

Раздел I. Принципы построения и архитектура суперкомпьютеров

УДК 004.315

DOI 10.18522/2311-3103-2016-11-413

Ю.И. Доронченко, А.Г. Коваленко

МЕТОДИКА РЕАЛИЗАЦИИ НА РЕКОНФИГУРИРУЕМЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ КРУПНЫХ УЗЛОВ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ В ВИДЕ БИБЛИОТЕЧНЫХ VHDL-ЭЛЕМЕНТОВ

Рассматривается методика реализации крупных вычислительных узлов в виде библиотечных VHDL-элементов для реконфигурируемых вычислительных систем. Представлена библиотека схмотехнических элементов для программируемых логических интегральных схем (ПЛИС) UltraScale фирмы Xilinx. Библиотека разработана в «Научно-исследовательском центре супер-ЭВМ и нейрокомпьютеров», г. Таганрог. Все элементы реализованы на языке проектирования VHDL, что позволяет использовать их не только для указанного семейства ПЛИС, но и для ПЛИС различных производителей и семейств. Наибольшие возможности библиотека предоставляет программистам на языке высокого уровня COLAMO, поскольку применение того или иного элемента библиотеки происходит автоматически в зависимости от программного кода. Библиотека схмотехнических элементов разбита на группы по функциональному назначению (коммутация, операции над числами в формате с плавающей точкой, операции над целыми числами, логические операции, хранение данных и т.д.), обрабатываемой разрядности (переменная разрядность, 32-х разрядные, 64-х разрядные, иные разрядности) и наличию триггеров на выходных шинах элементов и включает в себя 20 групп. Всего библиотека содержит VHDL-описание более 300 схмотехнических элементов. Тактовая частота работы элементов варьируется от 450 до 550 МГц на ПЛИС семейства UltraScale. Текущая структура представления схмотехнических элементов адаптирована для их удобного использования в среде разработки на языке COLAMO. VHDL-описание библиотечных элементов содержит в себе специальную структуру комментариев, предназначенных для синтезатора Fire!Constructor, входящего в программный комплекс разработки прикладных программ. Описана методика, которая позволяет создавать библиотеку крупных узлов вычислительных задач для PBC. Разработанная библиотека схмотехнических элементов может быть полезна как опытным разработчикам для сокращения времени подготовки проектов, так и начинающим схмотехникам и программистам для достижения достаточно высоких характеристик производительности своих решений без многократных оптимизаций кода и углубленного изучения архитектуры ПЛИС. Она может корректироваться, улучшаться и дополняться пользователем исходя из предметной области решаемых задач. Предлагаемая методика реализации крупных вычислительных узлов в виде библиотечных VHDL-элементов позволяет сократить время реализации вычислительно трудоемких задач на PBC до 30 %.

Реконфигурируемая вычислительная система; программирование ПЛИС; язык VHDL; COLAMO; библиотека схмотехнических элементов.

Yu.I. Doronchenko, A.G. Kovalenko

**METHODOLOGY OF IMPLEMENTATION OF LARGE NODES
OF COMPUTATIONAL PROBLEMS AS LIBRARY VHDL-UNITS
ON RECONFIGURABLE COMPUTER SYSTEMS**

The paper covers a methodology of implementation of large computational nodes as library VHDL-units for reconfigurable computer systems. A library of circuit elements for Xilinx UltraScale FPGAs is presented. The library was developed in Scientific Research Centre of Supercomputers and Neurocomputers (Taganrog, Russia). All elements were implemented in the VHDL language. Owing to this they can be used not only for the abovementioned FPGA family but for FPGAs of various vendors and families. The maximal capabilities of the library are given to developers who create applications in the high-level language COLAMO, because use of any element from the library is automatic and depends on the program code. The library of circuit elements is divided into groups according to functions (communication, floating point operations, integer operations, logical operations, data storage, etc.), capacity (variable capacity, 32-digit capacity, 64-digit capacity, other capacities) and presence/absence of triggers in output buses of elements. The total number of groups is 20. The library contains VHDL-descriptions of more than 300 circuit elements. The clock rate of the elements is from 450 to 550 MHz for UltraScale FPGAs. The current structure of circuit elements description is adapted for their use in the development environment using the language COLAMO. VHDL-description of the library elements contains a specialized structure of commentaries for a synthesizer Fire!Constructor, which is a tool of the application development software suit. The methodology, which allows creating a library of large nodes of computational problems for RCS, is described. The developed library of circuit elements can be useful for experienced developers for reduction of the projects development time, and for entry-level circuit engineers and applications developers. It will help them to achieve rather high performance of their solutions without multiple optimizations of code and detailed analysis of FPGA structure, which can be modified, improved and added by the developer depending on the problem domain of solving tasks. Owing to the suggested methodology of implementation of large computational nodes as library VHDL-units it is possible to reduce the implementation time of computationally laborious problems on RCS up to 30 %.

