

УДК 004.27; 004.315: 004.25; 004.382.2

А.В. ПАЛАГИН\*, Ю.С. ЯКОВЛЕВ\*

## ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ НА КРИСТАЛЛЕ ПЛИС

\*Институт кибернетики имени В.М. Глушкова НАН Украины, Киев, Украина

**Анотація.** На підставі аналізу вітчизняних і закордонних публікацій визначені особливості проектування «системи на кристалі» ПЛИС (System on Chip-SoC), що включає складні функціональні блоки (СФ-блоки), а також вбудовані процесорні ядра. При цьому представлені можливі типи СФ-блоків, а також програмного забезпечення, та визначені проблеми, що виникають при проектуванні. Наведено параметри деяких синтезованих СФ-блоків, а також інформацію про маршрут проектування проекту на кристалі.

**Ключові слова:** система на кристалі, складні функціональні блоки (СФ), програмовані інтегральні схеми (ПЛИС), маршрут проектування.

**Аннотация.** На основании анализа отечественных и зарубежных публикаций определены особенности проектирования «системы на кристалле» ПЛИС (System on Chip-SoC), включающей сложные функциональные блоки (СФ-блоки), а также встроенные процессорные ядра. При этом представлены возможные типы СФ-блоков, а также программного обеспечения, и определены проблемы, возникающие при проектировании. Приведены параметры некоторых синтезируемых СФ-блоков, а также информация о маршруте проектирования проекта на кристалле.

**Ключевые слова:** система на кристалле, сложные функциональные блоки (СФ), программируемые интегральные схемы (ПЛИС), маршрут проектирования.

**Abstract.** On the bases of the analysis of domestic and foreign publications there were defined the features of designing “systems on a crystal” FPGA (System on Chip-SoC), including difficult functional blocks (DFblocks) as well as the built in processor cores. Moreover there were presented the possible types of DFblocks and software arising at designing. The parametres of some synthesized processor DF-blocks as well as brief information about results of two developed projects on a crystal are resulted.

**Keywords:** system on a crystal, the difficult functional blocks (DF), programmed integrated schemes (FPGA), a designing route.

### 1. Введение

Для программируемых логических интегральных схем (ПЛИС) в связи с достигнутым уровнем интегральной технологии (не менее 10 млн вентиляей на кристалле) и свойствами адаптации ПЛИС под реализуемые алгоритмы методы проектирования с использованием микросхем на плате оказываются неприемлемыми. Поэтому существует необходимость в создании новой методологии проектирования системы на кристалле (в иностранной литературе – System-on-Chip, SoC) и определения её особенностей по сравнению с традиционными методами.

Система на кристалле – это сверхбольшая интегральная схема (СБИС), содержащая на кристалле различные сложные функциональные блоки (СФ-блоки), которые образуют законченное изделие для автономного применения в электронной аппаратуре [1–3]. СФ-блоки, предназначенные для использования в разнообразных проектах, часто называют IP (Intellectual Property)-модулями. Эти модули должны быть повторяемыми и настраиваемыми под решаемые задачи в ряде проектов SoC. Повторное применение таких модулей (IP Core reuse) можно назвать вычислительными заготовками за их функциональную и технологическую адаптируемость, что позволяет уменьшить трудозатраты и сроки проек-

тирования архитектуры SoC несмотря на то, что недостающие функциональные блоки приходится проектировать самостоятельно. При этом в состав СФ-блоков, как правило, входит микропроцессорное ядро с периферийными устройствами в различном сочетании. СФ-блоки, используемые при проектировании SoC, имеют две основные формы представления:

- в виде топологических фрагментов, которые могут быть непосредственно реализованы в физической структуре кристалла – аппаратно реализованные (hard) СФ-блоки;
- в виде моделей на языке описания аппаратуры (Verilog, VHDL), которые средствами САПР могут быть преобразованы в топологические фрагменты для реализации на кристалле – синтезируемые (soft) СФ-блоки.

В крупных фирмах, которые долгие годы занимаются разработкой СБИС, а теперь и SoC, наработаны большие библиотеки стандартных модулей: ОЗУ, АЛУ, периферийных устройств, а также предложены этапы проектирования и моделирования работы SoC и ее верификация в комплексе. Необходимую заготовку можно поискать, например, в банке бесплатных IP cores, которые находятся на сайте [www.opencores.org](http://www.opencores.org), а также на сайтах, где указаны компании-производители структурных ASIC и средств САПР для них [4–10]:

- AccelChip, Inc. (<http://www.accelchip.com>);
- Aldec, Inc. (<http://www.aldec.com>);
- AMI Semiconductor (<http://www.amis.com>);
- Celoxica, Ltd. (<http://www.celoxica.com>);
- ChipX, Corp. (ранее Chip Express, Corp.) (<http://www.chipexpress.com>);
- eASIC, Corp. (<http://www.easic.com>);
- Fujitsu, Ltd. (<http://www.fujitsu.com>);
- Leopard Logic, Inc. (<http://www.leopardlogic.com>);
- Lightspeed Semiconductor, Corp. (<http://www.lightspeed.com>);
- LSI Logic, Corp. (<http://www.lsillogic.com>);
- NEC Electronics, Corp. (<http://www.necel.com>);
- Synplicity, Inc. (<http://www.synplicity.com>);
- Tera Systems, Inc. (<http://www.terasystems.com>);
- Triscend, Corp. (<http://www.triscend.com>).

В результате существует большой выбор библиотек специализированных СФ-блоков для различных прикладных областей и технологий изготовления микросхем, в частности, библиотек СФ-блоков для ПЛИС, представленных в виде синтезируемых блоков на языках высокого уровня, списков цепей в элементном базисе производителей ПЛИС и готовых макросов с топологической реализацией.

