

В заключение отметим, что, несмотря на чрезвычайную важность, задача защиты информации в сетях до сих пор весьма далека от своего решения. Большой шаг в этом направлении будет сделан при переходе на сетевые протоколы нового поколения TCP/IPv6. Вместе с тем, авторы, руководствуясь древней мудростью «Кто осведомлен – тот вооружен», будут удовлетворены, если пользователи Internet, прочитавшие данную статью, задумаются о том, насколько хорошо защищен их хост.

Мы хотели бы обратить внимание читателей на то, что прежде чем разработать и внедрять на практике политику сетевой безопасности, следует задать себе вопрос – насколько адекватна цена защищаемой информации тем затратам, которые будут иметь место при ее защите. Необходимо принять во внимание следующую аксиому: требования обеспечения безопасности сетевой информации всегда находятся в противоречии с такими сетевыми характеристиками как функциональность, быстродействие, удобство работы и т.п. Поэтому присту-

пать к осуществлению политики сетевой безопасности следует лишь хорошо взвесив все «за» и «против». При этом нужно проанализировать и определить возможность использования последних наработок в данной области. В частности тех, о которых шла речь выше.

Литература: 1. *Медведовский И.Д. и др.* Атака на Internet. Изд. ДМК, 1999. 2. *Nemchenko V., Schaff A.* Vulnerabilities and test of Internet protocols / Radioelectronika i Informatika. 2003. No.3 P.194-195.

Поступила в редколлегия 28.09.2004

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Немченко Владимир Петрович, канд. техн. наук, профессор ХНУРЭ. Научные интересы: техническая диагностика, сетевые технологии. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 702-14-17, e-mail: vrn@kture.kharkov.ua.

Дао Тху Фьонг, магистрант ХНУРЭ. Научные интересы: использование сетевых технологий для решения экономических задач. Адрес: Украина, 61166, Харьков, пр. Ленина, 14; тел. 702-18-05.

УДК 681.326

МЕТОДЫ И СРЕДСТВА СОЗДАНИЯ ЦИФРОВЫХ СИСТЕМ НА КРИСТАЛЛАХ. АНАЛИТИЧЕСКИЙ ОБЗОР

*ХАХАНОВ В.И., ЕГОРОВ А.А.,
КАМИНСКАЯ М.А., ХАХАНОВА И.В.*

Рассматриваются методы и средства создания и разработки цифровых систем на кристаллах. Приводится аналитический обзор существующих систем моделирования, верификации, синтеза и имплементации. Дается классификация чипов с позиции создания экономичных и высокопроизводительных цифровых систем. Показываются перспективные пути развития рынка электронных технологий и возможности их проникновения в Украину.

Введение

Цифровые системы (ЦС) управления, сбора, хранения, преобразования и анализа данных в настоящее время применяются во всех областях человеческой деятельности (производство, научные исследования, быт, офис, транспортные средства). В связи с этим к ЦС все более возрастают требования по уменьшению стоимости, размерности, энергопотреблению, аппаратурной избыточности. Последняя должна быть максимально подогнана под выполняемую задачу. Очевидно, что универсальные, мощ-

ные персональные компьютеры, потребляющие большое количество энергии и имеющие внушительные размеры и стоимость, не могут конкурировать на рынке специализированных устройств. Что же взамен?

Ответом на данный вопрос могут быть: Embedded Systems (встроенные системы), Systems on Chip (SoC – системы на кристаллах) [1], Networks on Chip (сети на кристаллах) [2]. Другими словами, чтобы добиться увеличения производительности, уменьшения потребляемой мощности и физических размеров, систему необходимо размещать не на плате, а на кристалле. При проектировании SoC проблема функциональной верификации и тестирования является одной из самых сложных: по оценкам ведущих зарубежных фирм и специалистов общества IEEE стоимость ее решения составляет 50 – 70% от совокупных затрат полного цикла разработки кристалла [3,4]. На рынке микроэлектроники для изготовления цифровых кристаллов наиболее популярными являются заказные БИС (cell based ASIC) и программируемые логические интегральные схемы – ПЛИС (PLD – CPLD, FPGA) [5]. Два других типа логических устройств постепенно уходят в прошлое – это SDL (дискретные логические микросхемы) и GA (вентильные матрицы) [6]. На базе ASIC и PLD существуют следующие направления разработки цифровых систем (рис. 1).

ASIC (Application Specific Integrated Circuits) – специализированные интегральные схемы, которые выигрывают у ПЛИС по энергозатратам и по производительности, но значительно проигрывают по времени разработки и производства. Кроме того, здесь следует учитывать высокую стоимость производства подложки для кристаллов ASIC, которая целесообразна только при больших партиях, начиная с 50000 микросхем [6].

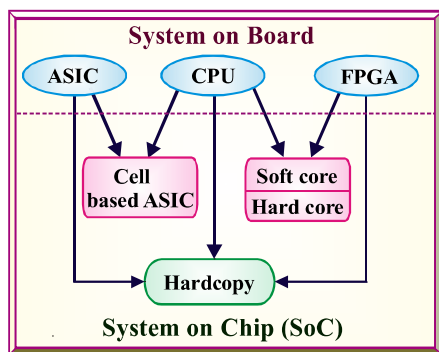


Рис. 1. Направления развития цифровых систем

Традиционно структурно-функциональное описание устройства на кристалле (ASIC или ПЛИС) представляется в виде множества взаимосвязанных логических примитивов [7]. Все автоматизированные средства верификации и тестирования были ориентированы на отладку только логических схем, представленных схематически или с помощью языков описания аппаратуры. Но когда речь идет о наличии микропроцессоров внутри системы на кристалле, то функциональность определяется не только логическими элементами, выполненными в кремнии, но и от программного обеспечения, реализованного в процессорах системы на кристалле [8]. Поэтому для верификации и тестирования таких SoC требуются совместные усилия разработчиков аппаратных и программных компонентов системы. Такие же требования предъявляются к моделям, методам, алгоритмам и средствам проектирования, которые должны быть в состоянии решать задачи совместного тестирования и верификации гетерогенных частей системы [9].

Учитывая сказанное, весьма актуальной представляется проблема разработки новых моделей, методов, алгоритмов и программно-аппаратных комплексов совместного тестирования и верификации на стадии проектирования цифровых систем и сетей на кристалле.

Цель аналитического обзора – предоставление начинающим разработчикам цифровых систем на кристаллах информации о перспективных направлениях имплементации моделей, методов и алгоритмов, которые могут быть востребованы на рынке электронных и информационных технологий.

Задачи: 1) Сравнительный анализ интегральной схемотехники, поставляемой на рынок EDA-технологий для реализации цифровых систем. 2) Перспективные маршруты проектирования, тестирования

и верификации создаваемых программно-аппаратных комплексов. 3) Анализ рынка электронных технологий и перспективы его продвижения в Украину.

1. Специализированные интегральные схемы (ИС) – ASIC

Это доминирующие кристаллы на рынке микроэлектроники, которые используются для реализации специализированных и универсальных цифровых и цифро-аналоговых схем. ASICs являются лидерами по вместимости кристалла, его производительности, по расходу электроэнергии и цене за единицу изделия при массовом производстве [6]. Существует несколько разновидностей ASICs, различаемых по архитектуре, а следовательно, по рабочим характеристикам и циклам разработки: Gate Array (GA) – вентиляционная матрица; Embedded Array (EA) – полужаказные ИС; cell based ASIC (Standard cell – SC) – заказные ИС со встроенными ячейками [10].

GA принадлежит к семейству ASIC и характеризуется незначительным (недели) временем проектирования и реализации на кристалле по самым низким ценам. Технология изготовления: 1,0 – 0,25 мкм; потребляемое напряжение 5 – 1,8В; количество доступных к использованию вентиляций – от 1 тыс. до 1,5 млн, число выводов от 20 до 680, рабочая частота до 133 МГц, количество слоев 2 – 4. Разработчик получает уже готовую подложку bulk wafers с матрицей транзисторов и с помощью средств проектирования задает связи транзисторов между собой (routing), что формирует функции чипа. Подложки производятся с различным количеством транзисторов, что дает потребителям свободу выбора по объему полезных функций, которые можно включить в кристалл. GA используется для создания специализированных больших интегральных схем (БИС). Здесь доступен ряд встроенной в кристалл памяти. Так, фирма NEC (www.nec.com) для GA предлагает RAM, ROM, синхронные и асинхронные, одно- и двухпортовые, размерностью до 8 Кбит [6]. Функциональные блоки (IP-cores) могут быть также использованы в GA. Но обычно это ограниченное количество малых по размеру ячеек. Их набор различен для производителей и семейств GA. Компанией NEC поддерживаются IP-cores: DMA-контроллер, блок управления, программный счетчик, блок прерываний, приемо-передатчик (UART с FIFO), PCI. EA сочетают положительные качества GA – малое время проектирования и возможность использования функциональности на системном уровне, как у Standard Cell [10]. Быстрая скорость проектирования достигается тем, что некоторые wafer изготавливаются параллельно процессу проектирования EA. Технология изготовления 0,6 – 0,18 мкм.

EA – это высокопроизводительная полужаказная ИС (вентиляционная матрица), которая включает в себя базовые матрицы транзисторов и позволяет использовать встроенные элементы памяти и IP-cores, как и у cell-based ИС [10]. При создании

проектов с большим объемом памяти полузаказные ИС являются более оптимальным решением по цене за единицу. Но при этом возрастает стоимость разработки из-за необходимости решения задачи размещения матрицы транзисторов на базовой подложке cell-based IC (специализированной ИС). Использование вентиляльного “моря” или бесканальной архитектуры ИС делает процесс разработки и последующее соединение транзисторов практически равным длительности фазы проектирования SoC в GA. Монолитную подложку VLSI (БИС) можно повторно использовать, изменяя только логическую часть во время проектирования и верификации, как и у кристаллов технологии GA.

SC – заказные ИС, которые позволяют оптимально проектировать логические ячейки, блоки памяти ROM и RAM, CPU и аналоговые компоненты на одном кристалле. SC предоставляют разработчикам более гибкую технологию проектирования по сравнению с GA и могут быть оптимизированы под нужды заказчика LSI. Технологии изготовления 0,35 – 0,18 мкм. В настоящее время уже заявлены технологии 0,13 – 0,09 мкм. Такая оптимизация позволяет достичь еще большей функциональности чипа и меньшего потребления электроэнергии в процессе эксплуатации.

Различие между типами ИС заключается в количестве сложных и простых функциональных компонентов, блоков или ячеек на кристалле, из которых строится цифровое устройство [10]. Так, для GA это транзисторы и некоторые элементы памяти; для заказных высокоинтегрированных ИС – это компоненты с расширенной функциональностью, позволяющие разрабатывать SoC. На рис. 2 приведены некоторые функциональные модули, которые могут быть интегрированы в кристалл (интерфейсы, процессоры, периферия, блоки памяти, аналоговые компоненты).