Reconfigurable computer structure; FPGA programming; VHDL; programming language COLAMO; library of circuit elements.

Введение. Создание прикладных программ для реконфигурируемых вычислительных систем (РВС) [1] при современном уровне интеграции ПЛИС требует от разработчика существенных временных затрат – каждая реализуемая операция должна достигать необходимого уровня эффективности по частоте реализации и используемому аппаратному ресурсу.

В настоящее время в САПР для ПЛИС наметилась устойчивая тенденция к переходу на более высокий уровень проектирования. В САПР Vivado [2] фирмы Xilinx [3] уже не поддерживается графический схемотехнический редактор, программирование ПЛИС осуществляется только на языке VHDL [4], активно развиваются возможности написания программ для ПЛИС на языке С [5]. Предполагается, что в дальнейшем разработчику не нужно будет знать архитектурных особенностей ПЛИС, эффективность отображения программного кода на ресурсы кристалла должен взять на себя транслятор.

Тем не менее, эффективность существующих средств трансляции, в том числе программ, написанных на языках С-группы, остается довольно низкой. Использование VHDL или IP-ядер по-прежнему требует специфичных знаний для достижения высокой тактовой частоты. Зачастую опытные разработчики имеют в своем арсенале собственные эффективно реализованные функции, которые могут быть использованы многократно в различных проектах.

Библиотека схмотехнических элементов. В НИЦ супер-ЭВМ и нейрокомпьютеров [6] разработана подобная библиотека схмотехнических элементов. Элементы реализованы на языке VHDL, что в целом позволяет использовать их для ПЛИС различных производителей и семейств, однако при разработке целевым являлось семейство UltraScale фирмы Xilinx [7, 8].

Наибольшие возможности разработанная библиотека предоставляет программистам на языке высокого уровня COLAMO [9] при использовании среды разработки прикладных программ, созданной коллективами НИИ многопроцессорных вычислительных систем Южного федерального университета [10] и НИЦ супер-ЭВМ и нейрокомпьютеров. Применение того или иного элемента библиотеки происходит в данном случае автоматически в зависимости от программного кода.

Библиотека схмотехнических элементов разбита на группы по функциональному назначению, обрабатываемой разрядности и наличию триггеров на выходных шинах элементов. Библиотека включает в себя следующие 20 групп:

- ◆ «COMMON» – элементы общего назначения; в группу входят служебные элементы, счетчики, программируемые буферы задержек данных и пр.;
- ◆ «COMMUTATION» – элементы, предназначенные для коммутации данных;
- ◆ «FP32» – элементы, выполняющие математические операции и операции сравнения над числами в формате с плавающей точкой одинарной точности;
- ◆ «FXS_RG» – элементы переменной разрядности, выполняющие математические операции и операции сравнения над знаковыми целыми числами, с триггерами на выходах;
- ◆ «FXS_OTHERS» – элементы фиксированной разрядности, выполняющие математические операции над знаковыми целыми числами;
- ◆ «FXU» – элементы переменной разрядности, выполняющие математические операции и операции сравнения над беззнаковыми целыми числами;
- ◆ «FXU_OTHERS» – элементы фиксированной разрядности, выполняющие математические операции над беззнаковыми целыми числами;
- ◆ «LOGIC» – элементы переменной разрядности, выполняющие типовые логические операции для 2-6 операндов;
- ◆ «LOGIC_OTHERS» – элементы фиксированной разрядности, выполняющие логическую операцию для 32 операндов;
- ◆ «STORAGE» – элементы переменной разрядности, предназначенные для хранения пользовательских данных.

Группы «COMMON_64_2T», «FP64_2T», «FXS64_2T», «FXU64_2T», «STORAGE_64_2T» выполняют операции с данными разрядностью 64 бита, которые передаются по 32-битным шинам за два такта.