## **2. Актуальность**

Успехи интегральной технологии создания сверхбольших интегральных схем (СБИС), способность адаптации ПЛИС под класс решаемых пользователем задач, а также непригодность для «систем на кристалле» существующих методов проектирования компьютерной системы на плате с микросхемами, определили объективную необходимость в создании новой методологии проектирования SoC и определения её особенностей по сравнению с существующими традиционными методами.

## **3. Цель работы**

На основе анализа существующих методов проектирования «систем на кристалле» определить особенности проектирования SoC, а также основные проблемы, возникающие при проектировании, в частности: проблема оптимального выбора СФ-блоков и проектирования недостающих, проблема аппаратно-программной верификации фактически на каждом

этапе проектирования, а также проблема одновременной разработки как аппаратной, так и программной части, и др., на решение которых разработчик должен обратить особое внимание.

#### 4. Структура, основные этапы и задачи, решаемые при проектировании SoC

SoC – это возможность использования широкой номенклатуры СФ-блоков, имеющих на рынке, которые могут быть реализованы на базе различных функциональных библиотек и технологий и интегрированы в кристалл средствами современных САПР [1, 10, 11].

Фактически весь процесс разработки SoC делится на четыре основных этапа:

1. Разработка архитектуры SoC на системном уровне.
2. Выбор имеющихся СФ-блоков из базы данных (внутри фирмы, других фирм или поставщиков СФ-блоков).
3. Проектирование недостающих блоков.
4. Интеграция всех блоков на кристалле.

В процессе проектирования SoC разработчик имеет возможность выбора следующих решений:

- самостоятельная разработка необходимых СФ-блоков;
- покупка СФ-блоков у ведущих разработчиков и производителей микросхем;
- поиск и применение СФ-блоков, предоставляемых в открытом доступе ([www.opensource](http://www.opensource) и др. источники [2]).

СФ-блоки должны включать элементы инфраструктуры системы (интерфейсы, системы питания и синхронизации, встроенные средства контроля). Эти блоки не должны занимать большую площадь кристалла и использовать много внешних компонентов. В табл. 1 приведены параметры небольшого набора синтезируемых процессорных СФ-блоков [4, 5], а на рис. 1 – один из возможных вариантов состава структуры SoC, представленного для седьмого семейства микросхем типа Zynq-7000 [7, 8].

Таблица 1. Параметры синтезируемых процессорных СФ-блоков

| Процессорные СФ-блоки     | Разрядность | Тактовая частота, МГц | Производительность, DMIPS | Число LUT |
|---------------------------|-------------|-----------------------|---------------------------|-----------|
| PicoBlaze (Xilinx)        | 8           | 250                   | 125                       | 110       |
| MicroBlaze (Xilinx)       | 32          | 200                   | 166                       | 1250      |
| Nios II Economy (Altera)  | 8           | 200                   | 31                        | 600       |
| Nios II Standard (Altera) | 16          | 165                   | 127                       | 1300      |
| Nios II Fast (Altera)     | 32          | 185                   | 218                       | 1800      |
| LEON3 (Gaisler)           | 32          | 150                   | 150                       | 3500      |

В данной таблице приведены характеристики некоторых СФ-блоков, предлагаемых компаниями Xilinx, Altera, Gaisler Research. Это процессорные ядра с RISC-архитектурой, для реализации которых требуется всего несколько сотен или тысяч логических блоков (LUT – Look-Up-Table). Альтернативой применению аппаратных блоков является использование синтезируемых процессорных ядер (Soft-Processor Cores), которые представляют собой СФ-модули, разработанные и оптимизированные для SoC, реализуемых на основе ПЛИС. Связь процессорной системы с программируемой логикой может изменяться в достаточно широких пределах, в зависимости от требований конкретных приложений. При

этом общее число сигнальных линий может превышать 3000 [7, 8]. Это позволяет разработчику эффективно использовать ресурсы программируемой логики для создания аппаратных ускорителей и блоков цифровой обработки сигналов (ЦОС).

Процессорная система и программируемая логика имеют независимые цепи питания, что позволяет отключать блок логики для снижения энергопотребления. В дополнение к этому возможно динамическое управление тактовой частотой процессоров и отключение неиспользуемых периферийных устройств. Система управления тактовыми сигналами включает высокоскоростные буферы и формирователи для распределения тактовых сигналов на синтезаторы частоты и схемы фазового сдвига.

Многофункциональные ОС, такие как Linux или Windows, обеспечивают эффективное использование сетевых возможностей и развитого пользовательского интерфейса. Специализированные ОС позволяют экономить аппаратные ресурсы, управлять заданиями или процессами, в том числе в режиме реального времени.