Учитывая существующие технологии производства и архитектурные особенности, используемые при построении систем на кристаллах, можно построить диаграмму (рис. 3), которая позволит визуально наблюдать разницу между GA, EA и SC при их использовании в качестве элементной базы. Характеристики взяты на сайте фирмы EPSON, одного из мировых лидеров по производству микросхем.

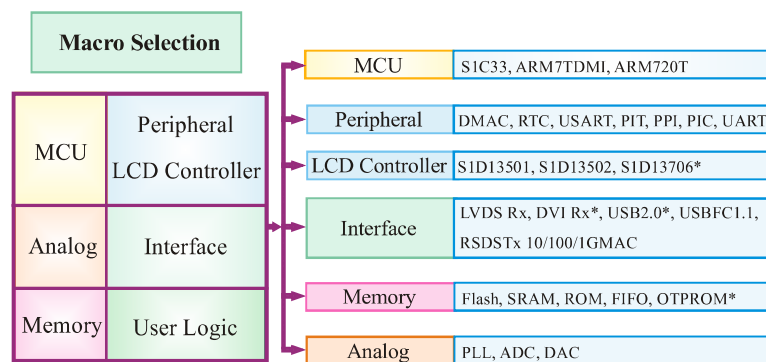
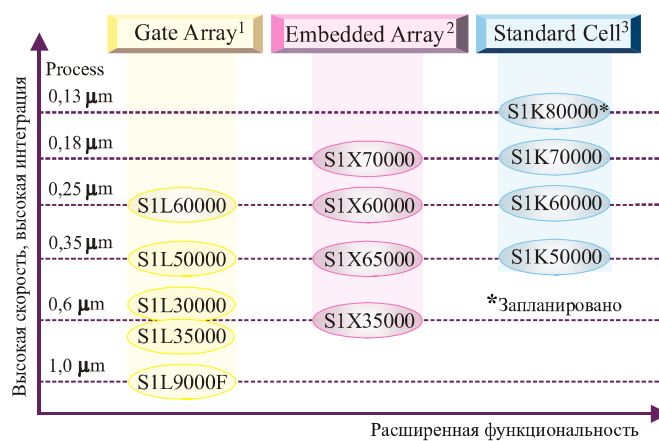


Рис. 2. Перечень функциональных компонентов фирмы EPSON



- 1 Мощные ИС с быстрым циклом проектирования и низкой стоимостью
- 2 Сочетают лучшие качества GA и standard cell IC
- 3 Высокая интеграция и расширенная функциональность

Рис. 3. Связь технологий проектирования с кристаллами в фирме EPSON

1.1. Технологии и маршруты процесса проектирования

На рис. 4 приведена схема полного цикла разработки GA и полузаказных микросхем. На первом этапе – системный уровень проектирования – определяются параметры и компоненты SoC и параллельно начинается разработка модулей RTL уровня и изготовление базовой подложки, на которой будут размещены все компоненты цифрового устройства [11]. После верификации модели системы выполняется имплементация проекта путем переноса (плавки) требуемых связей.

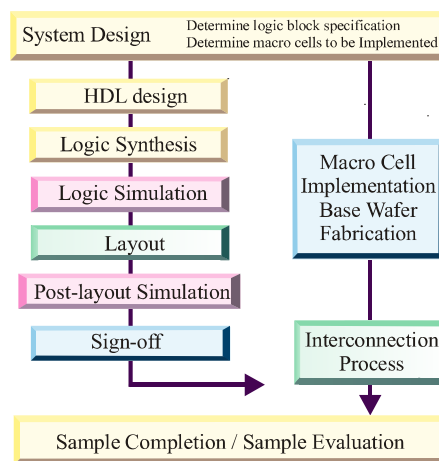


Рис. 4. Цикл разработки полузаказных ИС

На рис. 5 представлен процесс проектирования заказной ИС, где отличие заключается в отсутствии производства подложки кристалла. Разработка начинается с системного уровня, во время которого подсчитывается количество вентиля и макроячеек (macro cells). Затем под данный проект осуществляется производство основной пластины кристалла (www.epson.com). Далее на кристалле конструируются макроячейки путем превра-

щения вентиляльного “моря” методом плавки соединений в логические блоки требуемой функциональности. Как и в случае с GA, параллельно с производственными операциями выполняются стандартные этапы проектирования, начиная с описания проекта и заканчивая моделированием конечного Netlist и исправлением HDL описания, если это необходимо.

ASIC development flow and support tools

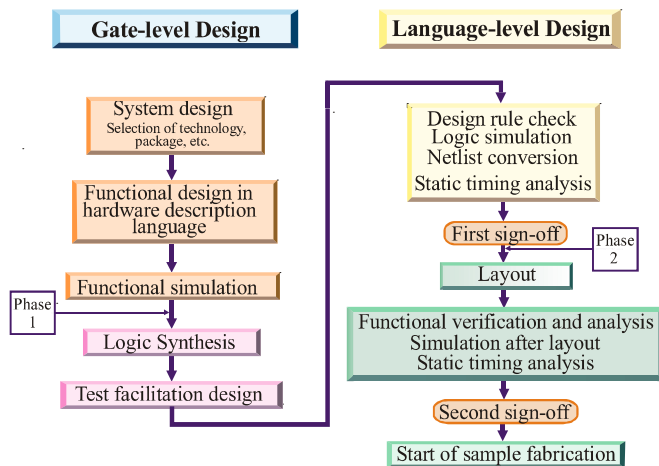


Рис. 5. Процесс разработки и верификации ASIC

Затем рабочая версия проекта “замораживается” и она же имплементируется в кристалл путем плавки соединений между транзисторами на подложке. После “заморозки” кода Netlist создаются пробные экземпляры микросхемы. Поскольку базовая подложка обладает свойством повторного использования, то стоимость изменения логики на кристалле аналогична затратам использования GA-технологии [10]. Представленную (см. рис. 5) схему разработки ASIC-проекта можно упростить до четырех этапов, показанных на рис. 6.

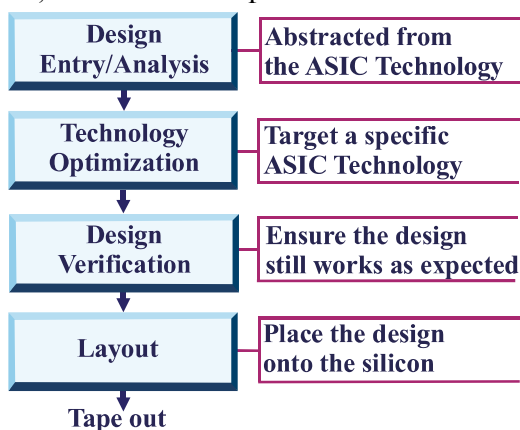


Рис. 6. Упрощенный процесс проектирования ASIC

Проектирование на вентиляльном уровне. Ввод схемного описания — это традиционно времяемкая фаза проектирования, которая используется как для проектирования ИС, так и для программируемых логических устройств. Для ввода схемного описания применяется GUI (Graphical User Interface) —

графический интерфейс, позволяющий выбрать необходимое количество и типы вентиля и/или примитивов и реализовать связи между ними. При таком подходе к проектированию можно выделить 4 основных этапа (www.k-micro.com) (рис. 7): 1) Выбор соответствующего пакета проектирования и библиотеки семейства ASIC, куда предполагается имплементировать проект. 2) Создание проекта устройства с использованием соответствующих вентиля и примитивов из выбранной библиотеки и связей между ними. 3) Добавление в проект входных и выходных (I/O) контактов устройства. 4) Генерация конечного Netlist с расположением логических примитивов на кристалле и связей между ними. Netlist является компактным текстовым универсальным форматом представления схемы в примитивах библиотеки выбранного кристалла, который поддерживается языками Verilog, VHDL и может быть использован как для имплементации, так и для timing verification. Netlist сменил формат EDIF, применяемый только для фазы Place and Route. Проектирование цифровых систем на уровне вентиля оправданно, если размерность устройства не превышает 20000 вентиля [7]. Другим недостатком такого проектирования является невозможность имплементации проекта в другой кристалл, поскольку необходимо заменять все примитивы одного производителя на элементы из библиотеки другого поставщика микросхем.

PLD Design Flow

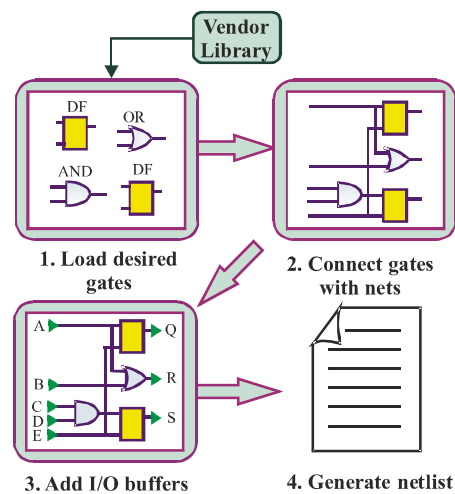


Рис. 7. Маршрут проектирования PLD

Проектирование на алгоритмическом уровне (High-Level Design). Здесь используются языки описания аппаратуры (hardware description language – HDL) для представления функционального поведения проекта без идентификации структуры. Основными стандартизированными HDL являются VHDL, System C и Verilog (www.cadence.com, www.magma.com). Идет работа по стандартизации System Verilog (www.accelegra.org). Языки описания аппаратуры являются идеальным средством, позволяющим повторно использовать готовые проекты или их части. После описания устройства на поведенческом уровне должен быть выполнен его синтез в схемное представление вентиляльного уровня, обеспечивающее заданную функциональность и

имплементацию в выбранный тип кристалла. Преимущества HDL-описания: это удобное и быстрое написание проекта; легкое внесение изменений; исключение процедуры соединения схемных блоков между собой; возможность получения нескольких альтернативных решений в зависимости от параметров синтеза; выбор различных типов кристаллов после Design Entry. Реально для проектов более 10000 вентилей нет альтернативы языку описания аппаратуры.

