



УДК 004.272.23:004.274

**С.Я. Гильгурт**, канд. техн. наук

Ин-т проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины  
(Украина, 03164 Киев, ул. Генерала Наумова, 15,  
тел.(044) 4249163, e-mail: hilgurt@ukr.net)

## **Реконфигурируемые вычислители. Аналитический обзор**

Проанализирован широкий круг вопросов, связанных с перспективным классом высокопроизводительных цифровых устройств — реконфигурируемыми вычислителями на основе программируемых логических интегральных схем (ПЛИС). Проведен сравнительный анализ с другими классами присоединенных вычислителей, предназначенных для ускорения решения ресурсоемких задач на персональных компьютерах и в составе кластерных систем. Исследованы сложности, препятствующие распространению реконфигурируемых вычислителей, проанализированы пути их преодоления.

Проаналізовано широке коло питань, пов'язаних із перспективним класом високопродуктивних цифрових пристроїв — реконфігуровними обчислювачами на базі програмуваних логічних інтегральних схем. Проведено порівняльний аналіз з іншими класами приєднаних обчислювачів, призначених для прискорення розв'язування ресурсомістких задач на персональних комп'ютерах і в складі кластерних систем. Досліджено складності, що перешкоджають розповсюдженню реконфігуровних обчислювачів, проаналізовано шляхи їх вирішення.

*К л ю ч е в ы е с л о в а: ПЛИС, реконфигурируемый унифицированный вычислитель, сопроцессор, ускоритель.*

В настоящее время при проведении инженерных и научных расчетов в различных отраслях народного хозяйства, в том числе, в энергетике, проблема повышения производительности становится все более актуальной. Высокая стоимость и ограниченные возможности масштабирования традиционных суперкомпьютеров привели к широкому распространению кластерных решений, позволяющих в результате применения стандартизированных компонентов и технологий существенно снизить стоимость владения вычислительным оборудованием.

Однако, в силу архитектурных особенностей, данное направление не покрывает весь спектр вычислительных задач, с которыми сталкиваются

© С.Я. Гильгурт, 2013

исследователи и производственники. По этой причине возникло и активно развивается направление, способное дополнить и расширить возможности кластеров. Это направление связано с использованием присоединенных вычислителей (сопроцессоров), которыми оснащаются узлы высокопроизводительных ЭВМ, а также системные блоки персональных компьютеров. Принципы построения таких устройств и их возможности различны. Проанализируем наиболее перспективный класс подобных ускорителей — вычислители на основе программируемой логики.

Существует большое число публикаций на эту тему [1—32], однако в них рассмотрены преимущественно конкретные разработки или их применение. Отсутствуют аналитические обзоры, отличающиеся системностью и полнотой исследуемых вопросов. Кроме того, реконфигурируемые сопроцессоры интенсивно развиваются, поэтому опубликованные сведения быстро теряют актуальность.

Предлагается аналитический обзор современного состояния исследуемого направления, а также анализ экономических предпосылок и ретроспективы для выявления закономерностей, систематизации накопленных сведений, более глубокого понимания проблемы и формирования предположений о дальнейшем развитии.

**Экономические факторы** всегда оказывали существенное влияние на процессы развития вычислительной техники. Сложность компьютерного оборудования приводит к необходимости при оценке эффективности финансовых вложений учитывать полную стоимость владения, т.е. затраты на разработку и изготовление аппаратуры, а также на обслуживание, создание и использование программного обеспечения всех уровней. Быстрое изменение цен на компьютерные комплектующие требует учитывать и фактор морального старения.

Интерес к присоединенным вычислителям также в значительной степени обусловлен экономическими причинами. Рассмотрим основные способы снижения стоимости владения применительно к сопроцессорам.

**Факторы снижения стоимости. Стандартизация и унификация.** Одним из основных факторов, позволяющих снизить стоимость владения вычислительными средствами, является стандартизация и унификация на всех уровнях (аппаратном, системном, программном) и, как следствие, переход к массовому изготовлению однотипных изделий.

Примером того, как открытость, унификация и совместимость могут способствовать широкому распространению и проникновению вычислительных средств во многие сферы деятельности человека, может служить история развития персональных компьютеров. Их массовое производство началось в конце 70-х годов прошлого века после беспрецедентного шага

компании IBM, опубликовавшей технические спецификации своих изделий. Данное решение позволило многочисленным сторонним разработчикам начать выпуск так называемых IBM-совместимых персональных ЭВМ (ПЭВМ) и в сжатые сроки сформировать рынок нового класса вычислительной техники. Увеличение объемов производства привело к снижению стоимости и повышению спроса, что, в свою очередь, лавинообразно повысило спрос и позволило еще больше снизить цены. В результате и сторонние разработчики, и фирма — создатель стандарта де-факто получили большую экономическую выгоду. Рынок ПЭВМ до сих пор значителен и в 2013 г. составил приблизительно 200 млрд. долларов, несмотря на широкое распространение в последнее время переносных и мобильных компьютеров [33].

Разработка унифицированных изделий и стандартов в области присоединенных вычислителей, в частности реконфигурируемых, также способно привести к существенному снижению стоимости и спровоцировать их массовое распространение, как это произошло с ПЭВМ [27—30].

*Использование имеющихся наработок и технологий.* Если при разработке некоторой проблемы имеется возможность применить уже существующее техническое решение, такой подход оказывается более предпочтительным по сравнению с новой разработкой. История развития сопроцессоров подтверждает данный тезис.

За время существования универсальных компьютерных систем на базе микропроцессоров накоплен колоссальный объем программного обеспечения (ПО) от прикладного уровня до операционных систем, создано и отработано множество технических решений, стандартов и протоколов. Именно по этой причине большинство новых высокопроизводительных компьютерных технологий реализуется в последнее время в виде всевозможных ускорителей, акселераторов, сопроцессоров, т.е. устройств, взаимодействующих с традиционными микропроцессорными системами.

Не является исключением и такое перспективное направление в суперкомпьютеринге, как реконфигурируемые вычисления на базе ПЛИС (Reconfigurable Computing). Анализ публикаций за последние 8—10 лет свидетельствует о том, что в деятельности ведущих разработчиков, а также научных школ как в Украине, так и в мире в целом, наблюдается смещение интереса от автономных, полностью реконфигурируемых систем, подобных, например, так называемым гиперкомпьютерам (Hypercomputers) фирмы Star Bridge Systems (США) [34], в сторону присоединенных решений, в которых реконфигурируемое оборудование функционирует в тесном взаимодействии с микропроцессорной техникой [13, 15]. Показа-

тельным в этом плане представляется появление ускорителей для ПЭВМ РУПК-25 и РУПК-50 в числе разработок НИИ многопроцессорных вычислительных систем Южного федерального университета (Россия) — организации, ориентированной на создание тесно связанных структур из большого числа ПЛИС [8, 7].

*Разделение труда разработчиков и производителей аппаратуры и ПО.* Известно, что специализация в узкой области дает возможность повысить профессионализм и качество труда. В области производства ПЭВМ, а также сопутствующих им товаров и услуг можно наблюдать убедительные примеры преимуществ такого подхода. Существуют, например, крупные корпорации мирового уровня, специализирующиеся только на системах охлаждения для компонентов персональных компьютеров. На рынке программного обеспечения разделение труда выражено еще более явно. Следует заметить, что рассмотренный выше фактор стандартизации и унификации является ключевым для обеспечения положительного эффекта от разделения труда.