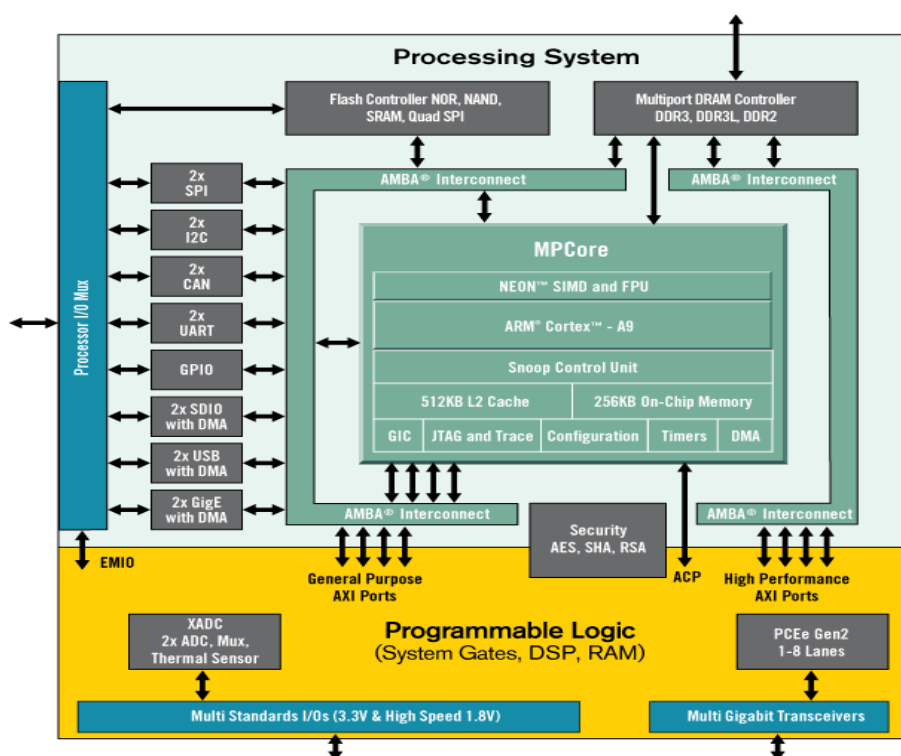


Рис. 1. Программируемый SoC Zynq-7000, содержащий два ядра семейства процессоров ARM Cortex-A9

Zynq-7000 компании Xilinx представляет интерес как для разработчиков программного обеспечения, так и для специалистов по применению FPGA. Основные параметры микросхем семейства Zynq-7000 приведены в табл. 2 [9].

Таблица 2. Состав и основные параметры микросхем семейства Zynq-7000

|                           | Z-7010 | Z-7015 | Z-7020 | Z-7030 | Z-7035 | Z-7045 | Z-7100 |
|---------------------------|--------|--------|--------|--------|--------|--------|--------|
| Logic Cells (K)           | 28     | 74     | 85     | 125    | 275    | 350    | 444    |
| Block RAM (Mb)            | 2,1    | 3,3    | 4,9    | 9,3    | 17,6   | 19,1   | 26,5   |
| DSP Slices                | 80     | 160    | 220    | 400    | 900    | 900    | 2,020  |
| Maximum I/O Pins          | 100    | 150    | 200    | 250    | 362    | 362    | 400    |
| Maximum Transceiver Count | -      | 4      | -      | 4      | 16     | 16     | 16     |

Однокристальное решение, в совокупности с 28 нм технологическим процессом, обеспечивает низкое энергопотребление при высокой производительности процессорной системы, использующей два процессора ARM Cortex A9 MPCore и программируемую логику. При этом размещение процессоров и логических схем в пределах одной коммутирующей матрицы существенно ускоряет и упрощает обмен данными между ними. Широкий спектр приборов семейства Zynq-7000 (табл. 2) образует платформу Zynq-7000 EPP, которая позволяет разработчикам ориентироваться как на рынок малобюджетных устройств, так и на высокопроизводительные приложения, используя стандартный набор инструментов разработки: телевизионные камеры, системы помощи водителю, управление электроприводами, промышленные сети, системы машинного зрения, интеллектуальные и IP-камеры, приёмопередатчики и модемы LTE, устройства медицинской диагностики и обработки изображений, многофункциональные офисные приборы, системы обработки видеосигналов и др. Общий вид такой платформы приведен на рис. 2 [12]. Для построения SoC различного назначения платформа Zynq-7000 EPP содержит следующие функциональные блоки: процессорную подсистему, включающую процессорный модуль, интерфейсы памяти, периферийные интерфейсы, межблочные интерфейсы и интерфейсы к программируемой логике, а также программируемую логику. При этом процессор ARM Cortex A9 MPCore имеет встроенную память, богатый набор периферийных устройств, интерфейсы к внешней памяти. Взаимодействие между двумя процессорами может осуществляться посредством межпроцессорных прерываний, через области общей памяти путём передачи сообщений.

Программируемая логика содержит конфигурируемые логические блоки (CLB), конфигурируемые двухпортовые блоки памяти (BRAM), ячейки ЦОС с 25×18-битным множителем, 48-битным аккумулятором и сумматором (DSP48E1), АЦП (XADC), управляемые блоки формирования тактовых сигналов (CMT), конфигурируемый блок шифрования (AES256) и аутентификации (SHA), конфигурируемый блок ввода-вывода (SelectIO).

