

- экономия программных ресурсов на разработке и обслуживании сервисных функций структуры данных;
- непрерывное наличие работоспособного комплекса с поддержкой актуальной структуры данных;
- использование удобных разработчику методики готовых профессиональных инструментов для развития XML схемы.

В то же время, данный подход в его текущем виде ориентирован, в основном, на разработчиков счетных методик, а не на конечного потребителя – использование специализированных диалогов, учитывающих расширенную информацию о природе данных, удобнее для пользователя, чем работа с универсальным диалогом. В этой связи именно развитие средств конфигурирования пользовательского интерфейса видится первоочередной и наиболее востребованной задачей при развитии описанного подхода.

## Литература

1. Сайт «XML Schema. Рекомендация W3C». [Электронный ресурс]. Режим доступа: <http://www.w3.org/TR/xmlschema-0>.
2. Сайт программного продукта «The Apache Xerces Project». [Электронный ресурс]. Режим доступа: <http://xerces.apache.org/index.html>.

## КОМПОНЕНТЫ ТЕХНИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ ПРОЕКТИРОВАНИЯ СИСТЕМ НА КРИСТАЛЛЕ КОМПАНИИ ALTERA CORPORATION

*Д. В. Чернов*

Вятский государственный университет, г. Киров

Основой цифровой системы в различных областях человеческой деятельности, как правило, является микропроцессор. Это или универсальный процессор, или цифровой сигнальный процессор DSP (Digital Signal Processor). Идея интеграции устройств различного назначения в единой системе привела к появлению микроконтроллеров. В отличие от микропроцессора, серийно выпускаемые микроконтроллеры обладают большим набором периферийных устройств.

Дальнейшим развитием идеи интеграции различных устройств в одной системе стало понятие «система на кристалле» SoC (system-on-chip). Идея SoC – объединить в одном кристалле как можно большее количество элементов цифровой системы. В итоге такие системы приобретают характеристики, присущие как микроконтроллерам, так и процессорам. Увеличение логической емкости ПЛИС привело к тому, что они становятся полноправными конкурентами при реализации SoC. В программной реализации система на кристалле получила аббревиатуру SoPC (система на программируемом кристалле).

Основная идея создания SoPC состояла в том, чтобы дать разработчику полный набор стандартных средств, имеющихся у DSP и микроконтроллеров, плюс дополнительный объем кристалла и необходимый сервис для реализации своих схемных решений. Ядром всей системы может являться встраиваемый RISC-процессор Nios II, архитектура которого приведена на рис. 1.

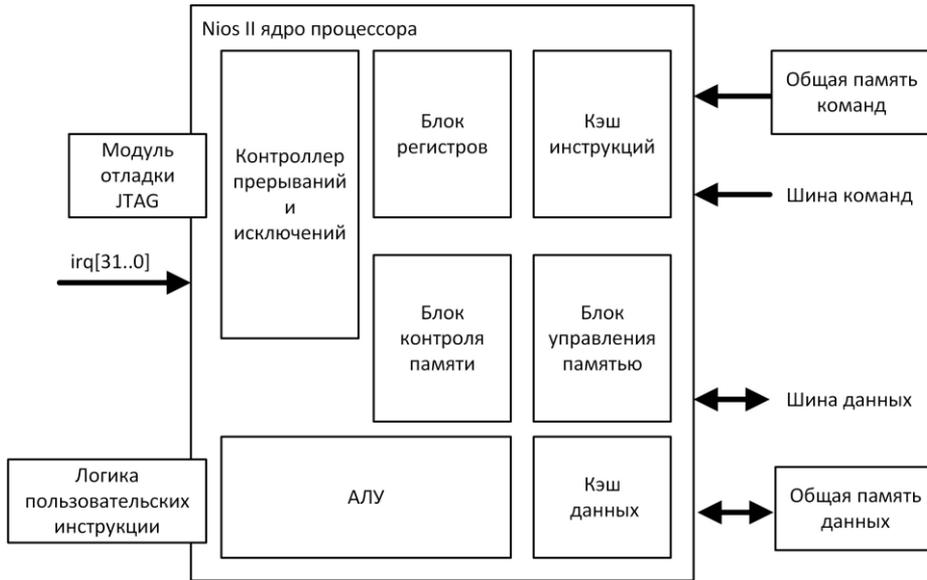


Рис. 1. Блок-схема архитектуры процессора Nios II

Nios II – программное процессорное ядро, описанное на языке программирования аппаратуры и предназначенное для реализации в ПЛИС фирмы Altera.

При этом необходимо учитывать, что для создания «системы на кристалле» к процессорному ядру необходимо добавить дополнительные компоненты – память, порты ввода/вывода, таймер, необходимые интерфейсы, что приводит к использованию дополнительных логических элементов (LE), которые могут более конструктивно использоваться при разработке специализированных систем. Например, только ядро NIOS II/f задействует более 3000 LEs [1] и более 1500 адаптивных логических модулей, а модуль отладки JTAG 4 уровня задействует 3100–3700 LEs [1]. Пример стандартной реализации системы на процессоре Nios II приведен на рис. 2.

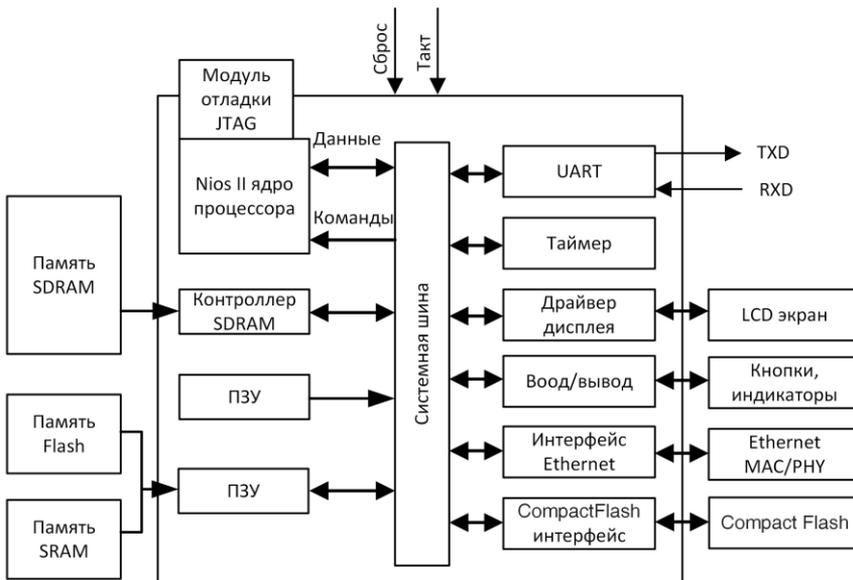
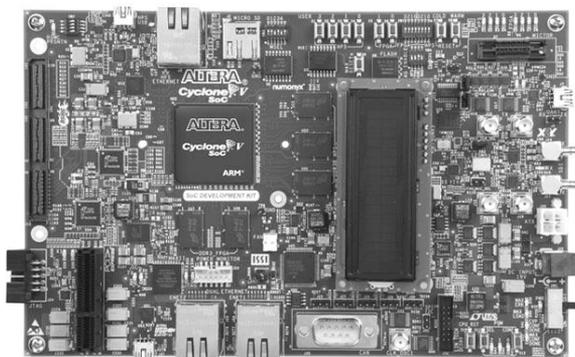


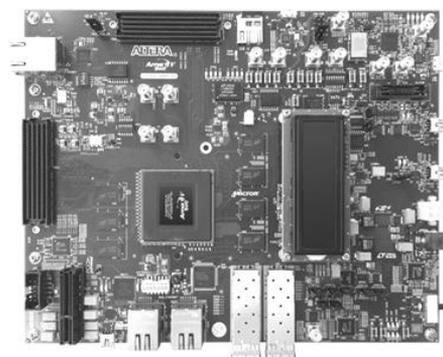
Рис. 2. Стандартная реализация системы на процессоре Nios II

В 2013 году компания Altera Corporation объявила о выпуске SoC FPGA, сочетающих двухъядерный процессор ARM Cortex-A9 и ПЛИС, и начала производство отладочных плат на основе Су-

clone V SoC и Arria V SoC, которые приведены на рис. 3. Предварительно, в 2011 году, компания Altera выпустила программный инструмент виртуального проектирования SoC Virtual Target, позволивший пользователям начать разработку программного решения до начала поставок первых SoC FPGA.



а



б

Рис. 3. Платы Altera SoC FPGA: а – Cyclone V; б – Arria V

Основное отличие Cyclone V и Arria V – в количестве логических элементов: Cyclone V содержит до 110 тысяч LEs, а Arria V – до 462 тысяч LEs. На рис. 4 приведена обобщенная схема Altera SoC FPGA.

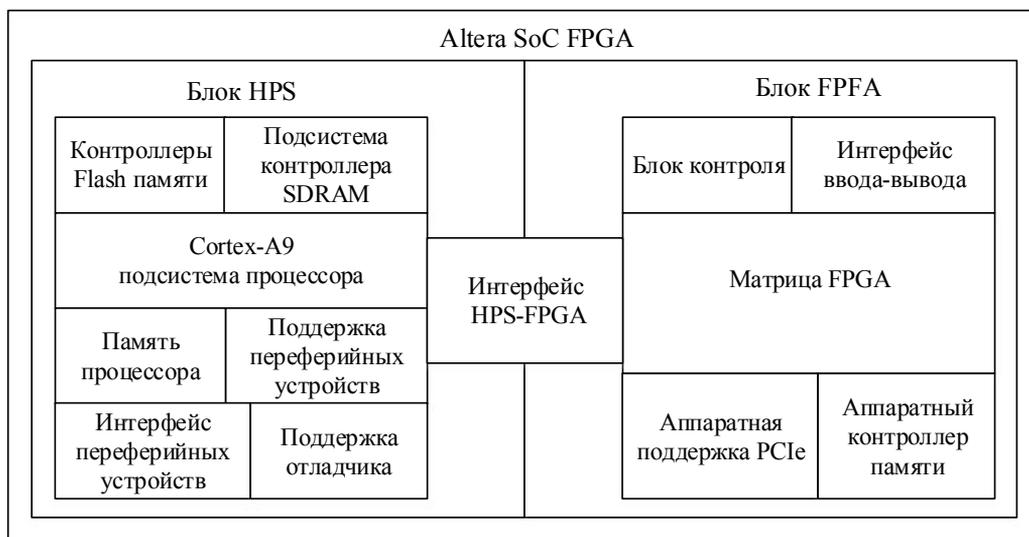


Рис. 4. Обобщенная блок-схема Altera SoC FPGA

Altera SoC FPGA можно разделить на два блока – систему аппаратного процессора (HPS) и ПЛИС (FPGA). Блок HPS состоит из подсистемы 32-битного процессора с одним или двумя ядрами ARM Cortex A9 MPCore, контроллеров флэш памяти, подсистемы контроллеров SDRAM L3 Interconnect, поддержки периферийных устройств, поддержки отладчика, менеджера FPGA. Двух-процессорный HPS поддерживает симметричный и ассиметричный режим работы [2]. Блок FPGA состоит из матрицы FPGA, блока контроля и интерфейса ввода – вывода. В зависимости от вариантов реализации Altera SoC FPGA в состав блока FPGA могут быть включены трансиверы высокоскоростных устройств, аппаратный контроллер PCI Express и аппаратный контроллер памяти [3]. Интерфейс HPS-FPGA поддерживает двунаправленный обмен между HPS и FPGA и позволяет управлять как из FPGA ведомым устройствам в HPS, так и наоборот.

На рис. 4 приведена подсистема процессора ARM Cortex A9 в двуядерном исполнении.

Подсистема процессора обеспечивает следующие особенности:

- одно или два ядра Cortex-A9;
- контроллер прерываний;
- интервальный и сторожевой таймер;
- глобальный таймер;
- расширения системы защиты TrustZone;
- симметричный и асимметричный режимы работы;
- модули отладки.

Подсистема процессора содержит в себе следующие аппаратные блоки:

- процессор ARM Cortex-A9 MPCore;
- контроллер кэша ARM L2C-310;
- систему управления связями;
- систему отладки и трассировки.

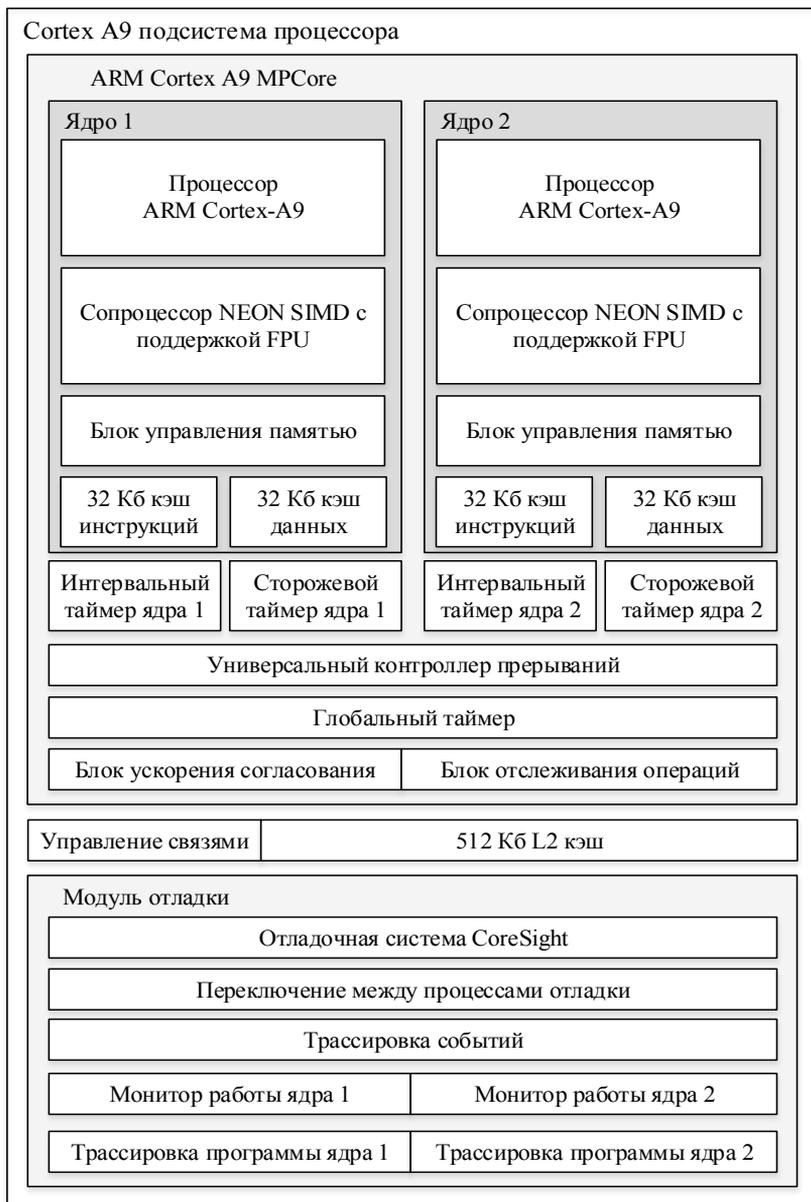


Рис. 5. Блок-схема подсистемы процессора Cortex A9

Стоит также отметить возможность загрузки операционных систем, поддерживаемых ARM Cortex-A9. Например – Altera SoC Yocto проект с пакетом поддержки SoC Linux Board Support Package. После загрузки операционной системы образ проекта может быть загружен в FPGA под контролем системы [4], также имеется возможность производить обновление образа проекта без перезагрузки всей системы [5]. Стандартные способы загрузки приведены на рис. 6.



Рис. 6. Стандартные процедуры загрузки SoC

По адресу сброса обычно выставлен адрес ПЗУ загрузчика, который может быть изменен на адрес внутрикристалльной памяти HPS или внешнего ОЗУ. Код ПЗУ инициализирует HPS, проверяет состояние выбора типа загрузки и, в случае загрузки с флэш-памяти, проверяет и загружает прелоадер в внутрикристалльную память HPS с последующей передачей управления. Возможны следующие варианты используемой памяти: NAND флэш-память, SD/MMC флэш-память, SPI/QSPI флэш-память.

Функции прелоадера определяются пользователем. Обычно это инициализация интерфейса ОЗУ, конфигурация ввода/вывода HPS и считывание загрузчика операционной системы с последующей передачей управления для дальнейшей работы. Размер прелоадера не должен превышать 60 КВ. Этот размер обусловлен размером внутрикристалльной памяти HPS в 64 КВ, из которых 4 КВ зарезервировано под стек и данные.

Для отладки взаимодействия загруженного проекта и процессора используется основанный на ARM Development Studio 5 (DS-5) пакет ARM DS-5 Altera Edition Toolkit, входящий в состав SoC Embedded Design Suite [6], который снимает ограничения для отладки взаимодействия между процессором и FPGA не только на уровне аппаратуры, но и на уровне разрабатываемых драйверов устройства. Пример рабочей области DS-5 приведен на рис. 7.

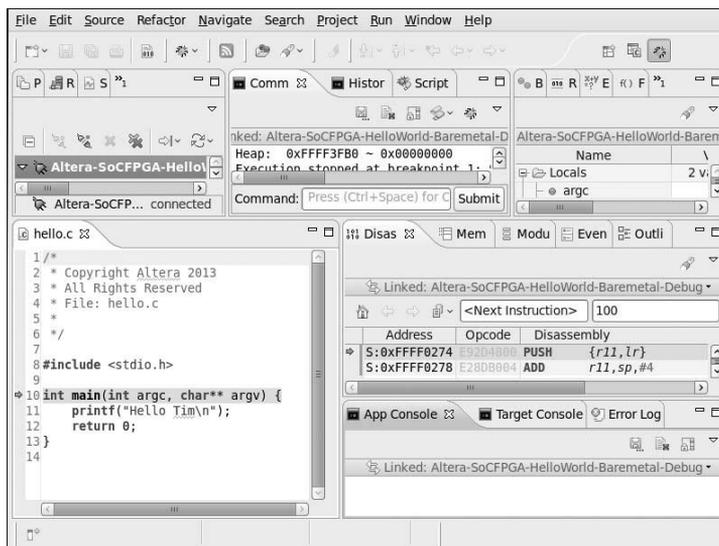


Рис. 7. Рабочая область ARM DS-5 Altera Edition

Среда разработки ARM DS-5 предлагает отладочные возможности для многоядерной архитектуры ARM. ARM DS-5 поддерживает отладку систем, работающих в асимметричной и симметричной многоядерных конфигурациях и может использоваться при написании программ начальной загрузки, разработки драйверов, портировании между операционными системами, разработку гипервизорных и Linux приложений с поддержкой JTAG и Ethernet отладочных интерфейсов, операционных систем Linux и систем реального времени (RTOS).

Среда разработки ARM DS-5 Altera Edition обеспечивает следующие возможности [6] [7]:

- программную отладку подключенных устройств, созданных в FPGA разработчиком, обеспечивая просмотр состояния программных и аппаратных регистров всего пространства SoC, как приведено на рис. 8;

- отладчик одновременно отображает данные отладки и трассировки процессорных ядер Cortex-A9 и пользовательских логических ядер, реализованных в FPGA;

- позволяет производить захват и визуализацию сигналов событий в FPGA, которые могут быть синхронизированы с программными событиями и трассировкой команд процессора, как приведено на рис. 9;

- захват сигналов в логике FPGA может быть осуществлен синхронно с программными или аппаратными событиями процессора и наоборот;

- анализатор производительности, приведенный на рис. 10, позволяет коррелировать потоки и информацию о событиях с аппаратными счетчиками SoC, давая возможность выявлять и исправлять узкие места в пропускной способности на уровне системы.

Наличие процессора с поддержкой периферийных устройств, памяти и возможностью интерактивной отладки позволяет обойти необходимость разработки управляющего процессора и использовать под разрабатываемый проект весь объем матрицы FPGA. FPGA – адаптивная возможность отладки повышает производительность и предоставляет высокий уровень визуализации и контроля над всем устройством в целом.

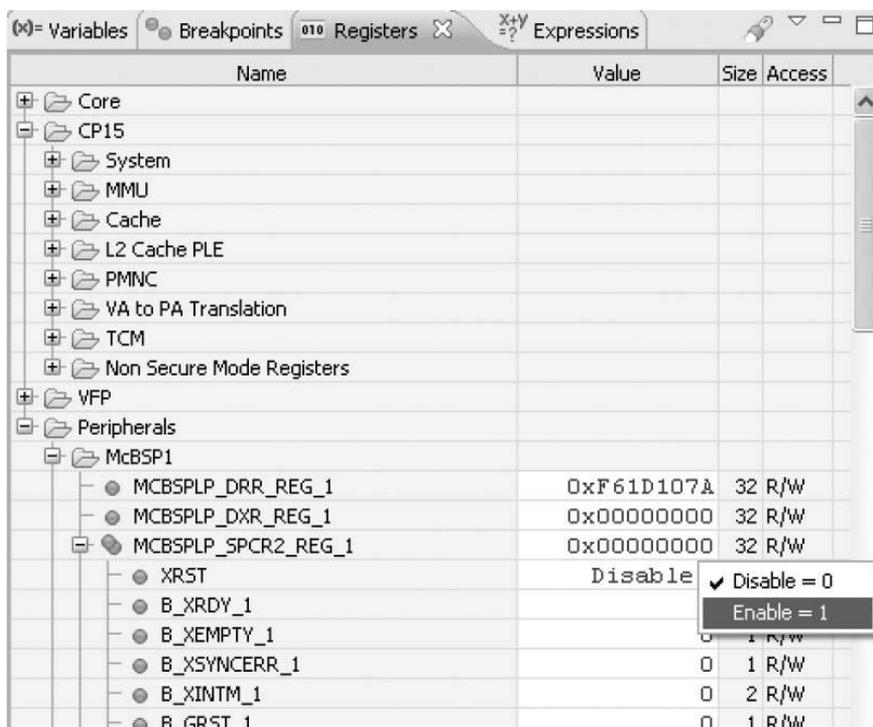


Рис. 8. Состояние регистров

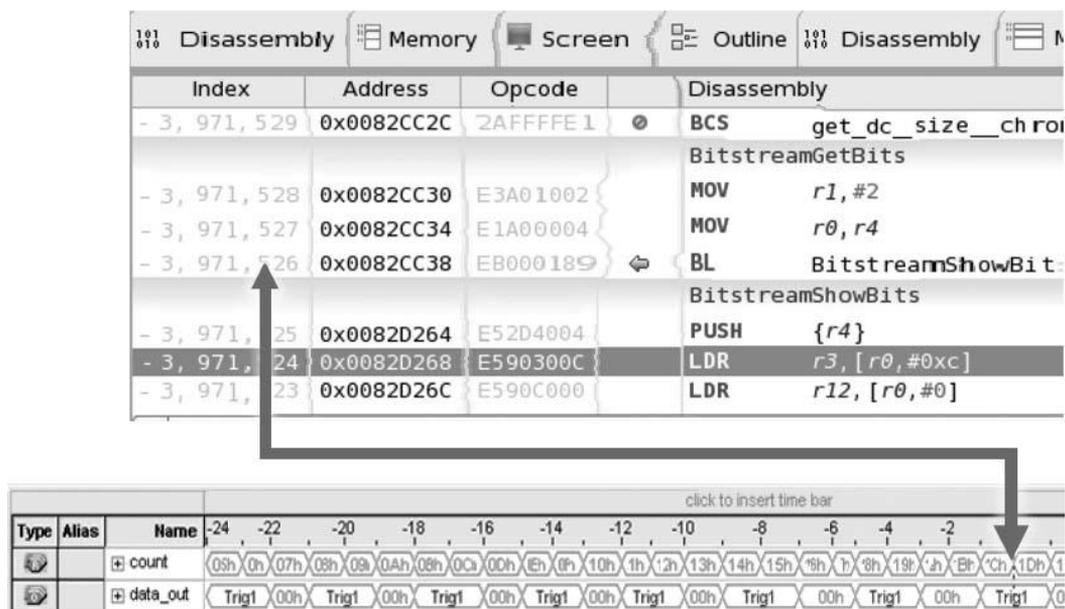


Рис. 9. Синхронизация между процессором и FPGA



Рис. 10. Анализ производительности

## Литература

1. Nios II Processor Reference Handbook. [Electronic resource]. Mode of access: [http://www.altera.com/literature/hb/nios2/n2cpu\\_nii5v1.pdf](http://www.altera.com/literature/hb/nios2/n2cpu_nii5v1.pdf).
2. Hard Processor System Technical Reference Manual. [Electronic resource]. Mode of access: [http://www.altera.com/literature/hb/cyclone-v/cv\\_5v4.pdf](http://www.altera.com/literature/hb/cyclone-v/cv_5v4.pdf).
3. Cyclone V Device Overview. [Electronic resource]. Mode of access: [http://www.altera.com/literature/hb/cyclone-v/cv\\_51001.pdf](http://www.altera.com/literature/hb/cyclone-v/cv_51001.pdf).
4. Booting and Configuration Introduction. [Electronic resource]. Mode of access: [http://www.altera.com/literature/hb/cyclone-v/cv\\_5400A.pdf](http://www.altera.com/literature/hb/cyclone-v/cv_5400A.pdf).
5. Configuration, Design Security, and Remote System Upgrades in Cyclone V Devices. [Electronic resource]. Mode of access: [http://www.altera.com/literature/hb/cyclone-v/cv\\_52007.pdf](http://www.altera.com/literature/hb/cyclone-v/cv_52007.pdf).
6. Altera SoC Embedded Design Suite User Guide. [Electronic resource]. Mode of access: [http://www.altera.com/literature/ug/ug\\_soc\\_edu.pdf](http://www.altera.com/literature/ug/ug_soc_edu.pdf).
7. FPGA-Adaptive Software Debug and Performance Analysis. [Electronic resource]. Mode of access: <http://www.altera.com/literature/wp/wp-01198-fpga-software-debug-soc.pdf>.