Оптимизация синтеза под технологию изготовления кристалла. HDL описание цифровых систем, как правило, не содержащее вентилей, должно быть преобразовано в логическую схему примитивов выбранной серии кристаллов. Данная процедура выполняется программами синтеза (Synopsys – Design Compiler). Результирующий netlist после такой операции зависит от библиотеки выбранного кристалла (FPGA, CPLD, ASIC) и от критериев оптимизации (размерность проекта или быстродействие). Автоматизация синтеза позволяет легко экспериментировать с различными параметрами оптимизации, сериями микросхем и их производителями для получения нескольких вариантов решений, в то время как на вентиляльном уровне проектирования существует один вариант (www.synopsys.com). Второй задачей синтеза устройства под выбранную технологию изготовления кристалла является обеспечение тестопригодности. Применяемая в настоящее время тестовая методология внесения нефункциональной избыточности (IEEE JTAG boundary scan 11.49, P1500) [12] при разработке SoC является стандартом для большинства производителей и лабораторий. Она включает имплементацию на кристалл средств управления и наблюдения в режиме тестирования функций (Synopsys DFT Compiler), Built-in Self Testing (BIST) для проверки памяти, логики, неисправностей, таких как single stack-at-fault, Iddq (Synopsys TetraMAX, Cadence/IBM TestBench) [13–16]. Для определения производственных дефектов в ASIC вводится регистр сканирования данных (рис. 8), который существенно повышает качество (до 95–100 %) теста и уменьшает его длину. При этом аппаратура сканирования незначительно влияет на производительность системы и увеличение площади кристалла. На рынке существуют десятки автоматизированных пакетов, выполняющих полный цикл Design for Test, единственным недостатком которых является высокая стоимость и использование дорогих внешних тест-станций, например, от фирм Advantest и HP (www.advantest.com, www.hp.com). После тестопригодного синтеза разработчик получает устройство на основе примитивов выбранного производителя и серии кристалла, включающее функциональность и блоки для тестирования производственных дефектов, готовые к имплементации на кристалле. Но перед выполнением последней процедуры необходимо верифицировать проект на правильность работы в целях устранения функциональных ошибок, которые

могут привести к нежелательному повторению фазы Place and Route.

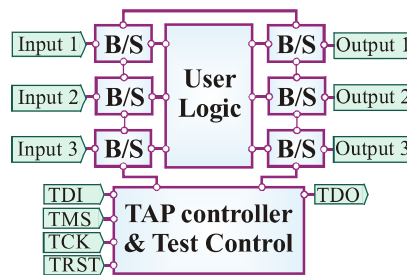


Рис. 8. Применение стандарта IEEE 1149.1 для SoC Design

Верификация проекта (Design Verification). Здесь основные фазы и программные средства для их реализации можно перечислить в следующем виде: 1) Logical Design Rule Check – Synopsys (DC Ultra, Design Compiler), K-Microtools; 2) Formal Verification – Synopsys (Design VERIFYer, Formality), Verplex (Tuxedo-LEC); 3) Static Timing Analysis – Synopsys (Prime Time); 4) Gate-Level Simulation – Mentor (ModelSim), Cadence (VerilogXL, NC-Verilog), Synopsys (VCS); Cadence Products are golden simulators for K-Micro; 5) Power Estimating (if necessary) – Synopsys (Power Compiler) [17–19]. С ростом микропроцессорных технологий размерность ASIC систем увеличивается с каждым годом [20]. Вместе с тем, для проверки всех временных характеристик мультимиллионного проекта необходимо синтезировать тест большой размерности (сотни тысяч векторов), что является трудновыполнимой задачей в рамках ограниченного времени проектирования ASIC [21]. Тем не менее на рынке появляются средства – static timing analysis (STA), позволяющие выполнить анализ временных характеристик за короткий промежуток времени, если проект выполнен по STA технологии [22–24]. Следует отметить высокую стоимость специализированных средств оценки статических временных параметров проекта после синтеза/имплементации (Synopsys PrimeTime). Это объясняется большими затратами переделки подложки в случае ошибки.

Formal Verification – FV (Verplex). Средства формальной верификации предназначены для статической проверки на соответствие текущего проекта уже отлаженному его представлению на предыдущей стадии разработки [25]. При проектировании с использованием HDL средства формальной верификации необходимы для проверки эквивалентности верифицированного RTL проекта с послесинтезным вентиляльным Netlist. В общем случае без логического моделирования за короткий промежуток времени можно определить эквивалентность проекта на различных стадиях разработки. Сочетание технологий STA и FV позволяет значительно повысить производительность проектирования ASIC большой размерности. Популярные маршруты, сочетающие моделирование, верификацию и временной анализ, представлены на рис. 9.

Планирование проекта в кристалле (Layout). Здесь основные фазы и существующие программные средства для их реализации можно представить в следующем виде: 1) Floor Planning – Cadence (First Encounter); 2) Power Analysis – Synplicity (Fortify RealPower, Fortify Power Planner); 3) Clock Tree Synthesis [26, 27] – Cadence (CTGen, First Encounter); 4) Place & Route [28, 29] – Cadence (Silicon Ensemble DSM, First Encounter + NanoRoute Ultra); 5) Interconnect RC Analysis – Synopsys (Star-RCXT); 6) Noise Analysis [30, 31] – Sequence (Physical Studio); 7) ATE Rule Check – K-Micro tools; 8) DRC & LVS – Mentor (Calibre) (www.mentor.com); 9) Static Timing Analysis – Synopsys (Prime Time, Customer Sign-Off); 10) TestProgram Generation – K-Micro tools [32].

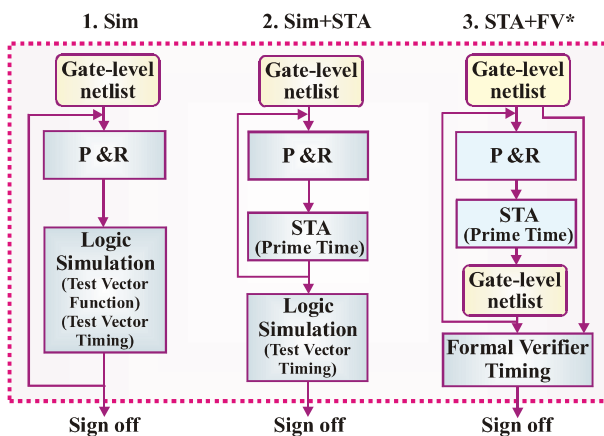


Рис. 9. Маршруты верификации проектов

Иерархический процесс проектирования. Используется для ASIC большой размерности и позволяет уменьшить количество итераций, а также время выполнения проекта. Внимание уделяется совместной работе команд: 1) разработка RTL проекта; 2) размещение его на кристалле, начиная со стадии Design Entry. При этом возникают новые задачи: hands-on design partitioning – проект разделяется на блоки с учетом функциональности, структуры проекта, его размерности и количества блоков (size balancing of partitions) [33]; оптимизация параметров внутри каждого модуля, анализ временных и синхронизирующих характеристик блоков; распределение и планирование бюджета времени по разработке блоков (timing budget allocation for the partitions); интегрирование блоков и последующая оптимизация, а также верификация всех основных параметров в рамках объединенной системы.

Выводы. На рис. 10 приведены характеристики GA и cell-based IC по времени производственного цикла [6]. Разработка RTL модели одинакова для обеих технологий, но GA выигрывает по совокупному времени (тестирование + производство). Данное обстоятельство уменьшает производственный цикл почти вдвое по сравнению с cell-based IC. Для GA, EA, cell-based IC в отличие от PLD добавляется большой блок – тестирования, занимающий до половины временных и материальных затрат полного цикла проектирования [12].

Для проектов большой размерности используются мощные и дорогие инструменты логического моделирования (фирмы Cadence, Synopsys), аппаратные ускорители, поскольку время одного цикла моделирования может достигать несколько дней (неделю) при регрессивном тестировании (Axis, Mentor Graphics, Cadence), когда верификация проекта осуществляется с возвратом от сложных тестов к простым при обнаружении дефекта.

Total TAT Comparison (from RTL to SAMPLE)

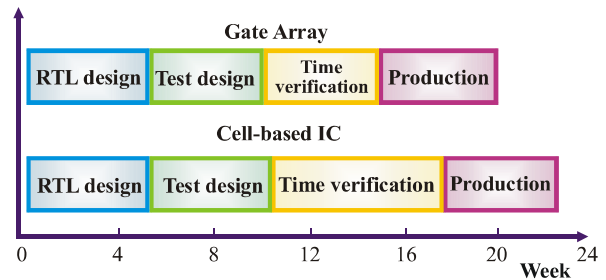


Рис. 10. Анализ производственных циклов

2. Программируемая логика (PLD)

В конце 70-х разработчики использовали стандартные логические микросхемы и размещали их на печатных платах для создания цифровых систем. Затем рынок потребовал интегрировать функции микросхем на одном кристалле для создания более сложных, компактных и быстродействующих устройств. Чтобы достичь гибкой функциональности, Рон Клинтон из Signetics (которая позднее была куплена Philips, а затем Xilinx) предложил использовать программируемые матрицы (Programmable Logic Array – PLA) (рис. 11), обеспечивающие любую комбинацию И/ИЛИ элементов [7]. Такая архитектура является гибкой, но при использовании 10 гм технологии задержка распространения сигнала значительна, что делает устройство относительно медленным.

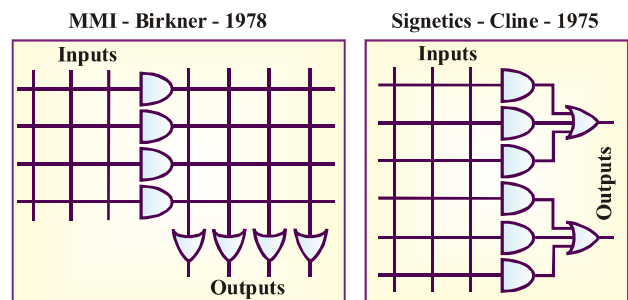


Рис. 11. Архитектура PLA и PAL

Фирма MMI, позже купленная AMD, является вторым источником появления программируемых матриц (Programmable Array Logic – PAL) (рис. 11, правая часть). Матрица элементов ИЛИ фиксированная. Предложенная PAL архитектура имеет преимущества в скорости и в более простом использовании программного обеспечения, но проигрывает в гибкости архитектуры перед PLA. Появились и другие архитектуры, развивающие данное направление. Семейство Simple PLD состоит из сети вертикальных и горизонтальных связей. При

программировании чипа пересечения связей становятся соединениями или остаются разомкнутыми. Входы кристалла подключаются к вертикальным связям матрицы – элементам И-ИЛИ, соединенным с соответствующими триггерами, выходы которых связаны с внешними контактами. Стало возможным программировать кристалл через порт ISP. PLD архитектура позволяет разместить в 50 раз больше вентилях на кристалле, чем на плате, при низком уровне энергопотребления (Xilinx) и возможности перепрограммирования логики устройства.

CPLD – Complex Programmable Logic Device – совокупность функциональных блоков (PAL), объединенных матрицей переключений (Interconnection wires), увеличивает плотность SPLD устройств (www.xilinx.com) с концепцией объединения нескольких PLD блоков на одном кристалле (рис. 12). Каждый PAL-блок имеет n двунаправленных выводов, а также m выводов, которые управляются ячейками ввода/вывода (I/O block), имеющими связи с матрицей переключений.