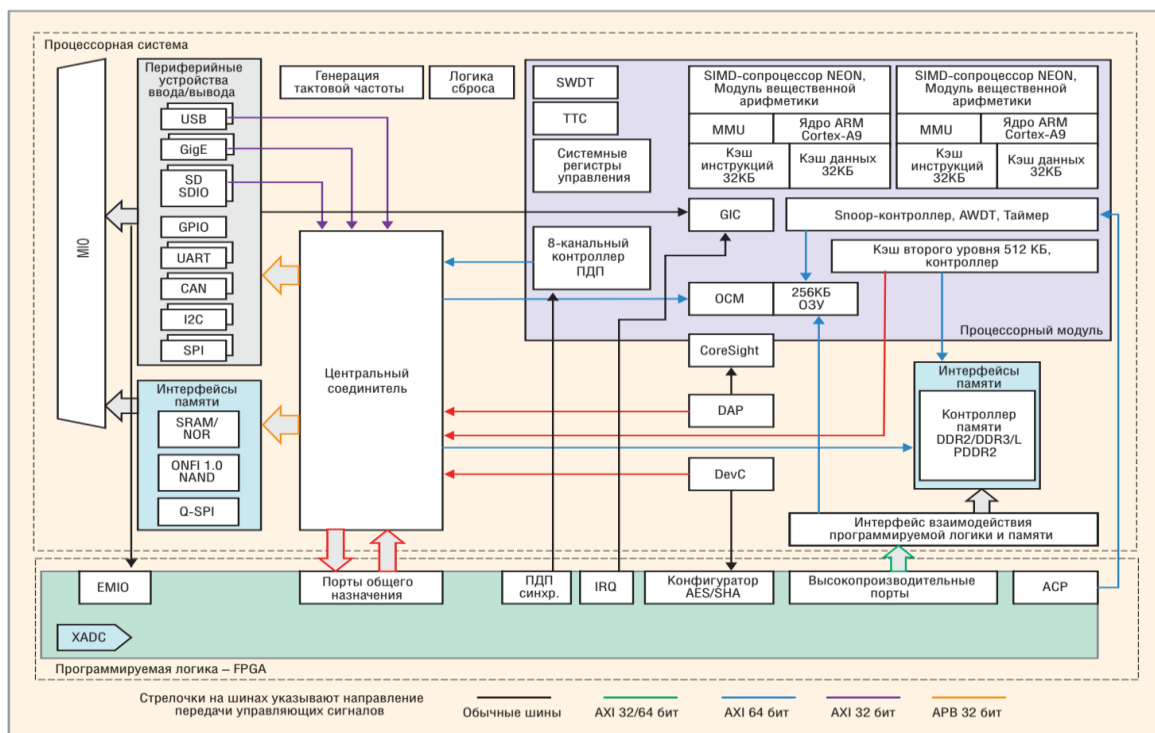


Рис. 2. Структурная схема расширяемой процессорной платформы Zynq-7000 EPP

В старших сериях семейства присутствуют высокоскоростные последовательные приёмопередатчики (GTX) и блоки PCI Express (PCIe).

Технология Advanced Extensible Interface (AXI) позволяет достичь высоких скоростей передачи данных между двухъядерной вычислительной подсистемой на базе

ARM Cortex A9 MPCore и программируемой логикой при низкой потребляемой мощности.

Периферийные устройства процессорной системы образуют до 54-х внешних мультиплексируемых линий ввода-вывода (MIO).

Блочная оперативная память в седьмой серии FPGA Xilinx может хранить до 36 Кбит данных и может быть сконфигурирована как два независимых блока по 18 Кбит или как один блок в 36 Кбит.

Специализированные ячейки цифровой обработки сигналов DSP48E1 позволяют существенно ускорить многие приложения. Основные характеристики ячейки ЦОС: аппаратный умножитель (25×18 бит) с 48-битным аккумулятором, 25-битный предварительный сумматор, конвейер, многофункциональное АЛУ, выделенные шины для каскадирования.

Встроенные АЦП поддерживают несколько режимов работы: запуск по внешнему сигналу; режим непрерывного преобразования; поддержку различных типов аналоговых сигналов (униполярных, дифференциальных); 17 внешних мультиплексируемых каналов; частоту выборки до 1 МГц; выбор встроенного или внешнего источника опорного напряжения; встроенные датчики температуры (погрешность ±4°C) и напряжения питания (погрешность ±1%); возможность доступа к АЦП через интерфейс JTAG.

Система управления тактовыми сигналами включает высокоскоростные буферы и формирователи для распределения тактовых сигналов на синтезаторы частоты и схемы фазового сдвига.

Компания Xilinx предоставляет средства разработки и отладки программного обеспечения для устройств Zynq-7000 EPP, которые содержат интегрированную среду разработки приложений C/C++, GNU-компилятор, JTAG-отладчик, набор вспомогательных программ. При этом спецификации должны разрабатываться как на аппаратную, так и на программную часть проекта, которые должны включать в себя следующую информацию.

Для аппаратной части [1]:

- выполняемые функции;
- внешний интерфейс к другим блокам (контакты, шины, протоколы);
- интерфейс с ПО (регистры);
- временные параметры;
- быстродействие;
- особенности физического уровня (площадь кристалла, потребляемая мощность).

Для программной части:

- выполняемые функции;
- временные параметры;
- быстродействие;
- интерфейс к аппаратной части;
- структура, ядро;
- инструментальные средства разработки сторонних производителей, драйверы для работы с устройствами в ПЛИС. При этом основной состав программных средств должен содержать структуру программного обеспечения в виде [6] коммуникационной библиотеки, программ и утилитов конфигурирования, управления и диагностики ПЛИС, примеров и тестов.

В состав ПО также входят утилиты загрузки. При этом используются следующие инструментальные средства [6]: средства разработки ПО для ПЛИС (синтез схем), а также средства для САПР и моделирования.

В маршрут проектирования на разных уровнях вводится операция программно-аппаратной верификации. Наиболее ответственными в этом смысле являются этапы функционального и логического проектирования. Относительно короткий цикл и низкая стоимость средств проектирования, возможность устранения ошибок путем перепрограммирования, возможность реконфигурации на класс решаемых задач делают реализацию проектов на основе ПЛИС весьма привлекательной. Однако потребляемая мощность, низкая производительность и высокая стоимость кристаллов по сравнению с ASIC существенно ограничивают область применения ПЛИС.