**Кластеры.** Перечисленные способы снижения затрат на разработку и эксплуатацию в полной мере реализуются при кластерном подходе. В соответствии с определением компании DEC кластер — это группа вычислительных машин, которые связаны между собой и функционируют как один узел обработки информации [35]. При переходе от традиционных суперЭВМ к кластерным вычислениям существенно удешевляются процессы проектирования и разработки узлов и блоков, средств обмена данными, создание системного ПО. Стандартизация на многих уровнях, вплоть до прикладного ПО, делает возможным разделение труда программистов, решающих вопросы межпроцессорного обмена, разрабатывающих инструментальные средства, универсальные и специализированные пакеты программ.

Можно выделить ряд задач, при решении которых применение кластерных технологий дает возможность получить результат на порядок быстрее, чем при использовании рабочих станций. К ним относятся, например, исследование генома человека, биоинформатика, биохимия, биофизика, моделирование погоды, анализ фондового рынка, а также исследование прочности, теплофизика, динамика жидкостей и газов, электромагнетизм и др. Тем не менее, эффективность вычислительного кластера как любой жесткой архитектуры существенно зависит от природы решаемой задачи. Но для таких задач характерна слабая связность вычислительных алгоритмов, вследствие чего относительно медленный обмен данными между узлами, обрабатывающими информацию, не приводит к значительному замедлению вычислений.

При попытке решать на кластере задачи, более тесно связанные по своей природе, значительно возрастает нагрузка на интерконнект. В результате эффективность расчетов резко снижается — с увеличением числа задействованных узлов уменьшается не только удельная загрузка процессорных модулей, но и замедляется решение задачи в целом [9]. Следовательно, для задач с такими свойствами, используя кластерную архитектуру, в принципе невозможно достигнуть существенного ускорения расчетов.

Именно по этой причине актуальным является использование присоединенных вычислителей, основанных на инновационных компьютерных технологиях и обладающих сильной связностью внутренней структуры.

**Сопроцессоры.** В течение длительного времени единственной возможностью быстро решать тесно связанные ресурсоемкие задачи без применения классических суперкомпьютеров было применение специализированных вычислителей или спецпроцессоров. Специализированное устройство позволяет максимально учитывать особенность выполняемого алгоритма, обеспечивая требуемое быстродействие для заданной задачи или узкого класса задач, однако область его применения, как правило, этим классом задач ограничивается. Необходимость для каждой вновь возникающей задачи разрабатывать, отлаживать и запускать в производство такое устройство, а также создавать для него специальное ПО делает данный подход нецелесообразным в современных условиях. Уникальность специализированных вычислителей затрудняет их массовое применение.

По этой причине возникло и активно развивается направление, связанное с применением различного рода присоединенных вычислителей и ускорителей, обладающих более высокой гибкостью и универсальностью по сравнению со спецпроцессорами. Фактически появился новый класс вычислительных устройств, рассчитанных на решение относительно широкого круга задач, в которых различными способами решается проблема узкой специализации. В связи с актуальностью вопросов стандартизации подобные устройства названы унифицированными вычислителями (УВ) [26].

**Унифицированные вычислители.** Проанализируем главное концептуальное противоречие, которое должны разрешить УВ.

**Тройное противоречие.** Строго говоря, противоречие между высокой производительностью и универсальностью вычислений не является принципиальным и носит преимущественно экономический характер. Методологически возможно для решения каждой вновь возникающей вычислительной задачи создавать новое компьютерное средство, учитывающее ее внутреннюю природу и обеспечивающее максимально возможное быстродействие, в принципе достижимое при использовании имеющихся тех-

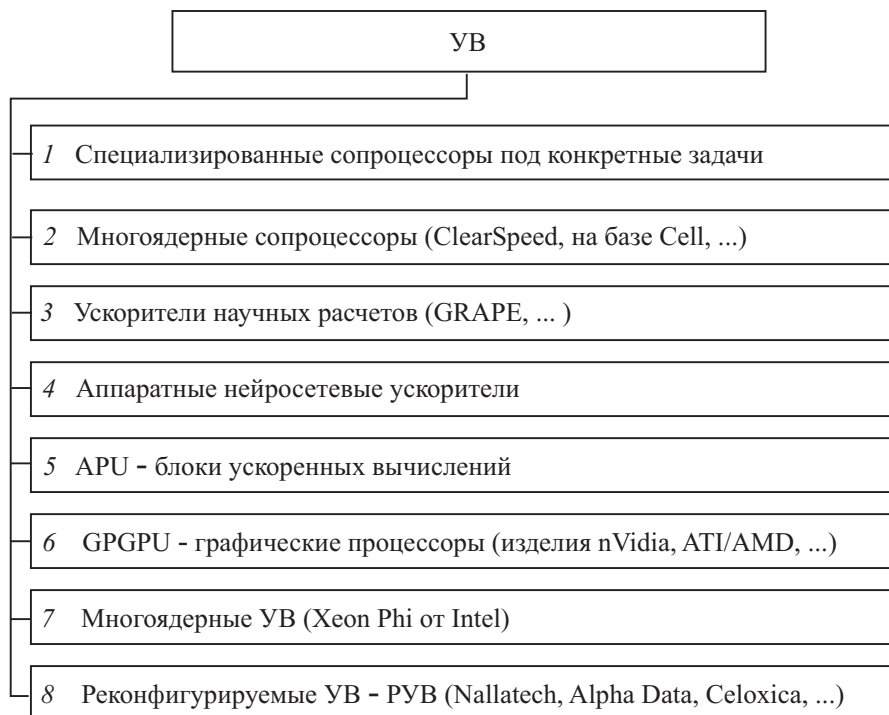


Рис. 1. Виды УВ

нологий. Однако стоимость такого подхода делает его неприменимым на практике. Поэтому более правильно говорить о тройном противоречии: производительность—универсальность—затраты.

Суть этого противоречия заключается в следующем. С одной стороны, каждой из имеющихся задач, в силу их различной природы, соответствует собственная вычислительная архитектура, с помощью которой данная задача решается наиболее эффективно. С другой стороны, создание гибкого и универсального вычислительного средства, способного идеально подстраивать свою внутреннюю структуру под решаемую задачу, приводит к существенным материальным затратам, поскольку конфликтует с упомянутыми факторами снижения стоимости владения, такими как унификация, совместимость и использование имеющихся наработок.

Рассмотрим, каким образом данное противоречие преодолевается в известных решениях (независимо от их популярности в настоящее время). На рис. 1 приведены группы известных разработок, которые в той или иной степени можно отнести к классу УВ, упорядоченные по степени разрешения сформулированного тройного противоречия. Специализиро-

ванные сопроцессоры 1 обладают малой универсальностью. Строго говоря, они не являются УВ и приведены на рис. 1 в целях обобщения.

Вычислители, входящие в группу 2, подробно исследованы в литературе [3, 13]. Они универсальны, но по структуре аналогичны традиционным многопроцессорным суперкомпьютерам, наследуя их недостатки и противоречия. Отличие заключается лишь в производительности и организации вычислительного процесса.

Группа разработок 3 позволяет несколько расширить функциональность спецпроцессоров, ориентируя их на алгоритмы, используемые в нескольких смежных областях. Например в Японии разработано семейство сопроцессоров GRAPE (GRAvity PipE), предназначенных для ускоренного решения гравитационной задачи  $N$  тел [36].

Круг проблем, эффективно решаемых на нейросетевых ускорителях 4, еще шире, но и он недостаточно широк, чтобы назвать данные устройства универсальными [37].

При построении следующих групп УВ использованы определенные идеи и технические решения, направленные на преодоление сформулированного тройного противоречия.

Анализируя имеющиеся сведения, можно выделить четыре подхода к построению УВ и определить четыре способа устранения указанного противоречия.