Группы «COMMON_64», «FP64» выполняют операции с данными разрядностью 64 бита, которые передаются по 64-битным шинам за один такт.

В группах «COMMUTATION_RG», «FXU_RG», «LOGIC_RG» схмотехнические элементы содержат триггеры на всех выходах. В среде разработки на языке COLAMO это позволяет полностью исключить работу по расположению в нужных местах схемы прореживающих триггеров для трассировки сигналов на необходимую тактовую частоту. Синхронизирующие триггеры устанавливаются автоматически. Однако при таком подходе количество триггеров может существенно вырасти.

Все элементы, выполняющие операции в формате с плавающей запятой, в том числе тригонометрические и логарифмические операции, соответствуют спецификациям стандарта IEEE-754 [11], при этом разработан ряд элементов с сокращенным аппаратным ресурсом, запрещающих обработку денормализованных чисел с плавающей запятой.

На рис. 1 изображено представление библиотеки в системе проектирования Vivado фирмы Xilinx. Всего библиотека содержит VHDL-описание более 300 схмотехнических элементов. Тактовая частота работы элементов варьируется от 450 до 550 МГц на ПЛИС семейства UltraScale с параметром “speed grade”, равным -1.

Адаптация библиотеки для системы программирования на языке COLAMO. Текущая структура представления схмотехнических элементов адаптирована для их удобного использования в среде разработки на языке COLAMO [12, 13]. При написании программы разработчик при помощи директивы include указывает операционную библиотеку элементов и группу элементов, которую нужно использовать в проекте, в формате:

Include <название_библиотеки>.<название_группы>;

Для удобства пользователь может выбрать нужные группы элементов из выпадающего списка, как это показано на рис. 2.

VHDL-описание библиотечных элементов содержит в себе специальную структуру комментариев, предназначенных для синтезатора Fire!Constructor, входящего в программный комплекс разработки прикладных программ [14, 15].

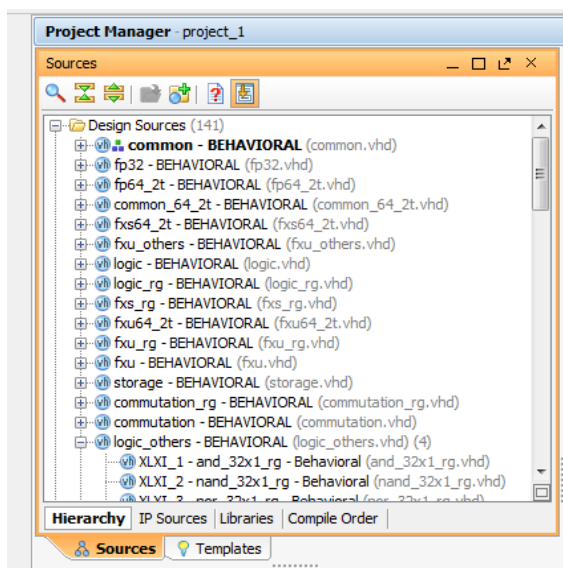


Рис. 1. Представление библиотеки схмотехнических элементов в САПР Vivado фирмы Xilinx

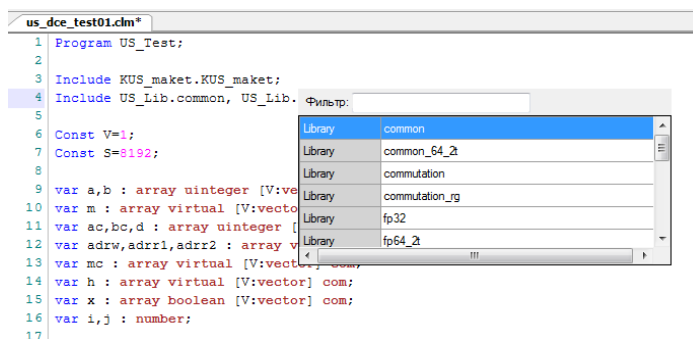


Рис. 2. Выбор групп элементов из библиотеки в среде разработки

Для автоматизации синтеза вычислительной структуры средствами синтезатора Fire!Constructor необходимо передавать различные свойства элемента с помощью специальной структуры комментариев, которая должна начинаться служебным словом «--CFC». Комментарии делятся на 2 вида:

- ◆ общие комментарии – информация, относящаяся ко всему библиотечному элементу;
- ◆ комментарии к выводам – информация по каждому выводу.

Общие комментарии оформляются каждый в виде отдельной строки в любом месте (обычно в начале) файла, содержащего VHDL-описание элемента, и содержат следующие сведения:

- ◆ Version – версия элемента;
- ◆ ElementFPGA – целевое семейство ПЛИС;
- ◆ ElementTypeApplication – тип применения элемента («scheme» – операционный элемент, «interface» – интерфейсный элемент);
- ◆ ElementTypeProcessing – тип обработки данных (конвейер/процедура);
- ◆ ElementFrequency – максимальная рабочая частота элемента (МГц);
- ◆ ElementResource : Lt, FF, LUT, RAM, DSP – справочная информация о порядке внесения параметров в разделе Resource;
- ◆ Resource – указываются целочисленные значения параметров использованного элементом ресурсов ПЛИС.

Ниже приведены типы выводов библиотечных элементов, которые указываются в комментариях к ним:

- ◆ CLK_IN – вход подключения сигнала от источника рабочей тактовой частоты проекта;
- ◆ CLK_OUT – вывод-источник рабочей тактовой частоты проекта;
- ◆ Reset_IN – вход «инициализация элемента» (сброс);
- ◆ Reset_OUT – выход, через который в вычислительную структуру задачи подаётся сигнал инициализации (сброс) для всех элементов;
- ◆ Marker_IN – вход «Маркер» (сигнал, сопровождающий некоторый поток данных);
- ◆ Marker_OUT – вывод-источник сигнала «Маркер»;
- ◆ Data_IN – вход данных в элементе;
- ◆ Data_OUT – вывод-источник данных;
- ◆ ICLK – вывод «Тактовая частота вычислительного модуля», подключаемый к определённому выводу кристалла ПЛИС (данный вывод указывается в файле описания структуры PBC);
- ◆ IReset – вывод «Сброс вычислительного модуля», подключаемый к определённому выводу кристалла ПЛИС (данный вывод указывается в файле описания структуры PBC).

На рис. 3 показан пример общего комментария к библиотечному элементу `comp_eq_fxs_ddd_rg_vnd`, а на рис. 4 – пример описания входов-выводов библиотечного элемента `dmx2x1_control_rg`.

```

3 -- Description: fixed point signed comparator
4 -----
5
6 --CFC Version 2.0
7 --CFC ElementFPGA Kintex-US
8 --CFC ElementName comp_eq_fxs_ddd_rg_vnd
9 --CFC ElementClass comp_eq_fxs_ddd_rg_vnd
10 --CFC ElementTypeApplication scheme
11 --CFC ElementTypeProcessing pipeline
12 --CFC ElementFrequency 550
13
14 --CFC ElementResource 1 (WI) : Lt, D1, FF, LUT, LUTm, BRAM, DSP
15 --CFC Resource 1 (WI) : 1,0,1,<WI/3>,0,0,0

```

Рис. 3. Пример общего комментария к библиотечному элементу *comp_eq_fxs_ddd_rg_vnd*

```

21 entity dmx2x1_control_rg is
22   port ( CLK : in   std_logic;           --CFC -CLK_In
23         D   : in   std_logic;           --CFC -Marker_In -0 -0
24         HD  : in   std_logic_vector (31 downto 0); --CFC -Data_IN -0 -1 -xxx
25         HS  : in   std_logic;           --CFC -Marker_In -0 -1
26         CS  : in   std_logic_vector (31 downto 0); --CFC -Data_IN -0 -2 -xxx
27         RST : in   std_logic;           --CFC -Reset_in
28         Q0  : out  std_logic;           --CFC -Marker_OUT -0 -0
29         Q1  : out  std_logic;           --CFC -Marker_OUT -0 -1
30   );
31 end dmx2x1_control_rg;

```

Рис. 4. Пример описания входов-выводов библиотечного элемента *dmx2x1_control_rg*

Методика реализации крупных узлов вычислительных задач в виде библиотечных элементов. На основании принципов программирования реконфигурируемых вычислительных систем и методологии создания параллельных прикладных программ для РВС была разработана методика реализации крупных узлов вычислительных задач в виде библиотечных VHDL-элементов с использованием языка высокого уровня COLAMO и описанной выше библиотеки.