## 5. Основные этапы (уровни) проектирования SoC

В настоящее время не известно какой-либо интегрированной системы САПР, способной эффективно решать задачи проектирования систем на кристалле во всем диапазоне возможных способов реализации. В этом плане в общем маршруте проектирования, как правило, выделяют три наиболее важных уровня проектирования, а именно: системный, функциональный и уровень проектирования топологии, где выполняется реализация проекта на физическом уровне с детальным размещением, трассировкой, генерацией тестовых структур и верификацией топологии.

На других уровнях, например, на уровне регистровых передач (register transfer level), проект представляется совокупностью арифметических и логических узлов, элементов памяти и т.д. Вентильный или логический уровень (logic level) описывает проект на уровне логических вентилях (logic gates) и триггеров (flip-flops). В этом случае поведение схемы можно описать системой логических уравнений. Эти логические элементы представляются на кремниевом (топологическом) уровне (geometric level) в виде топологических элементов и межсоединений [2–5].

Тем не менее, основной этап проектирования – это системный (system-level description), который состоит из поведенческого описания в терминах функций, выражений, алгоритмов. Системный уровень – это уровень выбора общей концепции построения системы, включающий моделирование операционной среды, в которой будет работать проектируемая система, определение статических и динамических сценариев, планирование целевых задач. Он включает архитектурный уровень с моделированием и анализом производительности систем, сетевых архитектур и протоколов, пропускной способности каналов, а также уровень микроархитектуры, то есть моделирование и анализ алгоритмов, протоколов, схем разрешения конфликтов на шинах, методов управления памятью, программно-аппаратное разделение и разработка программного обеспечения (драйверы и др.). Именно на этом этапе определяются все основные характеристики разрабатываемого микросистемного устройства. Возможно и исключение этапа макетирования SoC, если все используемые СФ-блоки аттестованы и адекватно описаны на языках высокого уровня (VHDL, VHDL-AMS и др.).

Функциональный уровень [2–5, 10–12]. Основные цели этапа – это создание исполняемой функциональной модели на языке описания аппаратуры (VHDL, Verilog), а также подготовка детальной спецификации всех блоков и системы в целом.

Первая задача этапа функционального уровня – разработка всех недостающих СФ-блоков. Для аналоговых блоков, проектируемых на транзисторном уровне, требуется еще и создание поведенческой модели на языке описания аппаратуры. Вторая задача этого уровня – согласование интерфейсов СФ-блоков и построение шинной архитектуры, а также прогноз параметров линий связи. На этом этапе функционального проектирования детализируется временная диаграмма работы БИС и рассчитывается баланс задержек между блоками. Третья задача – разработка и согласование внешних интерфейсов. Моделирование внешних интерфейсов должно проводиться с учетом реактивных параметров корпуса и



внешних линий связи. Четвертая задача – это расчет потребляемой мощности и разработка мер по ее экономии.