С п о с о б 1 основан на применении единого изделия, например в виде сверхбольшой интегральной схемы (СБИС), которое содержит в себе несколько различных специализированных вычислительных структур, оптимизированных под наиболее распространенные задачи. Предполагается, что такой набор будет эффективен для большого числа вычислительных алгоритмов. Избыточность и простои незадействованных ресурсов компенсируются низкой стоимостью таких изделий, их широкой распространенностью и стандартизацией программного обеспечения. Сюда можно отнести так называемые блоки ускоренных вычислений APU (Accelerated Processing Units), предложенные фирмой AMD [38]. К сожалению, идея не нашла практического применения. Термин APU приобрел иной смысл и означает теперь совмещение на одном кристалле традиционных микропроцессорных и графических ядер [39]. Данный способ рассмотрен для получения более полной картины.

С п о с о б 2 ориентирован на нестандартное использование в качестве математического ускорителя какого-либо из уже существующих компонентов вычислительной системы. В этом случае главное — низкая цена и распространенность задействованного компонента, что существенно снижает стоимость владения техническим решением в целом.

Примером реализации данного способа является использование видеоадаптеров персональных компьютеров для решения ресурсоемких задач. Технология называется GPGPU (General-Purpose Computation on Graphics Processing Units) или вычисления на графических процессорных устройствах (ГПУ). Высокая вычислительная сложность задач обработки видеоинформации, с одной стороны, и острая конкуренция производителей видеоадаптеров — с другой, привели к тому, что современное ГПУ при относительно невысокой стоимости содержит большое число арифметических блоков упрощенной архитектуры, являясь при этом стандартным компонентом универсального компьютера. Более подробно технология GPGPU рассмотрена, например, в [40].

С п о с о б 3 основан на идее, аналогичной GPGPU, но главная СБИС сопроцессора изначально разрабатывается как основа для ресурсоемких расчетов, сохраняя возможность обработки графических данных. Главное его отличие заключается в том, что вычислительные элементы имеют архитектуру, совместимую с распространенными процессорами общего назначения (так называемыми x86-совместимыми процессорами), для которых создано и освоено большое число программных разработок.

Преимуществом данного подхода является использование унифицированного ПО. При этом основная задача для многоядерных УВ состоит в том, чтобы, с одной стороны, не оказаться в проигрыше по стоимости сопроцессоров, а с другой — оставаться не менее стандартными компьютерными компонентами, чем графические адаптеры. Фирма Intel, являющаяся апологетом данного направления, постоянно совершенствует свои изделия. Последняя ее разработка данного класса продвигается на рынок под названием Xeon Phi Coprocessor [41].

С п о с о б 4 подразумевает использование реконфигурируемых УВ (РУВ), построенных на базе ПЛИС типа FPGA, известных также как Reconfigurable Coprocessors. При разработке и сопровождении РУВ предполагается максимальное использование стандартизированных технических решений и программных наработок [27— 30].

Реконфигурируемые вычислители позволяют задействовать мощный потенциал программируемой логики. Современные ПЛИС содержат миллионы эквивалентных логических элементов и позволяют синтезировать в себе произвольную цифровую вычислительную структуру высокой сложности, способную функционировать на частотах в сотни и тысячи мегагерц, которая теоретически может быть оптимизирована для решения произвольной вычислительной задачи [15]. Первое место в списке самых производительных ЭВМ мира TOP500 в ноябре 2009 г. и июне 2010 г. занял суперкомпьютер Jaguar — Cray XT5-HE (пиковая производительность 2331 Терафлоп) производства фирмы Cray Inc., в котором приме-



Номер шл.	УВ	Техническое решение	Гибкость	Стоимость		Разработка конфигу- раций	Наличие стандартов	Наличие систем- ного ПО	Сложность адаптации приклад- ного ПО	Доступ- ность для массового пользо- вателя	Энерго- потре- ление
				компонентов вычислителя	разработки вычислителя						
1.	APU	Комбинация нескольких вычислитель- ных структур	Средняя	Высокая	Высокая	Не требуется	Отсутст- вуют	Отсутст- вует	Высокая	Низкая	Среднее
2.	ГПУ/ GPGPU	Нестандартное использование имеющихся компонентов	Невысокая	Средняя	Низкая	Не требуется	Частично имеются	Частично имеется	Средняя	Высокая	Очень высокое
3.	Много- ядерные x86-сов- местимые	Использова- ние стандарт- ных элемен- тарных про- цессорных элементов	Средняя	Высокая	Низкая	Не требуется	Частично имеются	Частично имеется	Средняя	Средняя	Очень высокое
4.	РУВ	Применение ПЛИС в сос- таве типового изделия	Высокая	Средняя	Низкая	Требуется	Отсутст- вуют	Частично имеется	Средняя	Средняя	Низкое

нены реконфигурируемые вычислители на базе ПЛИС высокой степени интеграции [42, 9]. В настоящее время различными фирмами производится большое число изделий, которые могут быть использованы в качестве РУВ [43].

**Сравнительный анализ.** Рассмотрим, какие преимущества и недостатки имеют описанные способы создания УВ. В таблице приведены некоторые качественные оценки каждого из рассмотренных способов.

Главными недостатками УВ первого типа являются относительно невысокая гибкость и высокая стоимость. Этот подход, к сожалению, не устраняет противоречия между высоким быстродействием, универсальностью и стоимостью.

Второму типу УВ свойственны недостатки, присущие всем вычислительным устройствам с жесткой архитектурой. Фиксированный набор инструкций, однозначно заданная связь между процессорными элементами, а также высокое энергопотребление и тепловыделение ограничивают область возможного применения, затрудняя их интеграцию в системный блок универсального компьютера либо в вычислительный узел кластера.

К недостаткам данного способа следует также отнести необходимость понимания программистом архитектурных особенностей графических адаптеров. Правильная организация работы с памятью ГПУ существенно влияет на эффективность решения вычислительных задач [44]. Тем не менее, использование широко распространенных компонентов, хорошая проработанность вопросов стандартизации и наличие удобного инструментального ПО делают данный подход наиболее перспективным в ближайшие несколько лет.

Третий тип УВ тесно связан со вторым. В случае наследования их положительных качеств, а также с учетом повышенных возможностей стандартизации ПО данные УВ могут превзойти устройства второго типа по эксплуатационным показателям. Негативным фактором, способным помешать широкому распространению ускорителей класса Xeon Phi Coprocessor, является ориентация на одного производителя.

В наибольшей степени разрешить сформулированное выше концептуальное противоречие способны УВ четвертого типа, т.е. РУВ. Отсутствие у них большинства недостатков, свойственных УВ другого типа, к сожалению, компенсируется рядом сложностей и технических проблем, возникающих при их практическом использовании. Поэтому вычислители на базе ПЛИС пока еще менее распространены, чем устройства GPGPU. От того, насколько быстро и результативно удастся преодолеть данные сложности, зависит успешное освоение и массовое распространение этих перспективных устройств.

Для более глубокого понимания проблемы рассмотрим ряд положений общего плана, касающихся различных аспектов разработки и применения РУВ.

**Реконфигурируемые УВ. Общие вопросы.** Технические показатели РУВ определяются преимущественно параметрами ПЛИС, входящей в его состав. Проанализируем особенности и возможности современной программируемой логики, используемой при создании РУВ.

#### **Структура и состав ПЛИС.**

Первые ПЛИС — программируемые логические матрицы (ПЛИМ) — появились более 35 лет назад для снижения расходов на разработку электронных устройств, изготавливаемых относительно небольшим тиражом. Широко используемые тогда ПЛИМ представляли собой матрицы логических элементов «и», «или» и инверторов. Связи между элементами программировались посредством пережигания плавких перемычек, как в постоянных запоминающих устройствах. В результате ПЛИМ превращалась в комбинационную схему, выполняющую необходимую логическую функцию [1, 5].