Существуют предметные области, в которых разные задачи оперируют одними и теми же крупными вычислительными узлами в различной комбинации и с различными организациями вычислений (параллельная, конвейерная, мультипроцедурная и др.). В случае, когда в создании прикладных программ для РВС, реализующих подобные задачи, участвует множество разработчиков, необходимо обладать набором верифицированных и эффективно реализованных узлов, чтобы избежать возможных ошибок при передаче частей программного кода, а также многократной отладки одинаковых частей алгоритма решения задачи. Использование библиотеки крупных вычислительных узлов сокращает время разработки параллельных прикладных программ для РВС.

Предлагаемая методика позволяет создавать библиотеку крупных узлов вычислительных задач. Рассмотрим этапы методики для создания одного библиотечного элемента.

1. В информационном графе решаемой задачи G выделить крупные вычислительные узлы g_i , являющиеся функционально законченными фрагментами, которые реализуют повторяющиеся или типовые фрагменты задачи:

$$G = \bigcup_{i=1}^m g_i \bigcup_{i,j=1}^m Z(g_i, g_j),$$

где $Z(g_i, g_j)$ – множество связей между подграфами g_i, g_j .

2. Выполнить предварительную оценку требуемого аппаратного ресурса A_{g_i} для реализации выбранного из множества фрагментов крупного вычислительного узла g_i на одном кристалле ПЛИС целевой РВС, аппаратный ресурс которого равен A . Данная оценка необходима, поскольку ограничением при создании библиотечного элемента является условие его полного размещения в одной ПЛИС РВС.

3. Если аппаратного ресурса ПЛИС целевой РВС достаточно для реализации выбранного вычислительного узла ($A_{g_i} < A$), то перейти к п. 4. В противном случае ($A_{g_i} \geq A$) необходимо выполнить разделение вычислительного узла g_i на более мелкие функционально законченные фрагменты:

$$g_i = \bigcup_{j=1}^n q_j \bigcup_{j,k=1}^n Z(q_j, q_k),$$

и повторить п. 2 и остальные пункты методики для каждого фрагмента q_j .

4. Описать на языке COLAMO с помощью конструкции subcadr [16] выделенный крупный вычислительный узел g_i .

5. Выполнить трансляцию прикладной программы с помощью системного программного обеспечения для РВС [17]. При этом каждой вычислительной операции из текста программы ставится в соответствие библиотечный элемент из разработанной библиотеки. В результате для каждой ПЛИС формируется набор VHDL-файлов [14], среди которых имеется файл с описанием вычислительной структуры описываемого подкадра.

6. Для полученного файла VHDL-элемента выполнить этап размещения элементов и трассировки сигналов в соответствии с временными ограничениями (Implementation) [18], что выполняется с помощью средств автоматического проектирования [19].

7. В VHDL-файле элемента указать комментарии общего вида. Для заполнения полей «ElementFrequency» и «Resource» комментарий к создаваемому библиотечному элементу взять информацию о требуемом аппаратном ресурсе из отчетов о размещении и трассировки элемента, которые были сформированы в результате выполнения п. 6.

8. В файле VHDL-элемента в области описания интерфейса VHDL-компонента (entity) [20] для каждого порта (port) указать комментарий, описывающий назначение входа/вывода.

9. Выполнить добавление нового элемента в библиотеку схмотехнических элементов для программного обеспечения, предназначенного для программирования РВС.

Приведенная методика может быть использована без применения языка программирования COLAMO. В этом случае вместо пунктов 4 и 5 методики необходимо самостоятельно описать вычислительную структуру нового элемента на языке VHDL.

Заключение. В работе рассмотрена библиотека VHDL-элементов для реконфигурируемых вычислительных систем на основе ПЛИС и методика реализации крупных вычислительных узлов в виде библиотечных элементов. Разработанная библиотека схмотехнических элементов может быть полезна как опытным разработчикам для сокращения времени подготовки проектов, так и начинающим схемотехникам и программистам для достижения достаточно высоких характеристик производительности своих решений без многократных оптимизаций кода и углубленного изучения архитектуры ПЛИС. Кроме того, библиотека может корректироваться, улучшаться и дополняться пользователем исходя из предметной области решаемых задач. Предлагаемая методика реализации крупных вычислительных узлов сокращает время реализации вычислительно трудоемких задач на РВС до 30 %.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Левин И.И., Дордопуло А.И., Каляев И.А., Доронченко Ю.И., Раскладкин М.К.* Современные и перспективные высокопроизводительные вычислительные системы с реконфигурируемой архитектурой // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. – 2015. – Т. 4, № 3. – С. 24-39.
2. <http://www.xilinx.com/products/design-tools/vivado.html> (дата обращения: 14.10.2016).
3. <http://www.xilinx.com> (дата обращения: 14.10.2016).
4. *Бибило П.Н.* Основы языка VHDL. – М.: СОЛОН-Р, 2002. – 224 с.
5. Языки высокого уровня ImpulseC, Mitrion-C и Handle-C. – Режим доступа: <https://parallel.ru/fpga/lang.html> (дата обращения: 14.10.2016).
6. <http://www.superevm.ru> (дата обращения: 14.10.2016).
7. <https://www.xilinx.com/products/silicon-devices/fpga/kintex-ultrascale.html> (дата обращения: 14.10.2016).
8. <https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale.html> (дата обращения: 14.10.2016).
9. *Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И.* Реконфигурируемые мультимасштабные вычислительные структуры. – 2-е изд., перераб. и доп. / под общ. ред. И.А. Каляева. – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. – 344 с.
10. <http://www.mvs.sfedu.ru> (дата обращения: 14.10.2016).
11. 754-2008 - IEEE Standard for Floating-Point Arithmetic. – Режим доступа: <http://ieeexplore.ieee.org/document/4610935/> (дата обращения: 14.10.2016).
12. *Дордопуло А.И., Гудков В.А., Гуленок А.А.* Программирование многокристалльных реконфигурируемых вычислительных систем на языках высокого уровня Mitrion-C и COLAMO // Материалы VI международной научно-практической конференции «Академическая наука – проблемы и достижения» 25-26 мая 2015 г. – North Charleston, SC, USA, 2015. – Т. 1. – С. 122-127.
13. *Gudkov V.A., Gulenok A.A., Kovalenko V.B., Slaven L.M.* Multi-level Programming of FPGA-based Computer Systems with Reconfigurable Macroobject Architecture // IFAC Proceedings Volumes (ISSN 14746670), Programmable Devices and Embedded Systems. – 2013. – Vol. 12, part 1. – P. 204-209.
14. *Левин И.И., Дордопуло А.И., Гуленок А.А.* Синтез параллельных прикладных программ для многокристалльных реконфигурируемых вычислителей. Синтезатор Fire!Constructor: учеб. пособие. – Таганрог: Изд-во ТТИ ЮФУ, 2013. – 96 с.
15. *Бовкун А.В., Левин И.И., Дордопуло А.И., Гуленок А.А.* Средства трансляции параллельных программ на уровень логических ячеек ПЛИС для многокристалльных реконфигурируемых вычислительных систем // Труды Первого Международного симпозиума «Гибридные и синергетические интеллектуальные системы: теория и практика», 29 июня-2 июля 2012 г., г. Светлогорск Калининградская область. – Калининград: Изд-во БФУ им. Канта, 2012. – С. 97-107.
16. *Левин И.И., Дордопуло А.И., Гудков В.А.* Программирование реконфигурируемых вычислительных узлов на языке COLAMO: учеб. пособие. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – 114 с.
17. *Гузик В.Ф., Каляев И.А., Левин И.И.* Реконфигурируемые вычислительные системы: учеб. пособие / под общ. ред. И.А. Каляева. – Ростов-на-Дону: Изд-во ЮФУ, 2016. – 472 с.
18. *Перельройзен Е.З.* Проектируем на VHDL. – М.: СОЛОН-Пресс, 2004. – 448 с.
19. *Максфилд К.* Проектирование на ПЛИС. Курс молодого бойца. – М.: Издательский дом «Додэка-XXI», 2007. – 408 с.
20. *Сергиенко А.М.* VHDL для проектирования вычислительных устройств – К.: ЧП «Корнейчук», ООО «ТИД «ДС», 2003. – 208 с.

REFERENCES

1. *Levin I.I., Dordopulo A.I., Kalyaev I.A., Doronchenko Yu.I., Raskladkin M.K.* Sovremennye i perspektivnye vysokoproizvoditel'nye vychislitel'nye sistemy s rekonfiguriruemoy arkhitekturoy [Modern and perspective high performance computer systems with reconfigurable architecture], *Vestnik Yuzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Vychislitel'naya matematika i informatika* [Vestnik of the South Ural state University. Series: Computational mathematics and computer science], 2015, Vol. 4, No. 3, pp. 24-39.