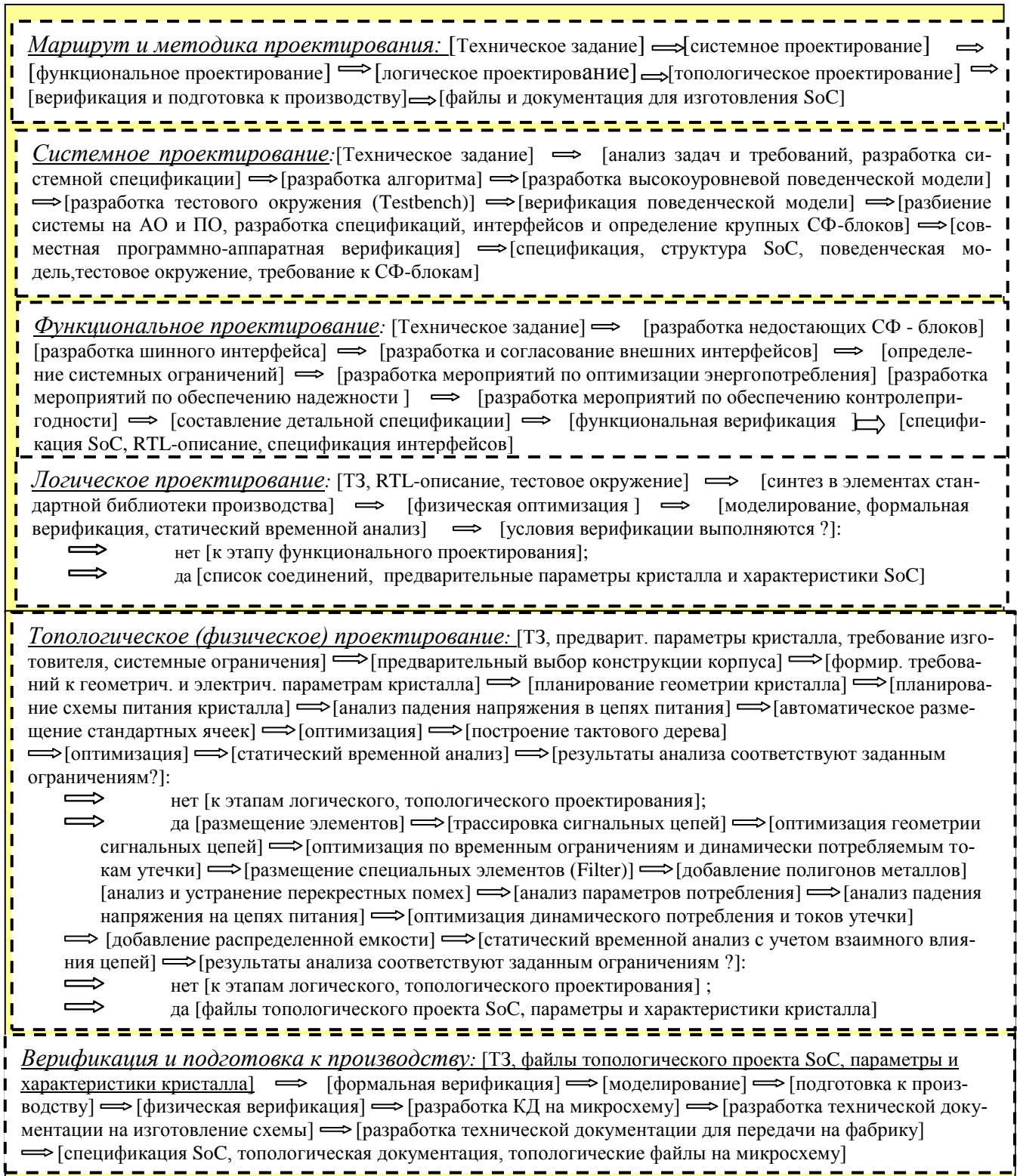


Рис. 3. Маршрут и методика проектирования микросхемы контроллера класса «система на кристалле»

Пятая задача – прогноз и повышение надежности, а также процента выхода годных путем введения избыточности и резервирования в схему. Шестая задача – обеспечение



контролепригодности и разработка встроенных средств контроля. Седьмая задача – составление окончательной детальной спецификации, которая будет использоваться и при физическом проектировании микросхемы, и при разработке программы.

Эффективный контроль и прогнозирование на функциональном уровне существенно сокращают число итераций и время проектирования.

Завершающим этапом является аттестация проекта. Цель аттестации – это принятие решения о готовности проекта к началу освоения производства. Для этого требуется соответствие опытных образцов СБИС требованиям нормативной технической документации (техническим условиям, техническим заданиям, справочному листу), а также наличие самой нормативной документации.

К очевидным достоинствам «систем на кристалле» можно отнести максимальную эффективность в решении прикладной задачи, обусловленную оптимизацией внутренней структуры. Кроме того, отсутствие в оптимизированной «системе на кристалле» избыточности, присущей системам, построенным из универсальных компонентов, позволяет комплексно минимизировать затраты энергии, повысить надежность устройства за счет замены потенциально ненадежных связей между его узлами (печатные проводники, разъемы, паяные соединения) на внутрикристалльные и в конечном итоге значительно уменьшить габариты изделия в целом. На рис. 3 (в качестве примера) приведены маршрут и методика проектирования микросхемы контроллера класса «система на кристалле» [11].

## 7. Особенности проектирования SoC

Таким образом, на основе анализа приведенной выше информации можно сделать вывод об особенностях проектирования SoC, которые определяются уровнем технологии, допускающим количество элементов на кристалле более 10 млн транзисторов, возможностью размещения на кристалле самостоятельно завершенных сложных функциональных блоков (СФ) и создания программируемых блоков (модулей), областью применения и, наконец, задачами, на решение которых ориентирована SoC. Например, уровень технологии может обеспечить следующие качественные и количественные признаки, определяющие особенности проектирования:

- высокую системную интеграцию: программируемая платформа – центральный процессор, конфигурируемая логика, ячейки ЦОС, ввод-вывод, АЦП и т.д.;
- увеличенную системную производительность: высокопроизводительный  $n$  – ядерный процессор (тактовые частоты  $\geq 1$  ГГц), более чем 10-кратное ускорение некоторых операций;
- снижение затрат на комплектующие: интегрированная платформа позволяет сократить затраты до 40% за счёт сокращения числа необходимых компонентов;
- снижение энергопотребления: реализация системы на одном кристалле в сумме позволяет сократить общее потребление на 50% по сравнению с отдельными решениями;
- ускоренный вывод продукта на рынок: гибкая и масштабируемая платформа, широкий выбор средств разработки, отладки, операционных систем и IP-ядер.

Что касается рекомендуемых задач, которые подлежат решению с помощью SoC, можно отметить широкий диапазон рекомендуемых применений (например, используя структурную схему расширяемой процессорной платформы Zynq-7000 EPP, рис.2), начиная от малобюджетных тем до задач, требующих высокоскоростных вычислений. Как было указано выше, к таким задачам можно отнести разработку на SoC средств управления и верификации: телевизионных камер, управления электроприводами, промышленных сетей, системы машинного зрения, интеллектуальных и IP-камер, приёмопередатчиков и модемов LTE, устройств медицинской диагностики и обработки изображений, многофункциональных офисных приборов, систем обработки видеосигналов и др. Естественно, что каждая задача требует своих особенностей проектирования SoC, например, связанных с выбором

(проектированием) СФ, решением проблемы надежности, с компоновкой функциональных узлов на кристалле и др. Однако существуют общие для всех задач особенности, к которым можно отнести:

1. Возможность использования достаточно широкой номенклатуры СФ-блоков, имеющих на рынке, которые могут быть реализованы на базе различных функциональных библиотек и технологий и интегрированы в кристалл средствами современных САПР. При этом СФ-блоки должны содержать элементы инфраструктуры системы (интерфейсы, системы питания и синхронизации, встроенные средства контроля и др.) [13], что определяет их функциональную и структурную законченность и возможность самостоятельной работы.

В принципе, в процессе проектирования SoC разработчик имеет возможность выбора следующих решений: самостоятельная разработка необходимых СФ-блоков; покупка СФ-блоков у ведущих разработчиков и производителей микросхем; поиск и применение СФ-блоков, предоставляемых в открытом доступе (например, по адресу <http://www.opensource> и др. источники).

2. Разбиение системы на аппаратную организацию (АО) с учетом СФ, программируемых модулей (ядер) и программное обеспечение (ПО).

3. Поскольку SoC включает одно или несколько программируемых процессорных ядер, разработчик должен принять решение о том, какие части поведенческой модели будут реализованы на аппаратном уровне, а какие – на программном в виде встроенного в СБИС программного обеспечения.

4. При реализации маршрута проектирования возникает необходимость выполнения совместной верификации и отладки программной и аппаратной частей на основных этапах проектирования: системном, функциональном, логическом и топологическом проектировании. Это объясняется тем, что SoC – это СБИС, в состав которой входят как аппаратная часть, так и программная, то есть встраиваемое программное обеспечение (ПО).

5. Также необходимо определить, каким образом будут взаимодействовать программная и аппаратная части, то есть следует разработать интерфейс между АО и ПО. Здесь же определяется общая архитектура SoC: тип процессора, тип памяти и ее объем, аппаратные блоки, интерфейс АО-ПО, тип используемой шины, описание программной части и т.п.

В итоге формируется набор спецификаций на разработку программного обеспечения и разработку каждого аппаратно реализуемого блока.

6. Особую сложность представляет взаимодействие цифровых подсистем с аналоговыми и СВЧ-схемами. В [14] для решения проблем верификации предлагается интеграция процесса создания программно-аппаратных средств тестирования с разработкой методологии сквозной верификации проекта на всех этапах.

7. Еще одна особенность SoC – наличие цифро-аналоговых систем в общем объеме SoC, поэтому в маршрут проектирования должны быть включены этапы по совместной разработке и верификации цифровой и аналоговой частей SoC.

8. Исходя из перечисленных свойств ПЛИС, можно сделать вывод, что их применение целесообразнее всего рассматривать в задачах, для которых не существует ASIC-решение (например, по таким параметрам, как габариты, вес, потребляемая мощность и др.), и вместе с тем ориентированных на однообразную интенсивную обработку потока данных, преимущественно параллельную или многоканальную.

9. Большой набор используемых для проектирования программных средств, зачастую подлежащих трудоемкому освоению и выбору. В мире создано множество программных средств, применимых для автоматизации системного проектирования. Эти средства можно классифицировать на пять групп [1, 2].

- Первая группа – средства разработки и отладки прикладного программного обеспечения. Достаточно удобным для представления спецификаций оказался язык программирования С и его модификация С++. Поэтому разработчики системного уровня активно используют такие средства разработки ПО, как Microsoft Visual Studio ([www.microsoft.com](http://www.microsoft.com)); Inprise Borland С++ Builder ([www.borland.com](http://www.borland.com)), а в ОС Linux – Borland Kylix С++ ([www.borland.com](http://www.borland.com)).

- Вторая группа – средства математического моделирования. Наиболее типичный представитель этой группы – программный пакет MATLAB/Simulink ([www.mathworks.com](http://www.mathworks.com)) С/С++.

- В третью группу можно объединить средства моделирования общего назначения, например, MLDesigner ([www.mldesigner.com](http://www.mldesigner.com)) или SES/Workbench ([www.hyperformix.com](http://www.hyperformix.com)). С их помощью можно моделировать архитектуру СБИС.

- К четвертой группе относятся узкоспециализированные программные средства, каждое из которых предназначено для решения какого-либо определенного круга проектных задач системного уровня. В качестве примера производителей данного ПО можно упомянуть такие фирмы, как Co-ware ([www.coware.com](http://www.coware.com)), Mentor Graphics ([www.mentor.com](http://www.mentor.com)), Elanix ([www.elanix.com](http://www.elanix.com)), Summit Design ([www.sd.com](http://www.sd.com)) и др.

- Пятая группа – это мощные интегрированные программные пакеты, при помощи которых разработчик способен выполнять весь цикл системного и функционального проектирования, а также весь цикл разработки СБИС, вплоть до физической реализации. На сегодняшний день в эту группу входят ПО двух фирм: Synopsys ([www.synopsys.com](http://www.synopsys.com)) – CoCentric System Studio, Design Ware, VCS, VCSi, Scirocco, SystemC HDL Co-Sim, CoCentric SystemC Compiler и Cadence Design Systems ([www.cadence.com](http://www.cadence.com)) – Incisive-SPW, Incisive unified simulator, Incisive-XLD, Incisive-AMS, NC-SystemC, NC-Verilog, NC-VHDL.

## 8. Выводы

Постоянный рост требований к радиоэлектронной аппаратуре со стороны заказчика как по функциональным, так и по эксплуатационным качествам, заставляет разработчиков создавать сложные устройства. В то же время значительный прогресс в области технологий производства СБИС (0,35 мкм и менее) позволил создавать чипы со степенью интеграции более 10 млн вентилей на кристалле [1–14]. В этом случае традиционные методы проектирования компьютерных систем, содержащих платы с микросхемами, стали не пригодными. Возникла необходимость в создании новых технологий и методологии, а также новых методов проектирования новой элементной базы, так как дальнейшее развитие электроники невозможно без внедрения новых прогрессивных решений с учетом их особенностей по сравнению с существующими методами. Одним из таких решений является технология «система на кристалле» [12], применение которой на основе ПЛИС позволяет выйти на новый уровень инженерных решений и может способствовать быстрому росту качества и функциональности продукции. SoC на базе FPGA могут стать основной элементной базой для широкой номенклатуры мало- и среднесерийных изделий с соответствующими требованиями к параметрам. В то же время процесс проектирования является достаточно сложным, так как требует одновременной верификации аппаратных и программных средств на каждом этапе маршрута проектирования.