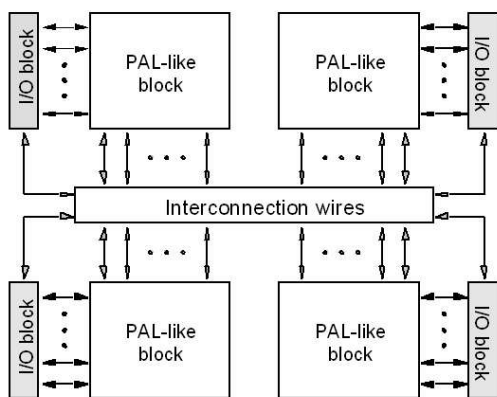


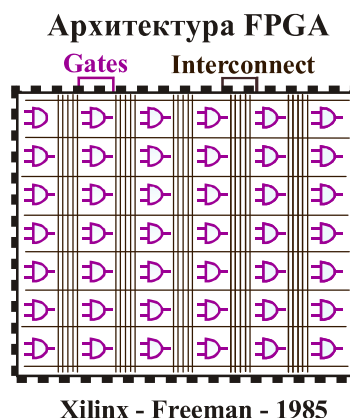
Рис. 12. Обобщенная структура микросхемы CPLD

Преимущества CPLD: 1) Просты в использовании. Достаточно разработать описание устройства на одном из стандартизованных языков описания аппаратуры. Затем с помощью средств САПР оптимизировать (под выбранное семейство и серию CPLD кристалла), верифицировать и разместить на кристалле логику проекта. 2) Низкие затраты при разработке. Разработчик CPLD может с минимальными временными и материальными затратами изменить проект или добавить новую функциональность после программирования чипа, а также верифицировать его на рабочих частотах (200 МГц). 3) Низкие цены на пакеты проектирования (лидер по производству кремния Xilinx предоставляет их бесплатно), а также невысокая стоимость цикла перепрограммирования. 4) Малый период разработки (Time-to Market) – конечный продукт быстро попадает на рынок и приносит больше прибыли. 5) Свойство перепрограммируемости позволяет реконфигурировать устройство через Internet, используя ISP (Internet Service Provider), что дает возможность усовершенствовать существующий продукт заменой прошивки кристалла, продлевая жизнь устройству для получения дополнительной

прибыли. 6) Большой уровень интеграции вентилях на единице площади, что идеально для проектировщиков, которые ранее были ограничены размером печатной платы.

FPGA архитектура (рис. 13). В 1985 г. компания Xilinx представила новый кристалл с параметрами разработки по времени и контролю дизайна как у PLD, а размерностью и плотностью как у интегральных схем (gate arrays). FPGA – это регулярная структура из логических ячеек/модулей и межсоединений, находящаяся под полным контролем разработчика, который может создавать, программировать и вносить изменения в аппаратную часть в любой стадии проектирования и верификации. FPGA кристаллы имеют размерность – 10 млн вентилях (Xilinx Virtex-II).

Разработаны 2 типа FPGA – перепрограммируемые на SRAM и OTP (One Time Programmable), программируемые только один раз (рис. 14). Последние отличаются процессом имплементации логических ячеек и механизмом организации связей между ними. Популярным является многократно перепрограммируемый, каждый раз при подаче питания, FPGA кристалл с SRAM памятью. Поэтому FPGA можно назвать разновидностью микросхемы памяти. При каждом FPGA SRAM необходимо иметь последовательную PROM память для хранения bit файла, который определяет конфигурацию FPGA.



Xilinx - Freeman - 1985

Рис. 13. Структура микросхемы FPGA

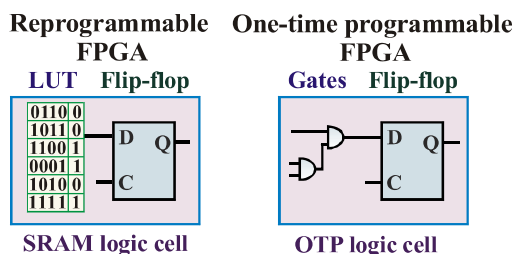


Рис. 14. Структура двух типов FPGA

В FPGA OTP типа используются плавкие перемычки для создания постоянных связей в кристалле в процессе программирования чипа. Поэтому OTP FPGA не требует наличия SPROM или другого устройства для программирования FPGA. При каждом изменении проекта уже запрограмми-

рованный чип выбрасывается. OTP архитектура сходна с PLD, которая также состоит из логических и триггерных примитивов. В FPGA на основе SRAM логический элемент представлен не в виде традиционных логических элементов, а в виде LUT (Look Up Table)-структуры, которая определяет выходное значение в зависимости от входного слова. SRAM биты используются для определения соединений. На рис. 15 приведена общая структура чипа FPGA, состоящая из блоков ввода/вывода (I/O block), матрицы настраиваемых логических блоков – Configurable Logic Block (CLB), программируемых горизонтальных и вертикальных межсоединений (Interconnection switches). Блоки I/O предназначены для согласования внутренних и внешних уровней логических сигналов; CLB – для реализации логических функций, задаваемых пользователем. Настройка каждого CLB определяется программированием функциональной таблицы (LUT). CPLD и FPGA составляют схемотехническую основу современной программируемой логики. Основное различие между CPLD и FPGA состоит в архитектуре, дающей возможность перепрограммировать функциональность кристалла, а также в количестве вентилях и стоимости. Процессы проектирования CPLD и FPGA практически не имеют различий и выполняются с помощью одинаковых программных пакетов (Xilinx® ISE, Altera Quartus).

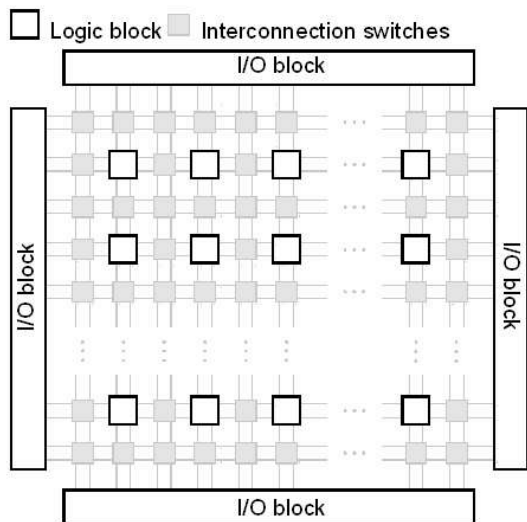


Рис. 15. Общая структура чипа FPGA

Процесс проектирования. Согласно опросу, проведенному на самой большой в мире шоу-конференции EDA (Electronic Design Automation) технологий DAC-2004, проходившей в Сан-Диего, Калифорния, 60% рынка по производству FPGA занимает фирма XILINX (годовой оборот 1,6 миллиарда долларов) и около 35% – ALTERA (годовой оборот – 993 миллиона долларов) (www.xilinx.com, www.altera.com). Поэтому при рассмотрении существующих средств проектирования SoC на PLD необходимо исследовать программные пакеты упомянутых фирм-лидеров по производству программируемых кристаллов. Появление на EDA-рынке программных продуктов Xilinx® ISE и Altera® Quartus намного упростило процесс разработки проектов для программируемой

логики на основе высокоуровневых языков описания аппаратуры (VHDL, Verilog). Общая схема проектирования, включая задачи верификации, представлена на рис. 16.

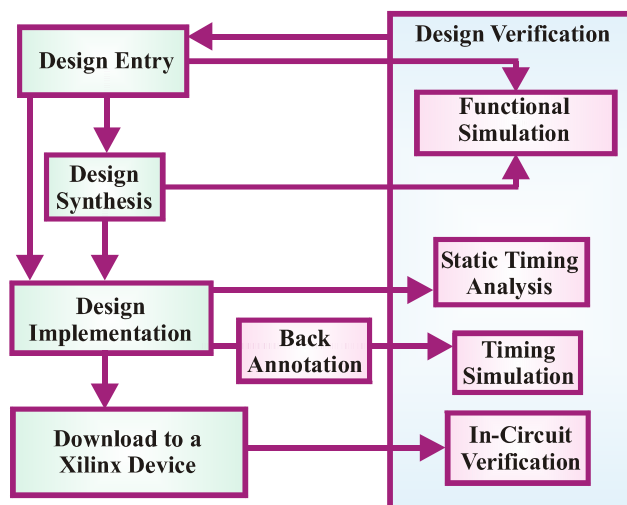


Рис. 16. Общая схема проектирования PLD SoC

Процесс разработки PLD SoC включает шаги: 1) Design Entry – описание проекта на схемном или вентиляльном уровне, или с помощью HDL-языков [34, 35]. 2) Синтез – преобразование исходного описания в вентиляльный Netlist выбранного семейства CPLD, FPGA. 3) Имплементация – размещение цифрового проекта на кристалле (тип корпуса, рабочая частота, другие параметры) с прокладкой межсоединений между блоками. Netlist проекта есть описание устройства вентилями определенного семейства кристаллов. После проверки на функциональность Netlist готов к имплементации на кристалле: оптимизация, трансляция в физические элементы чипа, проверка дизайна на достаточность компонентов, CLK буферов в выбранном чипе. При завершении имплементации генерируется отчет, где помимо предупреждений (warnings) и ошибок содержится информация об использовании блоков, входов/выходов, что позволяет определить оптимальность размещения проекта в кристалле.

3. Верификация CPLD, FPGA кристаллов

Верификация CPLD, FPGA проекта – это проверка правильности функционирования объекта путем анализа его реакций, полученных в результате его моделирования на тестовых воздействиях (Testbench) [36–38]. Критерии оптимальности верификации: 1) покрытие тестовыми наборами всех значений сигналов на линиях схемы; 2) покрытие тестом всех строк HDL кода; 3) покрытие тестом true-false ветвей в операторах условия; 4) полное соответствие устройства спецификации проекта; 5) покрытие тестом компонентов гетерогенной системы, поскольку существующие методы проверки RTL-проекта слабо ориентированы на верификацию SoC с программным обеспечением; 6) проверка на соответствие спецификации и корректности функционирования SoC на всех уровнях представления проекта [39].