2. Available at: <http://www.xilinx.com/products/design-tools/vivado.html> (accessed 14 October 2016).
3. Available at: <http://www.xilinx.com> (accessed 14 October 2016).
4. *Bibilo P.N.* Osnovy yazyka VHDL [The basics of the language VHDL]. Moscow: SOLON-R, 2002, 224 p.
5. Yazyki vysokogo urovnya ImpulseC, Mittrion-C i Handle-C [High-level languages ImpulseC, Mittrion-C and Handle C]. Available at: <https://parallel.ru/fpga/lang.html> (accessed 14 October 2016).
6. Available at: <http://www.superevm.ru> (accessed 14 October 2016).
7. Available at: <https://www.xilinx.com/products/silicon-devices/fpga/kintex-ultrascale.html> (accessed 14 October 2016).
8. Available at: <https://www.xilinx.com/products/silicon-devices/fpga/virtex-ultrascale.html> (accessed 14 October 2016).
9. *Kalyaev I.A., Levin I.I., Semernikov E.A., Shmoylov V.I.* Rekonfiguriruyemye mul'tikonveyernye vychislitel'nye struktury [Multiconference reconfigurable computing structure]. 2nd ed., under the ed. I.A. Kalyaeva. Rostov-on-Don: Izd-vo YuNTs RAN, 2009, 344 p.
10. Available at: <http://www.mvs.sfedu.ru> (accessed 14 October 2016).
11. 754-2008 - IEEE Standard for Floating-Point Arithmetic. Available at: <http://ieeexplore.ieee.org/document/4610935/> (accessed 14 October 2016).
12. *Dordopulo A.I., Gudkov V.A., Gulenok A.A.* Programmirovaniye mnogokristal'nykh rekonfiguriruyemykh vychislitel'nykh sistem na yazykakh vysokogo urovnya Mittrion-C i COLAMO [Programming a multi-chip reconfigurable computing systems high-level language Mittrion-C and COLAMO], *Materialy VI mezhdunarodnoy nauchno-prakticheskoy konferentsii «Akademicheskaya nauka – problemy i dostizheniya» 25-26 maya 2015 g.* [Materials of VI international scientific-practical conference "Academic science – problems and achievements" on 25-26 may 2015]. North Charleston, SC, USA, 2015, Vol. 1, pp. 122-127.
13. *Gudkov V.A., Gulenok A.A., Kovalenko V.B., Slasten L.M.* Multi-level Programming of FPGA-based Computer Systems with Reconfigurable Macroobject Architecture, *IFAC Proceedings Volumes (ISSN 14746670), Programmable Devices and Embedded Systems 2013 Vol. 12, part 1*, pp. 204-209.
14. *Levin I.I., Dordopulo A.I., Gulenok A.A.* Sintez parallel'nykh prikladnykh programm dlya mnogokristal'nykh rekonfiguriruyemykh vychisliteley. Sintezator Fire!Constructor: ucheb. posobie [Synthesis of parallel programs for multi-chip reconfigurable computers. Synth Fire!Constructor: a training manual]. Taganrog: Izd-vo TTI YuFU, 2013, 96 p.
15. *Bovkun A.V., Levin I.I., Dordopulo A.I., Gulenok A.A.* Sredstva translyatsii parallel'nykh programm na uroven' logicheskikh yacheek PLIS dlya mnogokristal'nykh rekonfiguriruyemykh vychislitel'nykh sistem [Means of translating parallel programs on the level of logical cells of the FPGA for multi-chip reconfigurable computing systems], *Trudy Pervogo Mezhdunarodnogo simpoziuma «Gibridnye i sinergeticheskie intellektual'nye sistemy: teoriya i praktika», 29 iyunya-2 iyulya 2012 g., g. Svetlogorsk Kaliningradskaya oblast'* [Proceedings of the First International Symposium on "Hybrid and synergetic intelligent systems: theory and practice", June 29-July 2, 2012, Svetlogorsk, Kaliningrad oblast]. Kaliningrad: Izd-vo BFU im. Kanta, 2012, pp. 97-107.
16. *Levin I.I., Dordopulo A.I., Gudkov V.A.* Programmirovaniye rekonfiguriruyemykh vychislitel'nykh uzlov na yazyke COLAMO: ucheb. posobie [Programming reconfigurable computational nodes in the language COLAMO: a training manual]. Taganrog: Izd-vo TTI YuFU, 2011, 114 p.
17. *Guzik V.F., Kalyaev I.A., Levin I.I.* Rekonfiguriruyemye vychislitel'nye sistemy: ucheb. posobie [Reconfigurable computing systems: the textbook], ed. by I.A. Kalyaeva. Rostov-on-Don: Izd-vo YuFU, 2016, 472 p.
18. *Perel'royzen E.Z.* Proektiruemye na VHDL [Design in VHDL]. Moscow: SOLON-Press, 2004, 448 p.
19. *Maksfield K.* Proektirovaniye na PLIS. Kurs molodogo boytsa [The design on the FPGA. Course of the young fighter]. Moscow: Izdatel'skiy dom «Dodeka-XXI», 2007, 408 p.
20. *Sergienko A.M.* VHDL dlya proektirovaniya vychislitel'nykh ustroystv [VHDL for the design of computing devices]. Kiev: ChP «Korneychuk», OOO «TID «DS», 2003, 208 p.