В дальнейшем для хранения конфигурации стали применять энергонезависимую электрически перепрограммируемую флэш-память. В настоящее время такая память в основном используется в ПЛИС, предназначенных для создания сложных комбинационных схем. В зарубежной литературе этот класс изделий известен как CPLD (Complex Programmable Logic Device). Но для синтеза сложных вычислительных схем, содержащих большое число триггеров и регистров, применяют ПЛИС другого класса, в которых конфигурация хранится в статической памяти. В англоязычной литературе эти изделия называют FPGA (Field- Programmable Gate Array) [16]. Основные производители микросхем данного класса — фирмы Xilinx [45] и Altera [46].

Типичная микросхема FPGA состоит из массива конфигурируемых логических блоков (КЛБ) и коммутационной матрицы (рис. 2). Вокруг КЛБ размещены блоки ввода-вывода (БВВ). Программируемые соединения обеспечивают все связи внутри микросхемы, а БВВ — интерфейс между ее внутренними компонентами и внешними контактами [18].

В простейшем случае логический блок содержит четырехходовую таблицу преобразований LUT (lookup table) и D-триггер. Но в большинст-

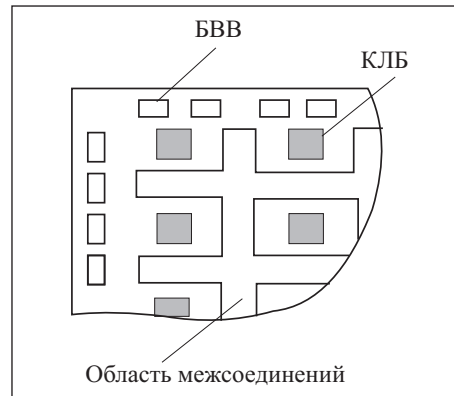


Рис. 2. Обобщенная структура ПЛИС типа FPGA

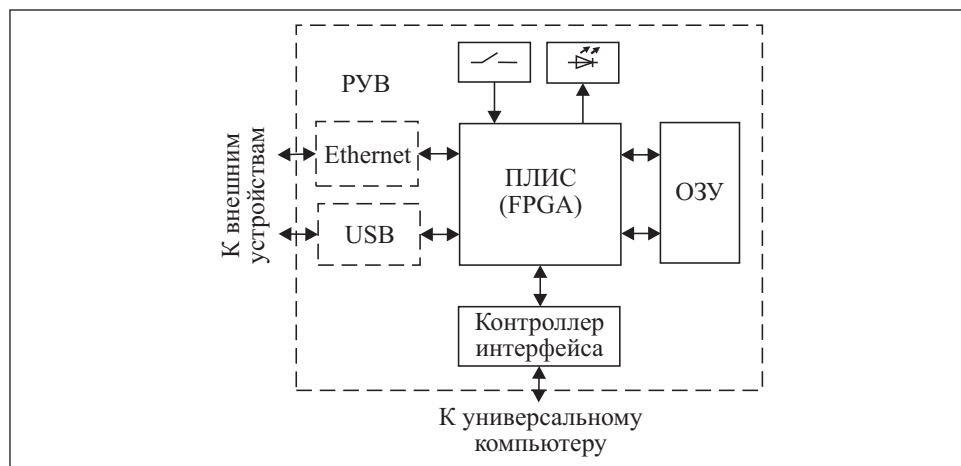


Рис. 3. Структурная схема РУВ

в современных ПЛИС типа FPGA используются блоки более сложной структуры, включающие шести-либо восьмивходовые таблицы преобразований, несколько триггеров и схемы ускоренного переноса [18]. На таких микросхемах можно синтезировать цифровые устройства высокой сложности. Уже производят кристаллы, содержащие миллионы эквивалентных логических элементов. При этом стоимость ресурсов ПЛИС в пересчете на один логический элемент постоянно снижается.

Последние семейства ПЛИС типа FPGA изготавливаются по 28-нанометровым проектным нормам и содержат, кроме упомянутых компонентов, такие аппаратно реализованные устройства:

- RISC-процессоры, совместимые по архитектуре с PowerPC или ARM;
- блоки цифровой обработки сигналов, производительностью до 10 Терафлоп;
- контроллеры памяти DDR3 с быстродействием до 1,866 Гбайт/с;
- восьмиканальные модули системной шины PCI Express 3.0;
- контроллеры физического уровня Ethernet 10/100/1000 Мбит/с;
- дифференциальные приемо-передающие модули со скоростью обмена до 28 Гбит/с;
- блоки мониторинга напряжений питания и температуры внутри кристалла [45, 46].

Для защиты проекта от копирования используется шифрование конфигурационной последовательности по алгоритму AES с 256-битным ключом.

**Структура и состав РУВ.** В работе [22] впервые обоснованы структура и состав РУВ (рис. 3). Его основа — интегральная схема програм-

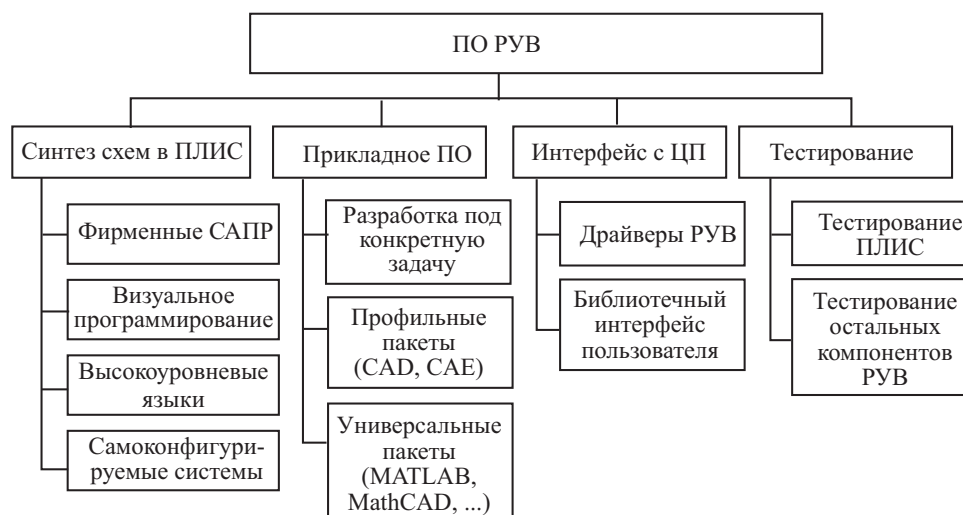


Рис. 4. Программное обеспечение для РУВ

мируемой логики. Обязательным требованием является наличие выделенного контроллера интерфейса, на который помимо обмена данными с центральным процессором (ЦП) возлагается также функция конфигурирования ПЛИС. Данное требование обусловлено энергозависимостью конфигурации, хранящейся в микросхеме ПЛИС типа FPGA. Вследствие этого становится неприемлемым решение, при котором контроллер интерфейса синтезируется внутри ПЛИС, занимая часть ее ресурсов, так как для инициализации реконфигурируемой системы после подачи питания потребовалось бы задействовать дополнительные средства программирования [5].

Необходимыми компонентами РУВ являются двухпортовое оперативное запоминающее устройство (ОЗУ) для хранения промежуточных результатов, а также отладочные индикаторы и переключатели. Конструктивно РУВ выполняется либо в виде платы расширения, либо в формате другого стандартного компонента ПЭВМ в зависимости от используемого интерфейса.

Кроме обязательных узлов в состав РУВ могут входить некоторые дополнительные компоненты, позволяющие без существенного удорожания значительно расширить его функциональность как массового изделия. Это могут быть сетевые порты Ethernet, контроллеры периферийной шины USB и др.