Главное условие для выполнения верификации – наличие стандартизированных форматов данных, которые позволяют: 1) многократно использовать (re-use) уже отлаженные проекты или его части; 2) применять программы моделирования, синтеза и имплементации различных производителей без изменения описания проекта, который может быть перекомпилирован с подключением других библиотек примитивов для достижения более оптимального результата. Верификация практически на всех стадиях прохождения проекта означает моделирование на основе его описания и входных тестовых последовательностей (Testbench). Результат выполнения процедуры – временные диаграммы значений внутренних и выходных линий, которые необходимо проверить на соответствие критериям оптимальности верификации. На рис. 16 в правой колонке представлены этапы верификации: 1. Functional Simulation [40] – проверяется правильность введенной модели на функциональность относительно исходной спецификации. Выполняется как для HDL-описания, так и для вентиляционного уровня, чтобы убедиться – синтез прошел без ошибок и функциональность не изменилась при переходе от HDL к вентиляционному представлению проекта. 2. Static analysis – после имплементации проекта появляется информация о временных параметрах схемы. В случае обнаружения ошибок (недопустимо большие задержки) можно изменить HDL код, перекомпилировать его и выполнить повторное моделирование для проверки функциональности. Затем выполнить синтез вентиляционного представления (Netlist) из HDL-кода и провести моделирование в целях определения корректности измененного проекта. Итеративная верификация и коррекция проекта до устранения всех дефектов занимает порядка 50% времени доведения проекта до требуемого уровня качества. 3. Timing verification – после имплементации вентиляционного представления выполняется моделирование Netlist проекта с учетом всех временных характеристик элементов и связей между ними в целях определения правильности функционирования будущего устройства. 4. In circuit verification – последний этап верификации SoC на рабочих частотах после программирования CPLD, FPGA. Используется прототип платы со стандартным набором компонентов и интерфейсом для связи с рабочей станцией, которая эмулирует внешнюю среду для SoC. Debug проекта на этом этапе верификации проблематичен из-за ограниченного доступа к внутренним линиям CPLD, FPGA кристалла для выявления и локализации ошибок (www.avnet.com).

Выводы. ПЛИС технология исключает стадии производства и тестирования силикона. PLD-кристаллы покупаются уже готовыми и свободными от дефектов производства. Создать SoC – значит запрограммировать кристалл под требуемую функциональность, исключая ошибки проектирования. Поэтому верификация проекта путем моделирования структурного HDL описания, вентиляционного

Netlist, временного описания параметров логических блоков выбранной серии FPGA или CPLD является преобладающей по материальным и временным затратам при проектировании SoC. Автоматизация этого сегмента работ является наиболее важной проблемой в мире на сегодняшний день. Более 90% всех PLD-проектов имеют размерность менее 1 млн эквивалентных вентиля с бюджетом, не превышающим 50 000 долларов. Поэтому EDA-рынок для ПЛИС отличается более медленными и дешевыми программами моделирования, достаточно простыми при их использовании (Mentor Graphics, Model Sim[®] and Aldec, Active-HDL[®]).

4. Intellectual Property (IP-) Cores

IP-Core – сложный уже протестированный, верифицированный и оптимальный функциональный модуль, который многократно используется при проектировании различных устройств для сокращения времени их разработки [41]. Он присоединяется к проекту на системном уровне при выполнении этапа Design Entry. Преимущества использования IP-Cores: 1) существенное уменьшение времени до выхода конечного продукта на рынок; 2) упрощение процесса проектирования путем создания интерфейса для связи системы с IP-Core; 3) минимизация риска проектирования благодаря включению уже отлаженных модулей с готовым Testbench; 4) уменьшение времени верификации всей системы; 4) предсказуемость функциональности и производительности создаваемого проекта; 5) упрощение спецификации проекта, благодаря использованию полностью готовых к применению модулей системного уровня (PCI шина, DSP фильтры) [42].

Интеллектуальная собственность. Цифровые IP-Cores, которые могут быть легко интегрированы в любую ASIC технологию, уже доступны из многочисленных источников, представленных в Internet. Но более ценными и эффективными являются решения, которые зависят от технологии производства чипа (аналоговые, смешанные сигналы, PNYs, CAM) и максимально используют все ее преимущества – value-added IP (VA-IP). Такие VA-IP предоставляют следующие преимущества проектировщику: 1) уменьшение затрат проектирования путем исключения фазы разработки функциональных блоков VA-IP; 2) уменьшение времени создания проекта благодаря возможности использования стандартных силиконовых технологий.

Формат и свойства предоставляемых VA-IP: 1) VHDL/Verilog модели для моделирования устройств во время проектирования; 2) тест-чипы для верификации правильности работы IP в силиконе; 3) гарантия корректности временных параметров при расположении IP-модулей на кристалле; 4) статистический анализ временных параметров модулей для проверки производительности; 5) высокое покрытие неисправностей поставляемыми Testbench (www.viragelogic.com).

Конвертация FPGA в ASIC. При развитии технологий производители кристаллов предлагают вла-

дельцам FPGA решений переходить на ASIC-технологии [43]. Конвертация в ASIC приносит следующие преимущества: 1) уменьшение энергопотребления; 2) снижение стоимости производства одной микросхемы при больших объемах выпуска микросхем; 3) уменьшение занимаемой площади на плате путем создания одного ASIC кристалла вместо нескольких FPGA; 4) исключение затрат на PROM, необходимых для хранения конфигурации FPGA. Для принятия решения о переходе с FPGA на ASIC-технологии следует учитывать моменты: 1) покрое ли экономия на стоимости производства партии микросхем временные и материальные затраты смены технологии; 2) выбор производителя и типа ASIC: менее дорогую EA с вентиляльным “морем” и ограниченным набором встроенных в кристалл компонентов или более производительную, но дорогую технологию Cell based ASIC.

При переходе на ASIC-технологии файл схемного описания FPGA-проекта со всеми специфическими FPGA-конструкциями конвертируется в эквивалентные ASIC-структуры (CoreGen/LPM/on-chip multipliers, PLL, DLL, DCM, DCI/XCITE/Terminator, programmable slew-rate I/Os, встроенная на кристалле RAM, on-chip CAM (content addressable memory – ассоциативная память)). После конвертации необходимо определить соответствие полученного ASIC исходному FPGA путем использования полных Testbenches для функциональной верификации, которых, как правило, для небольших FPGA проектов у заказчиков нет. Другим решением функциональной верификации полученного ASIC может быть использование пакета CoreSim, который автоматически выполняет следующие процедуры: 1) генерация векторов для рабочего кристалла FPGA и определение выходных реакций для последующей проверки FPGA netlist на идентичность поведения; 2) проверка временных параметров ASIC, которые не должны превышать их аналогов в FPGA или на печатной плате; 3) тестирование ASIC для проверки правильности его функционирования путем подачи полученных входных последовательностей на кристалл.

Как рынок EDA решает задачу конвертации FPGA в ASIC? Следует выделить одного из лидеров на рынке FPGA-компанию Altera. Для последних наиболее производительных серий FPGA (Stratix, Stratix II, APEX 20KС и APEX 20KE) фирма предлагает создание кристалла с твердой логикой и со структурой проекта FPGA – Hardcopy ASIC, причем без дополнительных затрат на доработку (design for test, замена FPGA примитивов на ASIC). Altera гарантирует: увеличение производительности в среднем на 50%, снижение энергопотребления на 40% по сравнению с FPGA-кристаллом, успех при имплементации проекта в первый кремний; сохранение FPGA-архитектуры и netlist, что гарантирует отсутствие изменений в функциональности цифровой системы; предоставление двух- и трехуровневой металлической технологии в зависимости от серии FPGA. Существует несколько вариантов конвертации FPGA в ASIC (рис. 17): 1) “1 to 1” –

проект из FPGA переносится в ASIC без изменений, но при этом для потребителя уменьшается стоимость микросхемы при массовом производстве; 2) “n to 1” – интеграция системы, состоящей из нескольких кристаллов FPGA, в проект на одном кремнии дает выигрыш в скорости, габаритах, энергопотреблении; 3) “n in 1” – перенос нескольких независимых кристаллов FPGA с платы в один кристалл ASIC, каждый из FPGA будет динамически адресуемым после имплементации и изготовления.

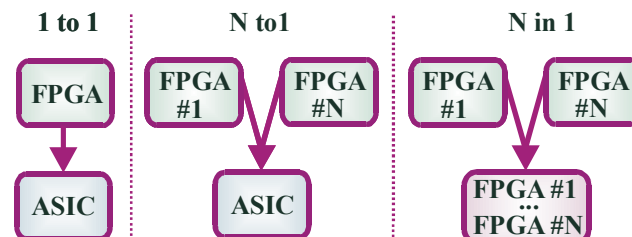


Рис. 17. Варианты конвертации FPGA в ASIC

Выводы. Автоматизация проектирования цифровых систем на кристаллах достигла уровня, когда ASIC design flow по затратам инженерного труда приближается к FPGA flow (RTL-flow), когда без вмешательства проектировщика можно выполнить проект на FPGA- или на ASIC-кристалле.

System-on-Chip [44]. Основные преимущества системы на кристалле по отношению к ее аналогу, размещенному на плате: низкое потребление электроэнергии; компактные габариты и минимальный вес; небольшое время разработки; расширенная функциональность. Типовые компоненты системы на кристалле: процессор (32 бит, технология RISC); периферия (RTC, LCD контроллер, DMA); блок памяти; интерфейсы (USB, LVDS, 10/100base TMAC); аналоговые устройства (DAC, PLL, ADC). Процессоры, используемые в SoC, делятся на две группы по реализации в кремнии: softcore-процессоры [45] – синтезируются и имплементируются на PLD; hardcore-процессоры – логическая структура не меняется во время проектирования и жестко реализована как примитив на FPGA-кристалле (Xilinx Virtex II Pro – имеется 4 IBM PowerPC).

5. Проектирование и верификации SoC

SoC на PLD-кристаллах. При проектировании SoC с последующей реализацией на PLD необходимо разработать модель устройства и описать ее на RTL-уровне. Вторым шагом является формирование тестовых воздействий (Testbench) для проверки функциональности разработанной модели SoC с помощью программ логического моделирования (рис. 18, левая часть) [7]. С возникновением концепции системы на кристалле и методологии ее имплементации в PLD-проекты появляется новый компонент – программное обеспечение, которое как и RTL-код влияет на функционирование системы (рис. 18, правая часть). Возникает вопрос об инструментах верификации, которые обеспечивают моделирование и отладку гетерогенной системы, состоящей из HW-SW компонентов. Традиционные программы логического моделирования не

подходят для этого по ряду причин: отсутствие средств для отладки программного обеспечения [46]; низкая производительность анализа проектов, содержащих большое количество строк программного кода.

Производителями PLD (Xilinx + Altera = 90%) предложена методология раздельной разработки аппаратной и программной частей SoC и соединения их вместе только на этапе создания кристалла (рис. 19). Другими словами, только при программировании PLD-кристалла n-й версией hardware возможна загрузка k-й версии программного обеспечения в PLD прототип в целях совместной верификации путем подключения отладчика программного обеспечения со стандартными средствами (пошаговое выполнение кода, доступ к содержимому регистров процессора, памяти, логических переменных).