Ожидается широкое использование в составе SoC процессорных СФ-блоков, предлагаемых ведущими производителями. При этом ряд проблем возникает при применении ПЛИС со встроенными процессорными ядрами. Тем не менее, развитие SoC на базе FPGA приведет к существенному сокращению доли проектов, реализуемых в виде систем на плате. Немаловажное значение для ПЛИС имеет их возможность реконфигурации [14–17] – перестройки на реализуемый алгоритм задачи пользователя, что обеспечивает существенное повышение производительности системы при улучшенных значениях некоторых дру-

гих параметров (например, габариты, вес и др.). Проект вычислительной системы (ВС), реализованный на ПЛИС, может быть выгодным благодаря меньшим затратам времени на разработку и производство такой ВС, которая, как минимум, в 2 раза длится быстрее, чем проектирование СБИС. Это обусловило широкое распространение ПЛИС как элементной базы SoC.

В отличие от традиционных систем на кристалле (SoC), программируемая логика Zynq-7000 позволяет легко изменять архитектуру системы, приспособив её под решение конкретной задачи при помощи специализированных периферийных устройств или модулей расширения.

## СПИСОК ЛИТЕРАТУРЫ

1. Евтушенко Н.Д. Методология проектирования систем на кристалле. Основные принципы, методы, программные средства [Электронный ресурс] / Евтушенко Н.Д., Немудров В.Г., Сырцов И.А. – Режим доступа: <http://www.mriprogress.msk.ru/news.php?id=7>.
2. Бухтеев А. Системы на кристалле. Новые тенденции / А. Бухтеев, В. Немудров // Электроника НТБ. – 2004. – № 3. – С. 52 – 56.
3. Бухтеев А. Методы и средства проектирования систем на кристалле / А. Бухтеев // Chip News. – 2003. – № 4 (77). – С. 4 – 14.
4. Шагурин И. Системы на кристалле. Особенности реализации и перспективы применения [Электронный ресурс] / И. Шагурин. – Режим доступа: <http://www.russianelectronics.ru/leader-review/2189/doc/40316/>.
5. Шагурин И.И. Создание «систем на кристалле» на основе ПЛИС с использованием синтезируемых процессорных ядер / И.И. Шагурин, В.А. Шалтырев // Проблемы разработки перспективных микроэлектронных систем: сб. научных тр. – М.: ИПИМ РАН, 2006. – С. 382 – 385.
6. Горбунов В.С. Применение программируемых реконфигурируемых схем при решении проблем технологий обработки больших графов [Электронный ресурс] / В.С. Горбунов, А.Г. Титов. – Режим доступа: [http://www.Rosta.ru/GraphHPC-2014\\_04-Gorbunov.pdf](http://www.Rosta.ru/GraphHPC-2014_04-Gorbunov.pdf).
7. Advantages of Xilinx 7 Series All Programmable FPGA and SoC Devices [Электронный ресурс]. – Режим доступа: <http://www.ni.com/white-paper/14583/en/>.
8. Платформа Zynq-7000. Очередной виток инноваций [Электронный ресурс]. – Режим доступа: <http://www.russianelectronics.ru/developer-r/review/2189/doc/57818/>.
9. All Programmable SoC with Hardware and Software Programmability [Электронный ресурс]. – Режим доступа: <http://Zynq-7000-All-Programmable-SoC.htm>.
10. Адамов Ю.Ф. Проектирование систем на кристалле [Электронный ресурс] / Ю.Ф. Адамов. – Режим доступа: [http://www.bmstu-sm5.narod.ru/puchkov/puchkov\\_lec.pdf](http://www.bmstu-sm5.narod.ru/puchkov/puchkov_lec.pdf).
11. Маршрут и методика проектирования микросхемы контроллера класса «системы на кристалле» для SD-карт стандарта SDHC [Электронный ресурс] / Д. Гречищев, Я. Губин, А. Руткевич [и др.]. – Режим доступа: [http://www.kit-e.ru/articles/circuit/2012\\_11\\_154.php](http://www.kit-e.ru/articles/circuit/2012_11_154.php).
12. Калачёв А. Многоядерная конфигурируемая вычислительная платформа Zynq-7000 / А. Калачёв // Современная электроника. – 2013. – № 1. – С. 22 – 31.
13. Тарасов И. Системы на кристалле на базе ПЛИС FPGA Xilinx со встроенными процессорами PowerPC. Ч. 2 [Электронный ресурс] / И. Тарасов. – Режим доступа: [http://kit-e.ru/articles/plis/2005\\_8\\_82.php](http://kit-e.ru/articles/plis/2005_8_82.php).
14. Непомнящий О.В. Проблемы верификации при проектировании систем на кристалле [Электронный ресурс] / О.В. Непомнящий, А.А. Шуплецов. – Режим доступа: [http://www.mrwolf.ru/Nauka\\_i\\_obrazovanie/Tochnye\\_nauki/9644](http://www.mrwolf.ru/Nauka_i_obrazovanie/Tochnye_nauki/9644).
15. Попович А. Применение технологии разработки систем на кристалле на платформе ПЛИС [Электронный ресурс] / А. Попович. – Режим доступа: [http://kit-e.ru/articles/plis/2004\\_4\\_114.php](http://kit-e.ru/articles/plis/2004_4_114.php).
16. Palagin A.V. Reconfigurable computing technology / A.V. Palagin, V.N. Opanasenko // Cybernetics and Systems Analysis. Springer New York. – 2007. – Vol. 43, N 5. – P. 675 – 686.
17. Палагин А.В. Реконфигурируемые вычислительные системы / А.В. Палагин, В.Н. Опанасенко. – К.: Просвіта, 2006. – 280 с.

*Стаття надійшла до редакції 12.04.2017*