Наиболее распространен в настоящее время интерфейс обмена с РУВ по системной шине PCI, PCI-X, либо PCI Express, что обусловлено наилучшим соотношением между стоимостью и производительностью. Относи-

тельно новый и интересный вариант обмена — использование в качестве интерфейса шин межпроцессорного обмена QPI [47] или Hypertransport [48], при разработке которых изначально закладывалась возможность взаимодействия с РУВ. Эти шины обеспечивают наиболее высокую пропускную способность. В настоящее время производятся многопроцессорные материнские платы, в свободные сокетты которых могут быть вставлены РУВ, конструктивно выполненные в форм-факторе универсального процессора.

Следует заметить, что здесь не рассматривается организация обмена информацией между реконфигурируемым устройством и прочими компонентами вычислительного комплекса, такими как центральный процессор и устройства оперативной памяти. Данные вопросы представляют собой отдельную научную проблему [13, 14, 24, 31].

**Программное обеспечение РУВ.** При практическом использовании РУВ возникает необходимость применения специального ПО. В работах [25, 32] рассмотрена и исследована классификация различных категорий ПО, связанного с разработкой, тестированием, конфигурированием и использованием РУВ. Программное обеспечение для РУВ можно условно разделить на четыре группы (рис. 4):

1. Инструментальные средства разработки вычислительных структур (конфигураций) для загрузки в ПЛИС.
2. Прикладное ПО, в котором РУВ использовано в качестве сопроцессора.
3. Интерфейс взаимодействия РУВ с микропроцессорной системой.
4. Тестирующее ПО.

**Инструментальные средства разработки конфигураций.** Первая группа ПО для РУВ — средства разработки загружаемых в ПЛИС вычислительных структур, закодированных последовательностью битов (bitstream) и организованных в виде так называемых файлов конфигурации. В этой группе, в свою очередь, можно выделить четыре подгруппы:

- 1) фирменные системы автоматизированного проектирования (САПР);
- 2) пакеты визуального программирования;
- 3) модификации известных языков программирования высокого уровня;
- 4) программное обеспечение самоконфигурируемых систем.

К первой подгруппе относятся фирменные пакеты разработки на языках описания оборудования. Примерами такого ПО являются пакеты WebPACK ISE, FOUNDATION SERIES (фирмы Xilinx), MAX+PLUS, QUARTUS (фирмы Altera) и др. Данный вид инструментальных средств позволяет создавать конфигурации, оптимальные по производительности и занимаемой площади кристалла.

Недостатком этой подгруппы является то, что данные пакеты требуют от разработчика высокой квалификации, знаний и навыков в области син-

теза цифровых схем, а также владения языками описания аппаратуры. Кроме того, фирменные САПР привязаны к конкретному семейству ПЛИС, что препятствует стандартизации.

Вторая подгруппа предъявляет менее жесткие требования к разработчикам. В качестве примера ПО данного вида можно привести среду разработки Viva software platform, упомянутой выше фирмы Star Bridge Systems [34, 49]. Такие средства проигрывают в универсальности и ограничивают возможности стандартизации. За последние годы интерес к инструментам данного класса заметно снизился.

Стремление упростить процесс синтеза реконфигурируемых устройств, а также привлечь разработчиков из огромного числа традиционных программистов привели к созданию третьей подгруппы инструментальных средств, к которой относятся модификации известных языков программирования высокого уровня, чаще всего С-подобные. Так, компания Mentor Graphics использует язык IEEE Standard System C, Celoxica — Handle C, Nallatech — DIME C, Impulse — Impulse-C, Mitronics — Mitrion-C. Некоторые разработчики используют стандартную версию языка — ANSI C [10]. Идея создания таких средств заключается в максимальном использовании богатого опыта программирования, накопленного разработчиками традиционного ПО. К сожалению, естественный параллелизм реконфигурируемых устройств, являющихся, по сути, аппаратными схемами, плохо поддается описанию языками, которые изначально предназначались для программирования традиционных процессоров, последовательных по своей сути.

Идея создания самоконфигурируемых систем заключается в полном освобождении разработчика вычислительных средств на базе ПЛИС от вопросов создания конфигураций и их загрузки [12]. Пользователь в этом случае описывает решаемую задачу на высокоуровневом языке, ПО самоконфигурируемой системы в автоматическом режиме распределяет код на две части — для ЦП и для РУВ. Затем компилируется первая часть и создается конфигурация для второй части. По команде оператора система загружает полученный выполняемый код в соответствующее оборудование и запускает его. Данный подход к созданию инструментального ПО вытекает непосредственно из предыдущего, наследуя при этом его недостатки.

*Прикладное ПО, использующее РУВ в качестве сопроцессора.* Прикладное ПО для РУВ создается аналогично обычному, предназначенному для работы с традиционными архитектурами, и отличается лишь наличием возможности обращаться к РУВ как к сопроцессору, чтобы использовать синтезированную в нем вычислительную структуру для выполнения наиболее ресурсоемких вычислительных операций.

Следует заметить, что важным фактором для активного продвижения РУВ как стандартизированных устройств может оказаться поддержка их использования в распространенных пакетах инженерных и научных расчетов универсального назначения (MATLAB, MathCAD и др.), а также специализированных САПР.

Менее исследованы в литературе третья и четвертая группы ПО для РУВ.

*Интерфейс взаимодействия РУВ с микропроцессорной системой.* Эта группа представляет собой библиотеку функций для языка программирования прикладного ПО, а также системные драйверы и несет ответственность за согласование разнородных компонентов вычислительного комплекса, образованного универсальным компьютером и РУВ как сопроцессором.

При использовании РУВ в составе ПЭВМ, информация, передаваемая от ЦП к загруженной в ПЛИС вычислительной структуре и обратно, проходит определенный путь, пролегающий через сферы компетенции специалистов различных профилей: прикладных и системных программистов, разработчиков аппаратуры и загружаемых в РУВ конфигураций. В работе [24] рассмотрены вопросы взаимодействия ПЭВМ и РУВ как обмена данными между прикладной программой, выполняемой в ЦП, и написанной на языке описания аппаратуры вычислительной структурой, загруженной в ПЛИС.

*Тестирующее ПО.* Вопросы диагностирования ПЛИС достаточно освещены в литературе. Однако в состав РУВ входят и другие компоненты, техническое состояние которых также необходимо оценивать. В работе [23] описано создание ПО для диагностирования и оценки эффективности функционирования вспомогательных компонентов РУВ, а также для обеспечения разработчиков необходимой информации о РУВ как о системном устройстве.

Теперь рассмотрим трудности, возникающие при использовании РУВ.

**Основные проблемы и пути их решения.** Анализ информационных источников позволяет выделить следующие факторы, препятствующие широкому распространению РУВ:

- 1) организационно-экономические, связанные с необходимостью стандартизации и унификации технических решений;
- 2) методологические, обусловленные сложностью создания конфигураций для ПЛИС;
- 3) технические, связанные с особенностями вычислительного процесса.

**Стандартизация. Создание альянсов.** В работах [27 — 30] рассмотрена актуальность вопросов стандартизации и унификации для широкого



распространения вычислений на ПЛИС и сформулированы конкретные аспекты, подлежащие стандартизации. Унификация требуется не только для интерфейсов, протоколов обмена данными между РУВ и другими компонентами вычислительной системы, но и для программных компонентов на всех уровнях передачи информации от прикладной программы, выполняемой на ЦП, до вычислительной структуры, синтезированной в ПЛИС, и обратно.