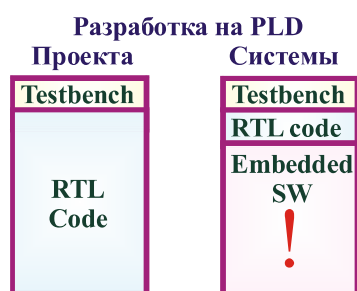


Рис. 18. Разработка проекта и системы на PLD

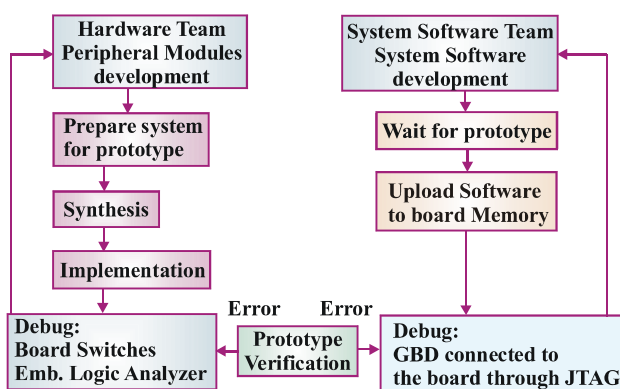


Рис. 19. Раздельная разработка частей SoC

Недостатки методологии: до начала отладки программного обеспечения аппаратная часть должна быть доведена до стадии законченного устройства; при этом ее верификация представляет трудоемкий процесс, связанный с ограниченным доступом к внутренним линиям кристалла аппаратного прототипа и высокими затратами на его создание.

Сегодня на PLD рынке появились softcore процессоры для создания систем на кристалле (PicoBlaze, MicroBlaze от Xilinx и Nios от Altera). Следует отметить гибкость архитектуры периферийной шины Avalon для Nios процессора, который является лидером в этом сегменте рынка. С появлением последней серии Virtex II Pro FPGA, где проектировщикам предоставляются четыре мощных RISC hardware IBM PowerPC процессора на кристалле, началась новая эра развития SoC на PLD.

SoC на ASIC-кристаллах. Интеграция аппаратной и программной части системы после производства кристалла (аппаратной части) — экономически рискованное дело (рис. 20). Могут возникнуть следующие проблемы, которые приводят к задержке выхода проекта на рынок и к большим материальным затратам: 1) программная часть не полностью закончена или отлажена к наступлению этапа интеграции, что требует дополнительного времени на разработку и отладку уже после создания кристалла; 2) отладка аппаратной части внутри большой микросхемы является трудоемкой и времязатратной задачей из-за ограниченного доступа к внутренним линиям чипа; 3) труднолокализуемые аппаратные ошибки приходится исправлять путем больших изменений программной части системы, или делать корректировку HW, что приводит к необходимости повторения цикла проектирования, верификации и изготовления нового кристалла.

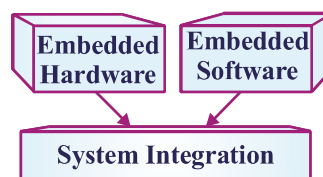


Рис. 20. Интеграция аппаратной и программной частей

Для устранения перечисленных выше рисков была предложена совместная верификация (co-verification) проекта. Она заключается в интеграции программной и аппаратной частей системы еще до создания первого образца чипа с помощью программного пакета Seamless (Mentor Graphics) [47]. Преимущества использования технологии совместной верификации для групп проектировщиков и тестеров путем создания виртуального прототипа (в течение месяца) вместо его аппаратного аналога: 1) ускорение доработки и отладки программной части, дающее временной выигрыш около 3-4 недель; 2) простота и быстрота отладки и локализации ошибок в аппаратной части проекта, которая моделируется в логическом симуляторе; 3) гарантированная 100% вероятность успеха первого силикона, поскольку SW и HW полностью верифицированы к моменту начала производства. Структура системы совместной верификации Seamless (Mentor Graphics) представлена на рис. 21 и состоит из следующих основных компонентов: программа логического моделирования; эмулятор процессора и памяти; модуль связи и синхронизации между собой первых двух компонентов.

Аппаратная часть проекта за исключением процессора и памяти, которые моделируются с помощью программных эмуляторов, отлаживается в HW-Simulator. Программные эмуляторы в 10 - 1000 раз увеличивают скорость моделирования процессора и памяти. Эмулятор процессора позволяет использовать добавочные отладочные инструменты: breakpoints C++/Assembler, выполнять пошаговую отладку кода, иметь доступ в любой момент времени к содержимому регистров процессора и перемен-

ных, которые недоступны при моделировании процессора в HW-simulator. Основная проблема совместной верификации в Seamless — координация работы программы логического моделирования с быстродействием около 10-30 инструкций в секунду и пакета ISS (Instruction Set Simulator) [48], который может обрабатывать в секунду до 500 000 команд. Минусом системы Seamless является ее ориентация только на ASIC рынок и поддержку hardcore процессоров, а softcore Xilinx MicroBlaze процессор — один из лидеров на FPGA рынке — остается в стороне. Пакет Seamless — 60000 долларов, дополненный моделью процессора (6000 — 22000 долларов), ориентирован на ASIC рынок, где стоимость каждого проекта измеряется миллионами долларов [49]. Для разработчиков проектов на FPGA, где большинство пакетов проектирования распространяется производителями кристаллов бесплатно, практически невозможно использовать пакет Seamless из-за высокого ценового барьера.

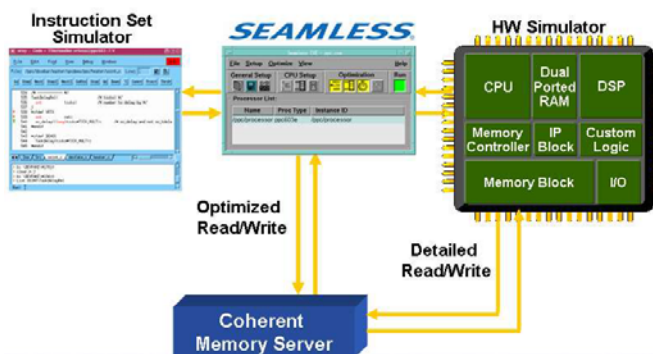


Рис. 21. Структура Seamless-пакета для верификации

Выводы. Сегодня проблема для ученых — как разместить сеть на кристалле (network on a chip), а инженеры уже перешли на проектирование систем на кристалле. Очевиден большой рост PLD-рынка, особенно с приходом в него процессоров (softcore, hardcore), с возможностью конвертации FPGA проекта в ASIC. Следовательно, сегмент рынка, занимаемый ПЛИС, будет только расширяться и отвоевывать место как у интегральных схем, так и у программного обеспечения. На рис. 22 показаны расходы на проектирование и производство по отношению к единице готового изделия в зависимости от выбранной технологии (FPGA, GA, cell-based IC) [6].

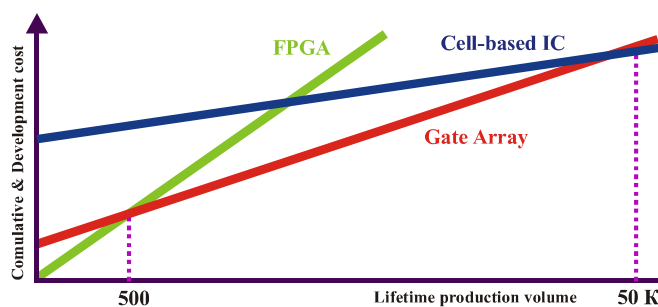


Рис. 22. Расходы по проектированию и производству SoC

До 500 единиц изделия экономически выгодно использовать FPGA-кристаллы. Для крупных партий более 50000 начинает выигрывать cell-based IC технология по цене за единицу изделия при покупке на рынке. Системы на кристалле ранее были лишь привилегией ИС, но с появлением последних серий ПЛИС с процессорами (PowerPc, ARM hardcore, PicoBlaze, MicroBlaze, Nios) SoC стали полноправными участниками рынка ПЛИС.

Рост рынка SoC обусловлен увеличением требований к производительности и надежности цифровых систем, а также к уменьшению их стоимости. Ранее ASIC был практически единственной технологией, позволяющей разработчикам размещать мультипроцессорные системы на одном кристалле. Совершенно новые возможности предоставлены инженерам с появлением последних серий FPGA. Чтобы полностью использовать эти преимущества, необходимы новые методологии и автоматизированные средства, которые позволят повысить производительность разработки. Одной из таких ASIC-методологий, которая становится все более актуальной для FPGA, является аппаратно-программная со-верификация. Также следует заметить, что разработка автоматизированных средств проектирования для использования мощностей, заложенных в последних сериях FPGA, значительно отстает от нанотехнологий. На рынке EDA отсутствуют эффективные средства, позволяющие производить HW-SW совместную верификацию до создания аппаратного прототипа системы в целях исключения ошибок на ранних стадиях проектирования, что приводит к уменьшению общих затрат на отладку проекта.

6. Заключение — практическая значимость представленного обзора

В результате проведенного обзора установлено: ПЛИС предоставляет разработчикам новые возможности для проектирования быстродействующих специализированных цифровых систем на кристаллах. Доказательством перспективности программируемой логики служит появление новых, имеющих более совершенную архитектуру, поколений ПЛИС со встроенными процессорами, возрастание объема их выпуска ведущими зарубежными производителями. Анализ тенденции развития архитектур программируемой логики дает основания утверждать, что в ближайшее время основу элементной базы цифровых (реконфигурируемых) систем будут составлять CPLD, FPGA, HardCopy, догоняющие по производительности и стоимости ASIC, но при этом не требующие высокочрезвычайно затратной тестовой и верификационной методологии для обеспечения массового производства микросхем.

Интерес для ученых и практиков, а также для современного украинского рынка научно-технической продукции представляет и соотношение экономических показателей, создающих целостную картину мирового рынка EDA. На рис. 23 представлены вклады четырех ведущих компаний, формирующих практически весь рынок PLD

(Programmable Logic Device) кристаллов. Что касается цифр, то доля FPGA (Field Programmable Gate Array) составляет 2,8 миллиарда долларов, а CPLD (Complex Programmable Logic Device) – 0,5. При этом Xilinx и Altera формируют 86% от PLD рынка.