## РЕАЛИЗАЦИЯ АЛГОРИТМА КАСКАДНОГО СБОРА ГЛОБАЛЬНОГО УРОВНЯ В МНОГОСЕТОЧНОМ РЕШАТЕЛЕ ПАКЕТА ПРОГРАММ ЛОГОС

*А. В. Ялозо, А. С. Козелков, Д. П. Силаев, С. В. Лашкин*

Российский федеральный ядерный центр –  
Всероссийский НИИ экспериментальной физики, г. Саров

Неявная дискретизация основной системы вычислительной гидродинамики – системы уравнений Навье–Стокса, порождает систему разностных уравнений, которая в основном решается итерационными методами [1]. При этом классические итерационные методы либо перестают работать, либо дают очень медленную скорость сходимости [2]. Одним из методов решения данной проблемы является использование алгебраического многосеточного метода, основанного на использовании последовательности вложенных сеток и операторов перехода от одной сетки к другой [3, 4].

В пакете программ ЛОГОС, предназначенном для расчета задач гидродинамики, аэродинамики, теплопереноса и распространения тепла в твердотельных конструкциях, при решении СЛАУ используется собственная реализация многосеточного метода [5]. Разработанный ранее в рамках данного решателя последовательный алгоритм сбора матрицы глобального уровня позволил получить значительное ускорение решения СЛАУ, но в то же время данная реализация имеет существенные ограничения на максимально возможный размер матрицы СЛАУ: суммарный объем объединяемых уровней не должен превышать объема памяти одного узла вычислительного кластера, на котором производится расчет. Кроме того, ввиду роста размерности задач, и соответственно, роста числа узлов, требуемых для решения задачи, растет и время, затрачиваемое на построение, и последующее огрубление глобального уровня в скалярном режиме.

На рис. 1 схематично представлена скалярная реализация алгоритма формирования глобального уровня для задачи, выполняемой на  $n$  процессорах.