Массовому распространению и, следовательно, удешевлению РУВ препятствует отсутствие международных нормативов и стандартов. Усложняющий фактор при этом — наличие большого числа конкурирующих фирм, производящих аппаратуру и создающих ПО для РУВ. Показательным в этом плане является длинный перечень высокоуровневых языков для разработки конфигураций, приведенный выше. Фактически каждая компания-разработчик инструментальных средств использует собственную модификацию языка программирования С, несовместимую с другими. Примечательно, что один из вариантов, а именно язык Standard System С, разработанный компанией Mentor Graphics, утвержден в качестве стандарта организацией IEEE. В данном случае стандартизация является не средством объединения интересов, а орудием конкурентной борьбы.

Очевидно, что решение таких проблем не под силу отдельной, даже очень влиятельной и состоятельной компании. Один из возможных путей их решения заключается в создании альянсов или консорциумов, объединяющих ряд организаций различного профиля, имеющих общие цели и одинаково заинтересованных в продвижении на рынок определенных изделий.

Примером такого сотрудничества является сообщество FHPCA (FPGA High Performance Computing Alliance), цель которого — разработка высокопроизводительных компьютерных решений на ПЛИС фирмы Xilinx «...для достижения беспрецедентного уровня быстродействия вычислений» [50]. Участники данного альянса: Xilinx — один из мировых лидеров по производству ПЛИС; Alpha Data и Nallatech — разработчики реконфигурируемых ускорителей; EPCC — Эдинбургский центр параллельных вычислений; ISLI — Институт системной интеграции (Шотландия); Scottish Enterprise — организация, спонсируемая ILSI; Algotronix — фирма по защите интеллектуальной собственности реконфигурируемых IP-ядер.

Альтернативным направлением движения в сторону стандартизации является создание некоммерческих объединений независимых специалистов. Пример такой организации — проект OpenRISC, который создан и поддерживается сообществом разработчиков свободно распространяемых IP-ядер OpenCores и является аппаратным аналогом проектов открытого

ПО [51]. Возможны также комбинированные решения. Например, фирма Convey Computer Corporation организовала некоммерческое сообщество свободных разработчиков, создающих вычислительные структуры для ее реконфигурируемых систем в различных прикладных областях [52, 53].

**Создание конфигураций. Стандарт OpenCL.** Проблема стандартизации — не единственное препятствие для широкого распространения РУВ. Одним из серьезных факторов является высокая трудоемкость процесса разработки загружаемых в ПЛИС конфигураций. В связи с высокой потребностью в разработках на базе ПЛИС в последнее время на решение данной проблемы направлены усилия многих исследователей и разработчиков. Как указано выше, кроме фирменных пакетов САПР для этого используются визуальные средства разработки и высокоуровневые языки программирования, модифицированные под проектирование аппаратуры, а также самоконфигурируемые системы.

Принципиальным ограничением возможностей инструментальных средств, использующих для синтеза аппаратных схем высокоуровневые языки, является необходимость автоматического распараллеливания программ. Это — одна из сложнейших научно-технических задач. Поэтому интерес представляет стандарт OpenCL (Open Computing Language — открытый язык вычислений), изначально создававшийся для нужд параллельного программирования. OpenCL является средством разработки параллельных кросс-платформенных программных приложений для широкого круга вычислительных систем [54—56].

Принципиальным является тот факт, что OpenCL, в отличие от инструментов на базе высокоуровневых языков, предоставляет программисту возможность самому определять и контролировать параллелизм. Таким образом, этот стандарт лучше согласуется с естественно-параллельной природой программируемой логики [57].

Об актуальности данного направления свидетельствует информационное сообщение [58] о том, что Ин-т системного программирования РАН, Ин-т прикладной математики им. М.В. Келдыша и компания «Т-Платформы» совместно со специалистами стран Евросоюза приступили к исследованиям в области суперкомпьютеров в рамках седьмой рамочной программы. Целью одного из трех направлений данного проекта является повышение продуктивности программирования систем на основе программируемых логических схем (FPGA). В сообщении отмечено, что широкое использование систем на базе FPGA-микросхем сдерживается сложностью их программирования, и важным условием их эффективного внедрения является использование открытых стандартов, например OpenCL.

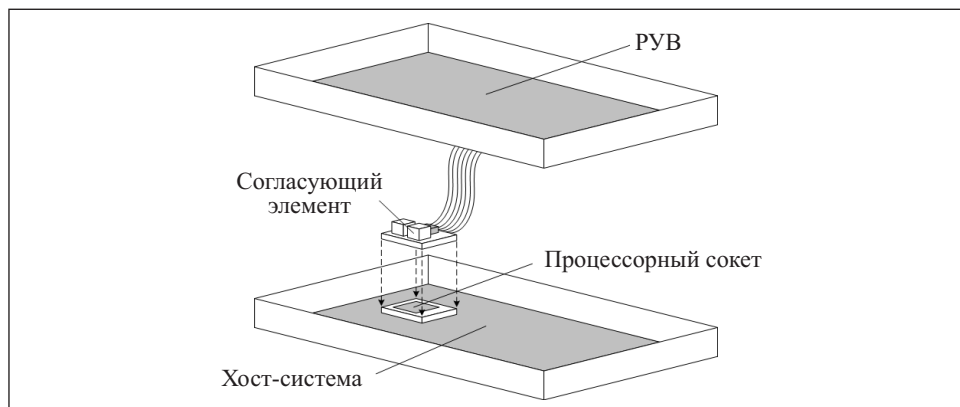


Рис. 5. Конструктивное решение РУВ

**Технические сложности.** Рассмотрим следующие две проблемы, ограничивающие дальнейшее повышение производительности РУВ:

1. Конструктивные ограничения на объем оборудования (микросхемы ПЛИС и ОЗУ), которое может быть размещено на присоединенном устройстве.

2. Увеличение объема локальной оперативной памяти без снижения эффективности ее функционирования.

При использовании наиболее быстрых интерфейсов, чувствительных к длине соединений, РУВ необходимо располагать внутри вычислительного модуля кластера либо системного блока ПЭВМ. Поэтому на размеры сопроцессора как электронного изделия накладываются ограничения, ужесточающиеся по мере ускорения обмена. Действительно, плата расширения, вставляемая в системную шину, может иметь большие габариты, чем модуль, устанавливаемый в слот памяти или процессорный сокет.

На протяжении всей истории развития вычислительной техники оперативная память была одним из самых проблемных компьютерных компонентов. Существенная зависимость цены от показателей емкости элементной базы ОЗУ вынуждает разработчиков задействовать механизмы кэширования, приводящие к сложной многоуровневой иерархии подсистемы памяти. В РУВ к сложностям кэширования добавляются конструктивные ограничения на объем, а также вопросы согласования по данным локального ОЗУ сопроцессора с основной памятью хост-системы. Однако наиболее существенной проблемой, по-видимому, является значительное усложнение труда программистов, вынужденных работать с локальной памятью сопроцессора. Рассмотрим, как решены перечисленные технические задачи в изделиях фирмы Convey Computer Corporation [53].

*Выбор конструктивного решения.* Противоречие между высокопроизводительным интерфейсом и малыми габаритами соответствующего разъема решено так. Реконфигурируемый вычислитель изготовлен в виде отдельного серверного модуля, установленного непосредственно над хост-системой (рис. 5) [59]. При этом в процессорный сокет хост-системы вставляется согласующий элемент скоростного канала передачи данных на вычислитель. Такое решение позволяет РУВ иметь габариты, соизмеримые с серверным блоком хост-системы и даже больше.

*Подсистема оперативной памяти.* При работе с присоединенными сопроцессорами пользователю приходится иметь дело как с общей памятью хост-системы (физически расположенной на материнской плате сервера или персонального компьютера), так и с локальной памятью сопроцессора. Пересылка данных из общего ОЗУ в локальное и обратно требует, с одной стороны, усилий программиста, с другой — является источником значительных временных задержек, способных свести на нет преимущества использования сопроцессора.