Статью рекомендовал к опубликованию д.т.н., профессор И.И. Левин.

Коваленко Алексей Геннадьевич – Общество с ограниченной ответственностью «НИЦ супер-ЭВМ и нейрокомпьютеров»; e-mail: k.a.g@bk.ru; 347928, г. Таганрог, пер. Итальянский, 106; тел.: 89281890413, 88634612111; к.т.н.; начальник сектора.

Доронченко Юрий Иванович – e-mail: doronchenko@superevm.ru; тел.: 88634648078, 88634612111; к.т.н.; технический директор.

Kovalenko Alexey Gennadievich – “Scientific Research Centre of Supercomputers and Neurocomputers” Co Ltd; e-mail: k.a.g@bk.ru; 106, Italyanskiy lane, Taganrog, 347928, Russia; phones: +79281890413, +78634612111; cand. of eng. sc.; head of sector.

Doronchenko Yuri Ivanovich – e-mail: doronchenko@superevm.ru; phone: +78634648078, +78634612111; cand. of eng. sc.; technical director.

УДК 004.272.34

DOI 10.18522/2311-3103-2016-11-1326

И.П. Осинин

МОДУЛЯРНО-ЛОГАРИФМИЧЕСКИЙ ПРОЦЕССОР*

Представлена организация перспективного арифметического модулярно-логарифмического процессора. Ключевое отличие от аналогов состоит в использовании непозиционной системы счисления на базе однородных вычислительных сред, что распараллелило вычисления вплоть до разрядов операнда, существенно повысив скорость вычислений, а также привнесло ряд уникальных свойств, значительно повысив надежность процесса счета. Применение системы остаточных классов совместно с логарифмической системой счисления вместо плавающей точки позволило использовать диапазон представления чисел аналогичный формату IEEE-754, отказавшись при этом от медленного округления и выравнивания порядков. Текущая реализация в виде IP-блока софт-процессора подтверждает реализуемость архитектурных особенностей, описанных в статье. В перспективе такой процессор может быть создан не только в качестве дополнения системы на кристалле, но и как самостоятельное устройство. Например, в виде арифметического ускорителя, подключаемого к компьютеру с традиционной архитектурой, либо в качестве центрального процессора на базе СБИС. В любом из этих вариантов высокая производительность и масштабируемость и реконфигурируемость достигается за счет применения однородных вычислительных сред, функционирующих в базе системы остаточных классов и логарифмической системы счисления. Сферами применения МЛП являются: высокопроизводительные вычисления, требующие работы с большой разрядностью данных – 1024 бит и более (задачи моделирования климата, исследование электромагнитного рассеяния, исследование орбитальной эволюции небесных тел и другие), где за счет распараллеливания счёта на множество независимых остатков достигается прирост производительности по сравнению с известными аналогами (например, для целых чисел минимум в два раза по сравнению с Intel AVX-512 при сокращении аппаратных затрат на 17 %); высоконадежные вычисления, требующие постоянного контроля правильности вычислений (задачи наведения ракет, управления атомной электростанцией, функционирования космических аппаратов и подобные), где за счёт применения корректирующих свойств СОК достигается уменьшение аппаратных затрат по сравнению с аналогами.

Процессор; система остаточных классов; логарифмическая система счисления; реконфигурируемая архитектура; высоконадежные вычисления.

* Исследование выполнено в ФГУП «Российский федеральный ядерный центр – Всероссийский научно-исследовательский институт экспериментальной физики».