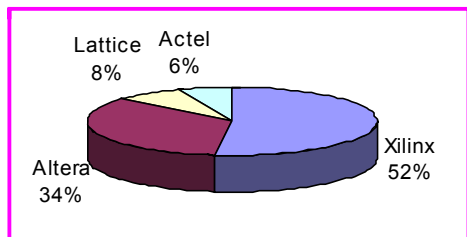


Рис. 23. Формирование рынка PLD

Что касается производителей средств EDA, то их доля на рынке электронных технологий может быть проиллюстрирована следующей диаграммой (четверка лидеров: Cadence, Synopsys, Mentor Graphics, Magma), представленной на рис. 24 (EDA Industry survey. By Dr. Jack Horgan, eda_news_letter@ibsystems.com). Эти фирмы обслуживают 95 % рынка и определяют современную политику технологий проектирования цифровых систем и сетей на кристаллах. Учитывая высокие темпы экономического роста Украины за последние 2 года, упомянутые компании начинают проявлять интерес прежде всего к университетской науке, которая является основным посредником между технологиями проектирования, предлагаемыми фирмами, и нарождающимися пользователями их продукции.

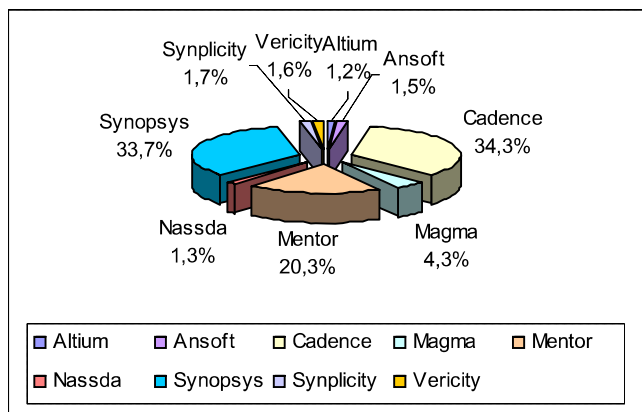


Рис. 24. Рынок производителей EDA

Для потребителей постсоветского и украинского рынка электронных продуктов интерес представляет следующая таблица, которая иллюстрирует преимущества применения FPGA по сравнению с ASIC.

К характеристикам, представленным в таблице, следует добавить факт прозрачности PLD (теоретическое отсутствие программно-аппаратных закладок деструктивного характера) по сравнению с ASIC, что дает возможность использовать FPGA для аппаратной реализации критических технологий (атомная энергетика, космические и авиационные системы, банковские и специальные средства защиты информации).

Выпуск последних серий FPGA Xilinx Virtex II Pro, Virtex 4 и Altera Stratix емкостью более 8 миллионов вентилей со встроенными IBM PowerPC и ARM процессорами обеспечил новый уровень схемотехники, необходимый для проектирования цифровых систем и сетей на FPGA-кристаллах. Аппаратные возможности, предоставляемые нанотехнологиями на 300 мм подложках, требуют перехода на новый уровень развития средств проектирования, где одним из критических и времяемких процессов является функциональная верификация, занимающая до 70% от общего времени создания цифровой системы. При наличии процессора в проектируемой системе главным объектом анализа становится согласование функционирования аппаратных и программных блоков. Для их совместной верификации необходимо иметь специализированные средства и методологии, обеспечивающие эффективную отладку проектируемых систем и сетей на FPGA. Совместная HW/SW верификация успешно используется для отладки проектов на ASIC. Повышенный интерес на рынке EDA к имплементации SoC в FPGA, содержащих процессоры, определяет актуальность задачи разработки аналогичных средств для отладки цифровых систем на ПЛИС.

Characteristic	FPGA	ASIC
Время выхода на рынок	Short	Long
Минимальная партия	None	High
Цена единицы в партии	High	Low
Затраты на подложку	None	High
Реконфигурируемость	High	None
Производительность	Medium	Very High
Плотность вентилей	Medium	Very High
Энергопотребление	High	Low
Сложность разработки	Medium	Very High
Сложность тестирования	Low	High
Длительность цикла разработки	Hours	Months

Исторически сложилась ситуация, когда Европа и Америка владеют технологиями в области электроники, а страны, относящиеся к Восточной Европе, традиционно сильны теоретическими исследованиями. Поэтому научный потенциал стран бывшего СССР становится все более востребованным на рынке электронных технологий в части математических решений и создания программного обеспечения. Известное высказывание К. Маркса “Наука только тогда достигает совершенства, когда ей удастся пользоваться математикой” можно было бы продолжить, например, так: а эффективность математики доказывается путем ее внедрения в практику. В самом деле, история развития радиоэлектронных технологий показывает, что эффект от любой ее практической реализации не менее, чем на 50 % зависит от математического обеспечения (теории, модели, методы, алгоритмы). Практически любое научное исследование связано с математической формализацией рассматриваемых процессов и явлений, превращаемых в общепринятые понятия моделей и методов. В максимальной степени это относится и к дискретной части радиоэлектроники, которая в ближайшие годы покрывает на 90% обозначенную

отрасль. Иначе, цифровое исполнение радиоэлектронных устройств, которые ранее были только аналоговыми (приемники, передатчики, частотные преобразователи, фильтры), приводит к такому соотношению рынка дискретных и аналоговых устройств, когда последние будут выполнять роль интерфейсов в формате АЦП- и ЦАП-изделий. Отсюда следует достаточно интересный вывод: XXI век будет столетием развития телекоммуникационных, но цифровых систем, в которых компьютер будет выполнять роль рассредоточенного в пространстве специализированного компонента, ориентированного не только на выполнение традиционных задач (преобразование, хранение информации), но и на прием и передачу информации в различных форматах, в том числе и с помощью беспроводных средств связи. Достаточно сказать, что современные SoC интегрируют не только АЛУ, память, устройство управления, порты ввода и вывода, но и обязательно приемник с передатчиком, что позволяет строить из таких кирпичиков эффективные макросистемы, рассредоточенные в пространстве. Для построения таких систем необходимо иметь специализированные вычислители, направленные на высокоэффективное решение конкретных проблем путем разработки алгоритмов и их последующей имплементации в универсальную машину или на кремниевый кристалл, выполненный в виде ASIC или PLD. Следует отметить, что эволюция (жизненный цикл) практической реализации математической идеи (законченного решения) проходит несколько стадий: 1) Программная реализация на универсальной машине. Здесь проверяется состоятельность алгоритма, выявление ошибок, его эффективность. 2) Создание IP-core, как интеллектуальной собственности, инвариантной по отношению к ее последующей имплементации в кристалл, описанной на HDL-языке. На данной стадии разрабатываются средства тестирования и верификации программы. 3) Имплементация IP-core в кристалл PLD, как заявка на международное признание эффективности решения конкретной проблемы. Выпуск кристаллов в малых сериях для продажи и дальнейшего изучения спроса. 4) Имплементация IP-core в кристалл ASIC, когда эффективность решения задачи признана на рынке математических идей и электроники, а также определена необходимость тиражирования кристалла в крупных сериях. 5) Аппаратурная реализация решения имплементируется в математический сопроцессор универсальной машины. Это уровень высших математических достижений, которые удостаиваются чести быть доступными всему человечеству.

Другой интересный вывод связан с отношением к Украине со стороны западных стран, которые видят ее не только как поле для инвестиций, но и как партнера и будущего игрока на рынке электроники. Сегодня инвестиционный климат на планете для выполнения работ по технологии Outsourcing весьма благоприятный для Украины. Свидетельством тому могут служить данные о стоимости годовых

услуг квалифицированного программиста в различных регионах мира, где Россия (\$7500) уступает по уровню цен Украине (\$5000). Ведущие европейские и американские фирмы находятся в состоянии ожидания от правительства Украины эффективных мер по привлечению зарубежных частных инвестиций в IT-технологии. Как только будут снижены налоги, украинские и зарубежные фирмы будут сотрудничать напрямую, минуя офшорные зоны. Таким образом, текущий год может стать поворотным для университетской науки и частного бизнеса в плане заключения хоздоговоров на разработку программных и аппаратных продуктов с европейскими, но прежде всего с германскими фирмами. Это может дать существенный импульс развитию научных университетских школ, которые вынуждены будут делать не то, что могут, а то, что необходимо и действительно актуально сейчас для мирового рынка. С другой стороны, для выполнения работ по технологии Outsourcing мы должны более интенсивно осваивать не только современные средства проектирования цифровых систем, но и менеджмент для своевременного и качественного выполнения заказов.

В последние годы доминирующей является тенденция к микроминиатюризации исполнения цифровых проектов, благодаря внедрению микро- и нанотехнологий: Board заменяется на Chip, более того, в кристалл имплементируется и сеть вычислительных устройств, с обменом информации по IP-протоколу. Вместе с тем микроминиатюризация дает возможность незаметно встраивать цифровые системы в традиционные изделия повседневного спроса. Так появляются wearing (embedded) systems – системы, встроенные в одежду и обувь, которые решают проблему энергопотребления с помощью теплоотдачи тела человека. Таким образом, микроминиатюризация дает новый импульс к развитию встроенных специализированных цифровых систем, которые в 50 % случаев своей функциональностью могут заменить дорогостоящие универсальные компьютеры, выигрывая к тому же в размерах, мобильности и энергопотреблении.

Важное методологическое значение для выбора актуальных направлений исследования имеют следующие законы развития рынка электроники [50]: закон Мура (Moore's Law) – “Число транзисторов в микросхемах удваивается каждые 18 месяцев”; закон Рока (Rock's Law) – “Производительность оборудования для выпуска полупроводниковых кристаллов удваивается каждые 4 года”; закон Макроуна (Machrone's Law) – “Компьютер, который вы хотите купить, всегда стоит \$5000 (1000 – для Украины)”; закон Меткафа (Metcalfe's Law) – “Рост значимости сети определяется квадратом числа её пользователей”; закон Вирта (Wirth's Law) – “Снижение скорости программного обеспечения происходит быстрее роста скорости аппаратуры”; закон Якоба Нильсена (Nilsen's Law) – доступная для пользователей пропускная способность сети возрастает на 50% ежегодно, но разработчики Web-сайтов не получают этого преимущества, которое

компенсируется увеличением числа клиентов и замедлением программного обеспечения. Законы, выведенные Е. Зорианом [51] (Y. Zorian): 1) Количество транзисторов на кристалле возрастает быстрее, чем их необходимый объем для проектирования сложных цифровых систем. 2) Инвестиции в технологический процесс преобладают над капиталовложениями для создания новых цифровых и микропроцессорных систем. 3) Материальные затраты на разработку программного обеспечения встроенной системы (embedded system) составляют 80% от ее стоимости. 4) Затраты на тестирование имеют экспоненциальную зависимость от объема (стоимости) выпуска продукции. 5) Число инженеров-верификаторов (тестеров) в 2 раза превышает количество проектировщиков цифровых и микропроцессорных систем.

Каждый тезис имеет методологическое значение для перспектив развития радиоэлектронного производства и рынка развивающихся стран, поскольку все компании мира обязательно учитывают обозначенные тенденции для планирования своей инвестиционной деятельности.