В решении, предложенном разработчиками фирмы Convey, вся физическая локальная память РУВ логически отображается в общую память, доступную программам, выполняемым на ЦП хост-системы, образуя с ней единое адресное пространство. Для обеспечения такой возможности между двумя видами памяти организован скоростной кэш-когерентный канал обмена данными. При этом использованы механизмы, специально заложенные для данных целей в межпроцессорную шину QPI. В результате ОЗУ хост-системы и локальная память сопроцессора для программиста представляется в виде единого виртуального запоминающего устройства, что существенно упрощает работу с РУВ.

## **Выводы**

Результаты проведенного анализа свидетельствуют о том, что РУВ позволяют в наибольшей степени разрешить противоречия между высокой производительностью, гибкостью и материальными затратами по сравнению с унифицированными вычислителями других классов. Поэтому в настоящее время они являются наиболее активно развивающимися изделиями вычислительной техники.

При исследовании развития вычислительных возможностей средств на основе ПЛИС выявлена тенденция к смещению фокуса внимания исследователей и разработчиков от автономных, полностью реконфигурируемых, суперкомпьютеров со своей операционной средой в сторону присоединенных вычислителей, способных эффективно использовать колоссаль-

ный объем программных наработок, технических решений, стандартов и протоколов, созданных для традиционной микропроцессорной вычислительной техники.

Давая оценку роли и месту класса РУВ в мире компьютерной техники, следует заметить, что данное направление не является конкурирующим не только по отношению к микропроцессорным системам, но и к другим видам УВ, в частности графическим процессорным устройствам широкого применения.

Можно предположить, что в дальнейшем каждый класс вычислительной техники займет свою нишу среди ресурсоемких вычислительных задач, способствуя взаимному развитию. Например, позитивными для развития РУВ в результате массового распространения технологии GPGPU стали такие явления, как решение некоторых проблем стандартизации и унификации, привлечение внимания пользователей кластерных систем к УВ, а также формирование новой категории программистов-прикладников, специализирующихся на использовании сопроцессоров.

A wide range of problems about a promising class of high performance digital devices — reconfigurable coprocessors based on FPGAs has been analyzed. A comparison with computation-intensive accelerators of another type was fulfilled. The obstacles that impede the propagation of reconfigurable coprocessors are investigated; the ways of their overcoming have been analyzed.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Hauck S.* The Roles of FPGAs in Reprogrammable Systems // Proc. of IEEE. — 1998. — Vol. **86**, № 4. — P. 615—639.
2. *Gokhale M.B., Graham P.S.* Reconfigurable Computing: Accelerating Computation with Field-Programmable Gate Arrays. — Springer Netherland. — 2005. — 238 p.
3. *Bovay J., Henderson B., Hsin-Ying Lin, Wadleigh K.* Accelerators for High Performance Computing Investigation // High Performance Computing Division Hewlett-Packard Company. [Электронный ресурс] Last Update: 01/24/2007 — Режим доступа: <http://www.hp.com/techservers/hpccn/hpccollaboration/ADCatalyst/downloads/accelerators.pdf>.
4. *Reconfigurable Computing: the Theory and Practice of FPGA-based Computation.* Edited by S. Hauck, A. DeHon. Amsterdam: Morgan Kaufmann/Elsevier, 2008. — 908 p.
5. *Максфилд К.* Проектирование на ПЛИС. Курс молодого бойца. — М. : Изд. дом «Додэка-XXI», 2007. — 408 с.
6. *Левин И.И.* Модульно-наращиваемая многопроцессорная вычислительная система со структурно-процедурной организацией вычислений на основе ПЛИС-технологии // Искусственный интеллект. — 2003. — № 4. — С. 446—453.
7. *Каляев И.А., Левин И.И., Семерников Е.А.* Архитектура семейства реконфигурируемых вычислительных систем на основе ПЛИС // Искусственный интеллект. — 2008. — № 3. — С. 663—673.
8. *Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И.* Реконфигурируемые мультимедийные вычислительные структуры. Под общ. ред. И.А. Каляева. 2-е изд., перераб. и доп. — Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. — 344 с.

9. *Каляев И.А., Левин И.И., Семерников Е.А., Дордопуло А.И.* Реконфигурируемые вычислительные системы на основе ПЛИС семейства VIRTEX-6 // Вест. Уфимского гос. авиационного технического ун-та (УГАТУ). — 2011. — 15, № 5. — С. 148—154.
10. *Мельник А.О., Мохаммад Аль Хабабсах* Програмні спеціалізовані процесори для реконфігурованих прискорювачів універсальних комп'ютерів // Наук. вісник Чернівецького ун-ту. — 2008. — Вип. 426. Фізика. Електроніка. — С. 20—25.
11. *Мельник А.О., Мельник В.А., Сарайрех З.Т.* Використання реконфігурованих прискорювачів для підвищення продуктивності персональних комп'ютерів // Наук. вісн. Чернівецького ун-ту. Комп'ютерні системи та компоненти. — 2010. — Т. 1, вип. 1. — С. 20—25.
12. *Мельник В.А., Сарайрех З.* Самоконфігуровні апаратні прискорювачі обчислень в комп'ютерах // Вісн. Нац. ун-ту «Львівська політехніка». «Комп'ютерні системи та мережі». — 2010. — № 688. — С. 163—171.
13. *Мельник А.О., Мельник В.А.* Персональні суперкомп'ютери: архітектура, проектування, застосування. — Львів: Вид-во «Львівської політехніки», 2013. — 516 с.
14. *Палагин А.В., Опанасенко В.Н.* Реконфигурируемые вычислительные системы: Основы и приложения. — К. : «Просвіта», 2006. — 280 с.
15. *Опанасенко В.Н.* Высокопроизводительные реконфигурируемые компьютеры на базе FPGA // Проблемы информатизации та управління. Зб. наук. праць НАУ. — 2009. — Вип. 3 (27). — С. 114—118.
16. *Соловьев В.В.* Проектирование цифровых систем на основе программируемых логических интегральных схем. — М. : Горячая линия — Телеком, 2001. — 636 с.
17. *Библио П.Н.* Синтез логических схем с использованием языка VHDL. — М. : СОЛОН - Р, 2002. — 384 с.
18. *Грушвицкий Р.И., Мурсаев А.Х., Узрюмов Е.П.* Проектирование систем на микросхемах программируемой логики. — СПб. : БХВ-Петербург, 2002. — 608 с.
19. *Сергиенко А.М.* VHDL для проектирования вычислительных устройств. — Киев: ЧП «Корнейчук», ООО «ТИД «ДС», 2003. — 208 с.
20. *Стешенко В.Б.* ПЛИС фирмы Altera: элементная база, система проектирования и языки описания аппаратуры. — М. : Изд. дом «Додэка-XXI», 2002. — 576 с.
21. *Зотов В.Ю.* Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx в САПР WebPACK ISE. — М. : Горячая линия-Телеком, 2006. — 520 с.
22. *Гильгурт С.Я.* Применение типовых устройств на базе программируемой логики для решения вычислительных задач // Тез. докл. II междунар. конф. «Параллельные вычисления и задачи управления». 4—6 окт. 2004 г. — М. : Ин-т проблем управления им. В.А. Трапезникова РАН, 2004. — С. 514—530.
23. *Гильгурт С.Я., Гиранова А.К.* Программное обеспечение для тестирования аппаратных средств реконфигурируемых вычислителей // Моделювання та інформаційні технології. Зб. наук. пр. ІПМЕ НАН України. — 2006. — Вип. 38. — С. 8—14.
24. *Гильгурт С.Я., Гиранова А.К.* Некоторые вопросы обмена данными между персональным компьютером и реконфигурируемым устройством // Там же. — 2007. — Вип. 43. — С. 86—94.
25. *Гиранова А.К.* Анализ программного обеспечения реконфигурируемых вычислителей // Там же. — 2007. — Вип. 41. — С. 43—48.
26. *Гильгурт С.Я.* Анализ существующих унифицированных вычислителей для выполнения ресурсоемких расчетов // Там же. — 2008. — Вип. 48. — С. 115—120.
27. *Гильгурт С.Я.* О применении реконфигурируемых унифицированных вычислителей для решения научно-технических задач / Параллельные вычислительные технологии // Тр. междунар. науч. конф. Санкт-Петербург, 28 января — 1 февраля 2008 г. — Челябинск: Изд. ЮУрГУ, 2008. — С. 358—363.

28. *Гильгурт С.Я.* Некоторые вопросы применения реконфигурируемых вычислителей для решения задач компьютерного моделирования // Тез. доп. XXVI наук.-техн. конф. «Моделирования» — Київ: Ін-т проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України, 2008. — С. 411—415.
29. *Гильгурт С.Я.* Обзор современных реконфигурируемых унифицированных вычислителей // Моделивання та інформаційні технології. Зб. наук. пр. ІПМЕ НАН України. — 2008. — Вип. 49. — С. 17—24.
30. *Гильгурт С.Я.* Анализ применения унифицированных вычислителей в интеллектуальных системах // Искусственный интеллект. — 2009. — № 1. — С. 144—148.
31. *Гильгурт С.Я.* Анализ типовых режимов обмена данными с реконфигурируемыми вычислителями // Зб. наук. пр. ІПМЕ НАН України. — 2011. — Вип. 59. — С. 113—121.
32. *Гиранова А.К.* Разработка пакета программ для проведения экспериментов с реконфигурируемыми вычислителями // Там же. — 2011. — Вип. 59. — С. 124—129.
33. *Smart Connected Devices in Emerging Markets to Surpass 1 Billion Unit Shipments by 2014 with More Than 60% Going to BRIC Countries, According to IDC* — Press Release [Электронный ресурс] Last Update: 06/10/2013. — Режим доступа: — <http://www.idc.com/getdoc.jsp?containerId=prUS24154913>.
34. *Starbridge: The Hypercomputing Company.* [Электронный ресурс] — Режим доступа: — <http://www.starbridgesystems.com>
35. *Ройзензон Г.В.* Выбор вычислительных кластеров на основе анализа количественной и качественной информации // Искусственный интеллект. — 2004. — № 2. — С. 73—79.
36. *Якуба А.А., Комухаев Э.И., Рябчун С.Г.* Развитие ускорителей специализированных вычислений // Математичні машини і системи — 2010. — № 2. — С. 10—20.
37. *Корнеев В.В., Киселев А.В.* Современные микропроцессоры. — Изд. 3-е перераб. и доп. — СПб. : БХВ-Петербург, 2003. — 448 с.
38. *AMD о перспективах многоядерности.* [Электронный ресурс] — <http://www.overclockers.ru/hardnews/24205/AMD—o-perspektivah-mnogoyadernosti.html> - 15.12.2006 05:47. — Режим доступа: — <http://www.overclockers.ru/hardnews/24205/AMD—o-perspektivah-mnogoyadernosti.html>.
39. *AMD Accelerated Processing Units/AMD.* [Электронный ресурс].— Режим доступа: — <http://www.amd.com/us/products/technologies/apu/Pages/apu.aspx>
40. *Боресков А.В., Харламов А.А.* Основы работы с технологией CUDA. — М. : ДМК Пресс, 2010. — 232 с.
41. *The Intel Xeon Phi Coprocessor: Parallel Processing, Unparalleled Discovery / Intel.* [Электронный ресурс]. — Режим доступа: — <http://www.intel.com/content/www/us/en/high-performance-computing/high-performance-xeon-phi-coprocessor-brief.html>.
42. *TOP500 Supercomputer sites.* [Электронный ресурс].— Режим доступа: — <http://www.top500.org>
43. *FPGA Boards and Systems.* [Электронный ресурс]. — Last Update: 07/09/2013 — Режим доступа: — [http://www.fpga-faq.com/FPGA\\_Boards.shtml](http://www.fpga-faq.com/FPGA_Boards.shtml).
44. *Боресков А.В. и др.* Параллельные вычисления на GPU. Архитектура и программная модель CUDA. — М. : Изд-во Московского университета, 2012. — 336 с.
45. *Xilinx.* [Электронный ресурс].— Режим доступа:— <http://www.xilinx.com>.
46. *Altera.* [Электронный ресурс].— Режим доступа:— <http://www.altera.com>.
47. *Intel Quickpath Interconnect Maximizes Multi-Core Performance / Intel.* [Электронный ресурс]. — Режим доступа: — <http://www.intel.com/content/www/us/en/io/quickpath-technology/quickpath-technology-general.html>

48. *AMD HyperTransport Technology / AMD*. [Электронный ресурс]. — Режим доступа: — <http://www.amd.com/us/products/technologies/hypertransport-technology/Pages/hypertransport-technology.aspx>.
49. *Akella S., Wake H.E., Davis J. P., Buell D.A. Porting EDIF Netlists to the Viva Environment for Integrated Custom Computing Applications / MAPLD-2003: Military Applications of Programmable Logic Devices* [Электронный ресурс]. — Режим доступа: — <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.1956&rep=rep1&type=pdf>.
50. *FHPCA*. [Электронный ресурс].— Режим доступа:— <http://www.fhpcsa.org>.
51. *OpenCores*. [Электронный ресурс].— Режим доступа: — <http://www.opencores.org>.
52. *Convey Computer*. [Электронный ресурс]. — Режим доступа: — <http://www.convey-computer.com>.
53. *Bakos J.D. High-Performance Heterogeneous Computing with the Convey HC-1 // Computing in Science and Engineering*. — 2010. — Vol. 12, №. 6. — P. 80—87.
54. *The open standard for parallel programming of heterogeneous systems / Khronos group*. [Электронный ресурс].— Режим доступа: — <http://www.khronos.org/ocl/>.
55. *Munshi A., Gaster B., Mattson T., Fung J. OpenCL Programming Guide*. — Boston: Addison Wesley Professional, 2011. — 603 p.
56. *Implementing FPGA Design with the OpenCL Standard / Altera Corporation*. [Электронный ресурс]. — WP-01173-1.0, November 2011. — Режим доступа: — <http://www.altera.com/literature/wp/wp-01173-ocl.pdf>.
57. *Гильгурт С. Я. О применении стандарта OpenCL для создания реконфигурируемых устройств на базе ПЛИС // Зб. наук. пр. ІПМЕ НАН України*. — 2012. — Вип. 62. — С. 3—11.
58. *Лаврентьева Н. Россия и Евросоюз сделают суперкомпьютеры быстрее и эффективнее* [Электронный ресурс].— Режим доступа: — <http://www.cnews.ru/news/top/index.shtml?2011/03/18/432521/>.
59. *Mutke E.M. Putting Personality into High Performance Computing // Frühjahrstreffen des ZKI-Arbeitskreises*. — «Supercomputing». 19—20 Mai 2011. — DESY, Zeuthen.

Поступила 03.07.13

*ГИЛЬГУРТ Сергей Яковлевич, канд. техн. наук, ст. науч. сотр. Ин-та проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины. В 1986 г. окончил Киевский ин-т инженеров гражданской авиации. Область научных исследований — реконфигурируемые вычисления и процессоры с гибкой архитектурой.*