектур используются следующие уровни программирования: τ_0 — физический или «нулевой»; τ_1 — микропрограммный; τ_2 — программный; τ_3 — алгоритмический. Програм-

мирование на «нулевом» уровне определяет физическую структуру устройства, которая в конечном итоге реализует заданный алгоритм функционирования, т. е. выполняет программирование структуры устройства. В отличие от схемы (1), предложенная модификация (5) осуществляет не микропрограммную, а аппаратную реализацию алгоритмов на вентильном уровне.

Предложенная модель проектируемой вычислительной системы [5] представлена четверкой:

$S=<M,\ A,\ B,\ D>$,

где M — множество математических методов для предметной области, лежащих в основе функционирования системы;
A — множество алгоритмов реализации метода;
$B=\{b\}$ — алфавит конструктивов, из которых синтезируется структура;
D — процедура описания проекта (описание объекта).

Таким образом, процесс проектирования состоит в решении задачи синтеза структуры на основе конструктивов $\{b\}$ алфавита B для выполнения определенного алгоритма A, реализующего метод M, лежащий в основе функционирования структуры, в соответствии с требованиями спецификаций. Результатом процедуры D является описание проекта во входном языке САПР.

Предложен синтез структурной реализации последовательности алгоритмов, когда метод/задача (M) представляется последовательностью алгоритмов ($A_i, \forall i = \overline{1,\ n}$):

$$M = \bigcup_i A_i.$$

В реконфигурируемых устройствах изначально задана базовая (нулевая) архитектура, реализованная на ПЛИС в виде функционального обрабатывающего поля фиксированной размерности, контроллера шины Host-компьютера, поля памяти, а также хорошо структурированной библиотеки файлов конфигураций (**БФК**) структурных реализаций методов (алгоритмов), выполняющих отображение алгоритма в структурную реализацию ($F{:}A_i \Rightarrow B_i$). Каждый алгоритм имеет отображение $F{:}A_i \Rightarrow B_i$ в структурную реализацию (B_i), которая представляет собой файл конфигурации для кристалла ПЛИС. В общем случае имеется несколько вариантов реализации алгоритма (например, последовательная, последовательно-параллельная и параллельная):

$$B_i = \bigcup_z B_{iz},\ (z = \overline{1,\ k}).$$

Каждый вариант характеризуется параметрами быстродействия (время выполнения — t_{iz}) и аппаратными затратами (q_{iz}). Причем предполагаем, что мощность множества B является достаточной для реализации широкого набора алгоритмов. В том случае, если требуемая реализация i-го алгоритма в библиотеке отсутствует ($B_i=\varnothing$), то необходимо с помощью инструментальных средств САПР ПЛИС создать ее и включить в качестве стандартного элемента в библиотеку. Таким образом, задача оптимизации сводится к упорядоченному назначению каждой i-й вершине графа реализуемого алгоритма (B_{iz})-го элемента библиотеки с целью получения экстремального значения некоторого критерия качества. То есть любой оператор отображается только одним элементом из библиотеки. В результате определяется структура, реализующая заданный граф. Тогда решение задачи может быть получено методами целочисленного математического программирования.

Задача оптимизации состоит в определении минимума целевой функции, а критерием качества являются суммарное время выполнения всех алгоритмов и затраты оборудования:

$$\alpha \sum_i \sum_z t_{iz} x_{iz} + \beta \sum_i \sum_z q_{iz} x_{iz} = \min,$$

$$(\forall i = \overline{1,\ n},\ \forall z = \overline{1,\ k})$$

при ограничениях

$$\sum_{z=1}^{k} x_{iz} = 1,\ \ i = \overline{1,\ n},\ \ z = \overline{1,\ k},$$

$$\sum_i \sum_z q_{iz} x_{iz} \le Q_0,\ \ \sum_i \sum_z t_{iz} x_{iz} \le T_0,$$

где α, β — весовые коэффициенты, которые могут быть определены, например, методом экспертных оценок;
Q_0 — допустимые аппаратные затраты;
T_0 — допустимое время выполнения всех алгоритмов.

Методы решения таких задач достаточно хорошо разработаны и позволяют за допустимое время получать приемлемое решение [6].

Представленные выше модели и подходы положены в основу обобщенного алгоритма проектирования [7] реконфигурируемых устройств на ПЛИС (**рис. 1**), представляющих собой базовую плату (для сопроцессоров, подключаемых к стандартной шине Host-компьютера), несущую плату (для автономных устройств) с набором плат и модулей расширения или же кристалл ПЛИС (для реализации System-on-Chip).

Алгоритм представляет собой последовательность этапов: анализ проблемной области → постановка задачи → выбор подходящего алгоритма из БФК (синтез файла конфигурации для реализации алгоритма в случае его отсутствия с последующей записью в БФК) → отображение на уровне общей архитектуры (функциональная схема) → подготовка формализованного технического задания (**ТЗ**) → программирование структуры на основе файла конфигурации → программирование алгоритма → решение задачи → оценка характеристик параметров (структуры, процесса решения) → проверка параметров на соответствие установленным критериям (при необходимости следующая итерация) → ввод в эксплуатацию. Блок-схема на рис. 1 предусматривает также коррекцию критериев.

Разработанная методика проектирования, опираясь на лежащую в ее основе логико-информационную модель РУ, позволяет решить главную задачу проектирования — формализовать процесс поиска оптимальной пары «алгоритм — структурная реализация». Методика предназначается для проектирования проблемно-ориентированных сопроцессоров и автономных устройств, работающих с заданным набором алгоритмов; реконфигурируемых процессоров

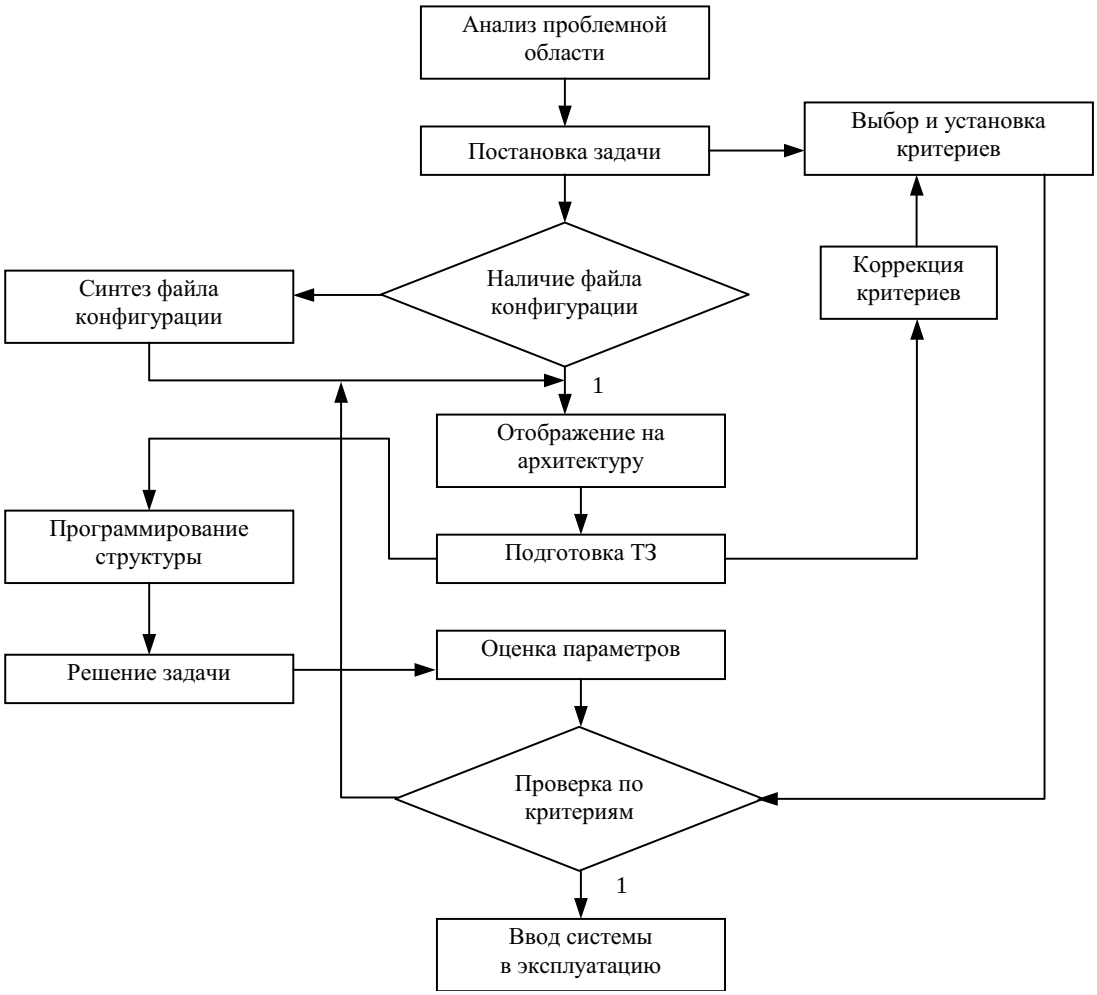

Рис. 1. Блок-схема алгоритма проектирования

с конвейерной обработкой данных; параметрических IP-Core для реализации заданных алгоритмов, которые представляются элементами библиотеки файлов конфигурации; System-on-Chip. Она может видоизменяться и развиваться в зависимости от исходного задания, класса задач, элементно-технологической базы и т. п.

Наиболее трудоемким и наименее формализованным является переход от анализа предметной области к постановке задачи, которая, в свою очередь, предполагает поиск подходящих математических методов ее решения и выбор оптимальных с использованием рассмотренной выше логико-информационной модели.

Проектирование реконфигурируемых проблемно-ориентированных устройств с аппаратной реализацией на базе кристаллов ПЛИС фирмы Xilinx в виде базовых библиотечных функциональных блоков посредством их описания на языке VHDL и схемотехнического редактора обеспечивает их использование широким кругом разработчиков цифровых устройств путем выбора оптимальной структуры (по критериям "быстродействие — сложность реализации").

На всех этапах проектирования выполняется верификация разрабатываемого устройства. Для реконфигурируемых систем необходимо в реальной среде с помощью тест-программ верифицировать проектируемое устройство. Для функционирования и верификации реконфигурируемого устройства в составе системы необходимо разработать драйвер этого устройства [8] и тест-программу для верификации всех компонентов системы.

Разработка тестов для верификации относится к прикладному программированию. В качестве среды для разработки программы используется среда Borland Studio 2006, компилятор языка Delphi. Выбор языка программирования Delphi обусловлен исключительно наличием мощных библиотек для работы с драйвером.

Пример проектирования

В качестве примера рассмотрим построение реконфигурируемой системы в виде программно-аппаратного комплекса на базе современных средств программирования, проектирования и верификации на основе реконфигурируемого устройства для реализации набора библиотечных функциональных блоков с плавающей точкой (соответствующих стандарту IEEE-754): сложения, умножения, деления и преобразования форматов представления чисел с плавающей точкой (**ФПТ**) в формат с фиксированной точкой (**ФФТ**) и обратного преобразования.

Аппаратный комплекс представляет собой реконфигурируемое устройство (плату ADS-XLX-SP3-EVL400 фирмы Avnet), подключаемое к Host-компь-

ютеру через шину PCI (что требует наличия на кристалле контроллера шины PCI). В файл конфигурации (проект) для программирования кристалла ПЛИС должны быть включены, помимо контроллера шины PCI, также функциональные блоки для реализации базовых арифметических операций.

Для верификации аппаратных блоков разработано программное обеспечение, основное окно графического интерфейса которого приведено на **рис. 2**.

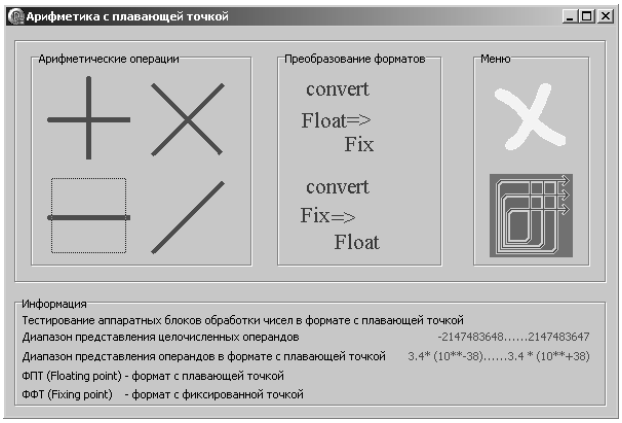

Рис. 2. Главное окно программы

По функциональному назначению операции разделяются на два типа: арифметические операции, операции преобразования форматов. Каждая из кнопок (+, –, /, и т. д.) при нажатии на нее клавиши "мышки" инициирует отображение окна соответствующей операции. Например при нажатии на кнопку «+» отображается окно, графический интерфейс которого для работы с операцией сложения чисел в ФПТ приведен на **рис. 3**.

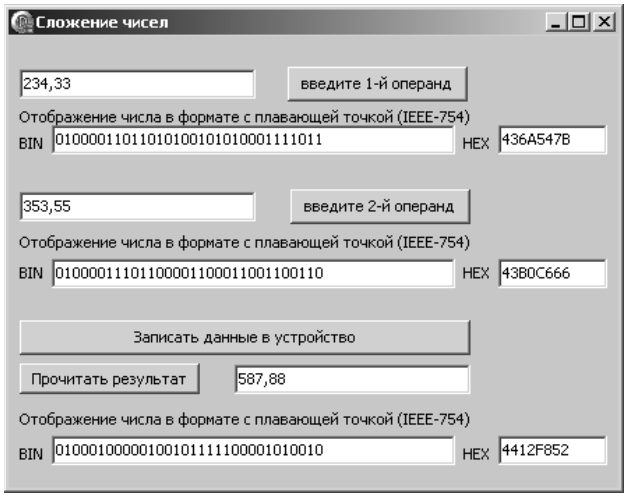

Рис. 3. Графический интерфейс модуля программы для операции сложения чисел

Обмен данными с устройством осуществляется по следующему алгоритму: в окна для ввода текста (параллельно кнопкам «введите 1-й операнд» и «введите 2-й операнд») заносятся операнды, которые входят в диапазон представления чисел стандарта IEEE-754; нажимаются вышеобозначенные кнопки, что приводит к формированию и отображению 32-разрядных векторов (согласно стандарту IEEE-754) в двоичной и шестнадцатеричной форме; нажатие на кнопку «записать данные в устройство» инициирует передачу операндов и код операции сложения по шине в устройство, устройство обрабатывает операнды и записывает в ОЗУ операнд-результат; результат с ОЗУ читается и отображается в соответствующих текстовых окнах в нормальном виде и в формате в двоичном и шестнадцатеричном виде. Обмен данными с устройством для всех других операций происходит аналогично. На **рис. 4** отображен графический интерфейс окна для операции преобразования числа ФПТ в число ФФТ. Число в ФФТ состоит из первого левого бита — знака и остальных битов — модуля числа.

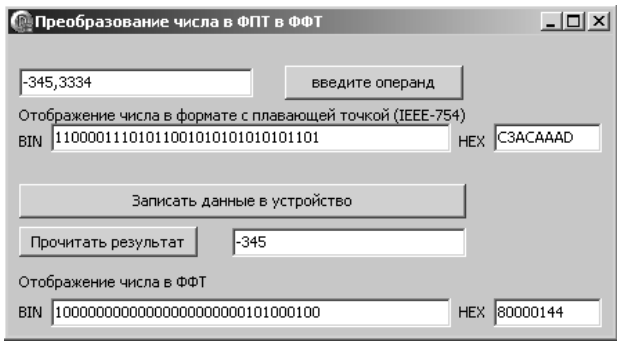

Рис. 4. Графический интерфейс модуля программы для операции преобразования форматов

Кроме функциональных секций, в программе имеется секция «Меню», которая имеет кнопку быстрого завершения работы и кнопку, открывающую информационное окно о продукте.

Разработанный проект включает (помимо устройства управления и контроллера шины PCI) следующие составные функциональные блоки с плавающей точкой: сложения, умножения, деления и преобразования форматов. Проект занимает 72% ресурсов кристалла ПЛИС типа Spartan 3 XC3S400 (2586 Slice из 3584 Slices, 4 из 16 умножителя 16×16) и характеризуется следующими частотными параметрами: сложение (вычитание) — конвейер с 5 ступенями (75 МГц); деление — конечный автомат с 6 состояниями (100 МГц); умножение — комбинационная схема (45 МГц); преобразование форматов (71 МГц).

Выводы

Разработаны принципы построения и функционирования нового класса устройств вычислительной техники и систем управления с реконфигурируемой архитектурой на базе кристаллов ПЛИС, архитектура которых отличается от традиционных архитектур фоннеймановского типа свойствами динамической реконфигурируемости, многоуровневости и параллельности обработки данных, что позволяет разработчику (пользователю) создавать функциональные средства вычислительной техники для произвольного алгоритма, обеспечивая при этом возможность адаптации в зависимости от выполняемой задачи (алгоритма).

Модифицирован известный логико-информационный подход к проектированию применительно к реконфигурируемым устройствам на ПЛИС, который

позволяет установить однозначное соответствие между объектами логико-информационной модели — алгоритмами, фрагментами алгоритмов и их информационно-кодовыми представлениями на всех уровнях иерархической системы, какой представляется проектируемое устройство.

Верификация разработанных блоков в реальной среде осуществляется с помощью программного комплекса, который включает в себя драйвер устройства, библиотеку функций для работы с драйвером, программу с модулями для верификации каждого блока в отдельности. При этом существенным достоинством программы является отображение векторов значения в форматах в двоичной и шестнадцатеричной форме, что позволяет использовать эти значения в системе моделирования ModelSim, которая отображает число с плавающей точкой только в виде вектора значения.

Верификация и тестирование разработанных функциональных блоков подтвердили корректную работу блоков, которые могут быть использованы любыми пользователями в своих проектах в виде готовых IP-Core.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Estrin G. Organization of computer system: the fixed plus variable structure computer // Proc. Western Joint Computer Conf.— 1960.— N 5.— P. 33—40.

2. Микропроцессорные системы обработки информации / А. В. Палагин, Е. Л. Денисенко, Р. И. Белицкий, В. И. Сигалов.— К.: Наук. думка, 1993.

3. Cosoroaba A., Rivoallon F. Achieving higher system performance with the virtex-5 family of FPGAs / Xilinx Inc., White Paper WP245 (v1.1), May 17, 2006 / Available at http: // www.xilinx.com.

4. Палагин А. В. Перспективы развития и вопросы теории проектирования микропроцессоров и микроЭВМ // УСиМ.— 1982.— № 3.— С. 24—29.

5. Палагин А. В., Опанасенко В. Н. Реконфигурируемые вычислительные системы.— К.: Просвіта, 2006.

6. Сергиенко И. В., Шило В. П. Задачи дискретной оптимизации. Проблемы, методы решения, исследования.— К.: Наук. думка, 2003.

7. Палагин А. В., Опанасенко В. Н., Лисовый А. Н. Проектирование реконфигурируемых систем на ПЛИС / Тр. Науч.-практич. конф. «СИЭТ-2006». Т. 1.— Одесса.— 2006.— С. 164.

8. Солдатов В. П. Программирование драйверов Windows.— М.: ООО «Бином-Пресс», 2004.

Д. т. н. В. Л. КОСТЕНКО, С. О. ЖАРОВЦЕВ

Украина, Одесский национальный политехнический университет
E-mail: staz@normaplus.com

Дата поступления в редакцию
06.10 2006 г. — 14.02 2007 г.

Оппонент *А. И. РАДКЕВИЧ*
(НИИ микроприборов, г. Киев)

МНОГОКАНАЛЬНОЕ ВЫСОКОТОЧНОЕ УСТРОЙСТВО УПРАВЛЕНИЯ НЕЙРОЧИПОМ

Предложено многоканальное высокоточное устройство управления нейрочипом, позволяющее минимизировать погрешность при формировании значений весовых коэффициентов синапсов и расширить диапазон их рабочих значений.

Современные нейрочипы строятся на основе сложных многонейронных сетей, использующих комбинированную технологию, основанную на применении в одном нейрочипе цифровых и аналоговых схем [1], в состав которых входят нейроны, матрица связей, схемы обучения и восстановления, АЦП и ЦАП. В настоящее время для универсализации нейрочипов и повышения гибкости их нейросетей используют синапсы, обеспечивающие значения весовых коэффициентов в широком диапазоне. Поэтому минимальная погрешность, внесенная в значение весового коэффициента при его формировании, приведет к значительному снижению точности вычислений. Следовательно, актуальными представляются исследования возможности минимизации погрешности при формировании значений весовых коэффициентов синапсов.

Необходимо учитывать также, что при разработке нейрочипа задачу повышения производительности нельзя отрывать от задачи снижения погрешности преобразования и точности задания значений весовых коэффициентов.

Известно, что для миниатюризации нейрочипа применяются внешние устройства управления [1]. Такие устройства позволяют не только задавать и восстанавливать значения весовых коэффициентов, но и обучать нейросеть. Однако применение таких устройств для управления матрицей синапсов нейросетей ограничено быстродействием выходного элемента.

Известны устройства формирования временных интервалов коммутационного типа, позволяющие уменьшить паразитную задержку, возникающую в тракте преобразования при формировании среза задающего импульса [2, 3].

Задача исследований состояла в минимизации погрешности значений весовых коэффициентов за счет оптимизации выходных элементов, согласования выходного сигнала с входами нейрочипа, снижения временной задержки при формировании среза задающих импульсов.

На основе анализа принципов построения известного многоканального формирователя задающих импульсов нами установлено, что минимум паразитной временной задержки, которая определяется суммой переходных процессов последовательно соединенных электронных блоков в тракте преобразования, достигается за счет включения двухступенчатой памяти на базе Т-триггера и Д-триггера с динамическим син-