Литература: 1. *Benoit Clement, Richard Hersemeule, Etienne Lantreibecq, Bernard Ramanadin, Pierre Coulomb, Francois Pogodalla*. Prototyping: a System Design Flow Applied to a Complex System-On-Chip Multiprocessor Design. Proceedings of the Design Automation Conference. 1999. P. 420-424. 2. *Nollet V., Marescaux T., Verkest D., Mignolet J.-Y., Vernalde S.* Operating-System Controlled Network on Chip. Proceedings of the Design Automation Conference. 2004. P. 256 - 259. 3. *Fournier L., Arbetman Y., and Levinger M.* Functional verification methodology for microprocessors using the Genesys test-program generator. Proceedings of the 1999 Design, Automation and Test in Europe Conference. 1999. P. 434-441. 4. *Fine S. and Ziv A.* Coverage directed test generation for functional verification using Bayesian networks. Proceedings of the 40th Design Automation Conference. 2003. P. 286-291. 5. *Synplicity Inc.* White Paper. Architecture and Automation: Optimized Design Flow for NEC Electronics' ISSPs. 2003. Synplicity. 7 p. 6. *NEC* White paper. Gate Array Solution. Document No. A16841EJ2V0PF00 (2nd edition). March 2004 N CP(K). NEC Electronic Corporation 2004. 9 p. 7. *Karen Parnell, Nick Mehta.* Programmable Logic Design Quick Start Handbook. August 2003. 210 p. 8. *Serge Leaf.* A Methodology for Virtual Hardware/Software Integration. Mentor Graphics Corporation white paper. 2003. 6 p. 9. *Passerone C., Lavagno L., Chiodo M., Sangiovanni-Vincentelli A.* Fast hardware/software co-simulation for virtual prototyping and trade-off analysis. Proc. Design Automation Conference. 1997. P. 389-394. 10. *EPSON* Electronic Devices white paper. ASIC Gate Array/Standard Cell Embedded Array. 2004. 39 p. 11. *Benoit Clement, Richard Hersemeule, Etienne Lantreibecq, Bernard Ramanadin, Pierre Coulomb, Francois Pogodalla.* Fast Prototyping: a System Design Flow Applied to a Complex System-On-Chip Multiprocessor Design. Proc. Design Automation Conference. 1999. P. 420-424. 12. *Bulent I. Dervisoglu.* A Unified DFT Architecture for Use with IEEE 1149.1 and VSIA/IEEE P1500 Compliant Test Access Controllers. Proceedings of the Design Automation Conference. 2001. P. 53-58. 13. *Agrawal V.D., Kime C.R., Saluja K.K.* A Tutorial on Built-In Self-Test, Part 1: Principles. IEEE Design & Test. 1993. Vol. 10, No. 1. P. 73-82. 14. *Bardell P.H., McAnney W.H., Savir J.* Built-

In Test for VLSI: Pseudorandom Techniques, John Wiley & Sons. 1987. 232 p. 15. *Wohl P., Waicukauski J. A., Patel S., Amin M. B.* (Synopsys, Incorporated). Efficient Compression and Application of Deterministic Patterns in a Logic BIST Architecture. Proceedings of the Design Automation Conference. 2003. P. 566-569. 16. *Negreiros M., Carro L., Susin A. A.* (Universidade Federal do Rio Grande do Sul). Ultimate Low Cost Analog BIST. Proceedings of the Design Automation Conference. 2003. P. 570-74. 17. *Chowdhury S. and Breuer M. A.* Optimum Design of IC Power/Ground Nets Subject to Reliability Constraints, IEEE Trans. of Computer-Aided Design. Vol. 7, No. 7. July 1988. P. 787-796. 18. *Stark D. and Horowitz M.* Techniques for Calculating Currents and Voltages in VLSI Power Supply Networks. IEEE Tran. of Computer-Aided Design. Vol. 9, No. 2, February 1999. P. 126-132. 19. *Wu S.-W.* (Elan Microelectronics Corporation), *Chang Y.-W.* (National Taiwan University). Efficient Power/Ground Network Analysis for Power Integrity-Driven Design Methodology Proceedings of the Design Automation Conference. 2004. P. 77-180. 20. *Wojciech Maly.* IC Design in High-Cost Nanometer-Technologies Era. Proceedings of the Design Automation Conference. 2001. P. 9-14. 21. *Fine S. and Ziv A.* Coverage Directed Test Generation for Functional Verification using Bayesian Networks. Proc. Design Automation Conference. 2003. P. 286-291. 22. *Lee D., Zolotov V., Blaauw D.* Static Timing Analysis using Backward Signal Propagation. Proc. Design Automation Conference. 2004. P. 664-669. 23. *Geng Bai, Sudhakar Bobba, Ibrahim N. Hajj.* Static Timing Analysis Including Power Supply Noise Effect on Propagation Delay in VLSI Circuits. Proceedings of the Design Automation Conference. 2001. P. 295-300. 24. *Dongkun Shin, Jihong Kim, Seongsoo Lee.* Low-Energy Intra-Task Voltage Scheduling Using Static Timing Analysis. Proceedings of the Design Automation Conference. 2001. P. 438-443. 25. *Schubert T.* (Intel Corporation). High Level Formal Verification of Next-Generation Microprocessors. Proceedings of the Design Automation Conference. 2003. P. 1-6. 26. *Vittal A., Lia H., Brewer F., Marek-Sadowska M.* Clock skew optimization for ground bounce control. ICCAD Digest of Technical Papers. 1996. P. 395-399. 27. *Mustafa Badaroglu, Kris Tiri, Stephane Donnay, Piet Wambaca, Ingrid Verbauwhede, Georges Gielen, Hugo De Man.* Clock Tree Optimization in Synchronous CMOS Digital Circuits for Substrate Noise Reduction Using Folding of Supply Current Transients. Proceedings of the Design Automation Conference. 2002. P. 399-404. 28. *Dunlop A. E. and Kernighan B. W.* A procedure for placement of standard-cell VLSI circuits. IEEE Trans. CAD. Jan. 1985. P. 92-98. 29. *Ke Zhong, Shantanu Dutt.* Algorithms for Simultaneous Satisfaction of Multiple Constraints and Objective Optimization in a Placement Flow with Application to Congestion Control. Proceedings of the Design Automation Conference. 2002. P. 854-859. 30. *Morton P., Dai W.* Crosstalk. Crosstalk Noise Estimation for use in an Efficient Noise Management Strategy. Technical Report UCSC-CRL-01-09, UCSC. 2001. 7 p. 31. *Giorgio Casinovi.* An Algorithm for Frequency-Domain Noise Analysis in Nonlinear Systems. Proceedings of the Design Automation Conference. 2002. P. 514-517. 32. *Abramovici M., Breuer M.A. and Friedman A.D.* Digital systems testing and testable design. Computer Science Press. 1998. 652 p. 33. *Mo F., Brayton R. K.* (University of California at Berkeley). A Timing-Driven Module-Based Chip Design Flow. Proc. Design Automation Conference. 2004. P. 67-70. 34. *Lipsett R.* VHDL: Hardware Description and Design. Kluwer Academic Publishers. 1989. 488 p. 35. *IEEE* Standard Verilog Hardware Description Language. The Institute of Electrical and Electronics Engineers. New York. 2001. 791

p. 36. *Bergeron Janick*. Writing testbenches: functional verification of HDL models.— Boston: Kluwer Academic Publishers. 2001. 354 c. 37. *Henfiling R., Zinn A., Bauer M., Zambaldi M., Ecker W.* (Infineon Technologies AG.) Re-Use-Centric Architecture for a Fully Accelerated Testbench Environmen. Proceedings of the Design Automation Conference. 2003. P. 372-37. 38. *Ur S., Yadin Y.* Micro-Architecture Coverage Directed Generation of Test Programs. Proceedings of the Design Automation Conference. 1999. P. 175-180. 39. *Asaf S., Marcus E., Ziv A.* Defining Coverage Views to Improve Functional Coverage Analysis. Proc. Design Automation Conference. 2004. P.41-44. 40. *Schubert K.-D.*(IBM Corporation). Improvements in Functional Simulation Addressing Challenges in Large, Distributed Industry Projects. Proceedings of the Design Automation Conference. 2003. P.11-14. 41. *Jagesh Sanghavi, Albert Wang.* Estimation of Speed, Area, and Power of Parameterizable, Soft IP. Proceedings of the Design Automation Conference. 2001. P. 31-34. 42. *Reyneri L. M., Cucinotta F., Serra A., Lavagno L.* A Hardware/Software Co-design Flow and IP Library Based of Simulink TM. Proceedings of the Design Automation Conference. 2001. P. 593-598. 43. *Epson* white paper. FPGA to ASIC Conversion. Copyright. 2003. *Epson Electronics.* 22 p. 44. *Rashinkar P., Paterson P. and Singh L.* System-On-Chip Verification Methodology and Techniques. Kluwer Academic Publishers. 2001. 432 p. 45. *Rowen C., Leibson S.* (Tensilica, Inc.). Flexible Architectures for Engineering Successful SOCs. Proceedings of the Design Automation Conference. 2004. P. 692-697. 46. *Marick B.* The craft of Software Testing, Subsystem Testing Including Object-Based and Object-Oriented Testing. Prentice Hall. 1985. 420 p. 47. *Milan Saini,* Technical Marketing Manager, Xilinx, Inc. White paper. Seamless FPGA Speeds Verification for Xilinx Virtex-4 and Virtex-II Pro FPGAs with Embedded IBM PowerPC Processors. 5 p. 48. *Harris D., Stokes D., Klein R.* Executing an RTOS on Simulated Hardware using Co-

verification. Proceedings of Embedded Systems Conference. 2000. P. 344-349. 49. *Dipert B.* Silicon Segmentation. Design Feature Magazine. September 2003. P. 57-66. 50. *Philip E. Ross.* Five Commandments. IEEE Spectrum. December 2003. P. 30-35. 51. *Don Edenfeld, Andrew B Kahng, Mike Rodgers and Yervant Zorian.* 2003 Technology Roadmap for Semiconductors. Computer, January 2004. P. 47-56.

Поступила в редколлегию 11.11.2004

Рецензент: д-р техн. наук, проф. Кривуля Г.Ф.

Хаханов Владимир Иванович, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: техническая диагностика вычислительных устройств, систем, сетей и программных продуктов. Увлечения: баскетбол, футбол, горные лыжи. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua

Егоров Александр Андреевич, аспирант кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и верификация цифровых систем на кристаллах. Увлечения: баскетбол, футбол, автопутешествия. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua

Каминская Марина Александровна, аспирантка кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и верификация цифровых систем на кристаллах. Увлечения: иностранные языки. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua

Хаханова Ирина Витальевна, канд. техн. наук, доцент кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и диагностика цифровых устройств. Увлечения: английский язык, гитара. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua