

# UniICS – универсальный внутрисхемный симулятор

**Михаил Долинский,  
Алексей Федорцов,  
Вячеслав Литвинов,  
Юрий Кадетов**

## Введение

По экспертным оценкам, верификация занимает от 40 до 80% стоимостных и временных ресурсов современных проектов. Особые трудности вызывает поиск и диагностирование ошибок на заключительном этапе — при работе с изготовленным прототипом. В качестве предлагаемых сегодня решений этих проблем выступают универсальные испытательные стенды, а также встраиваемые средства отладки. Данный материал посвящен разработкам, выполненным в СНИЛ «Новые информационные технологии» (<http://NewIT.gsu.unibel.by>) Гомельского государственного университета им. Ф. Скорины (Белоруссия).

## 1. Проблемы верификации «на последней миле»

Предположим, разработка выполняется уже несколько месяцев, и вышла на финальную стадию — изготовление прототипа. По синтезированному описанию построен конфигурационный файл, произведена его загрузка в ПЛИС, начато тестирование в реальном режиме. Однако выясняется, что устройство работает не так, как ожидалось. Как и с помощью каких средств выяснить, в чем причина?

Другой вариант: все большее число разработчиков приходит к пониманию того факта, что в целях упрощения и сокращения сроков разработки необходимо обеспечивать сквозное интегрированное проектирование, обеспечивая взаимодействие уже разработанных на физическом уровне компонентов с другими, для которых есть еще только высокоуровневая (на языке программирования) или HDL (Hardware Description Language) модель. То есть необходимо обеспечить симуляцию одной части проекта и эмуляцию другой части, да еще их взаимодействие с адекватными возможностями обмена сигналами. Какими средствами это можно сделать?

Третий вариант: система становится настолько сложной, что требуется разработка специальных средств, обеспечивающих съём (установку) информации с (на) определенных линий схемы.

Сходные проблемы, отягощенные гораздо более высокой ценой исправления ошибок, возникают при разработке устройств на базе СБИС или печатных плат, интегрирующих множество микросхем.

Кроме того, уже при производстве отлаженных изделий возникает потребность в контроле качества изготовления, а в случае многокомпонентных

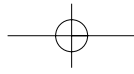
систем и в диагностике, позволяющей точно идентифицировать неисправные компоненты.

Для решения всех этих проблем в СНИЛ «Новые информационные технологии» разработаны универсальный внутрисхемный эмулятор UniICE и универсальный внутрисхемный симулятор UniICS [1–4]. UniICE ориентирован на специальную работу с микропроцессорами. UniICS обеспечивает общую работу с произвольными схемами. Данный материал посвящен именно UniICS. Другому прибору будет посвящена отдельная статья.

## 2. Предлагаемая идеология решения описанных проблем

Основным инструментом разработчиков цифровых систем в настоящее время является персональный компьютер (ПК) или выполняющая функции последнего, но с большей вычислительной мощностью, рабочая станция. На ПК выполняются программы ввода, редактирования, симуляции и синтеза описаний цифровых систем. И потому, естественно, предпочтительно водрузить на этом же рабочем месте программно-аппаратный комплекс, который позволит решать вышеозначенные проблемы «последней мили» при разработке цифровых систем. Очевидно, для этого нужно соответствующее программное обеспечение, в качестве которого используются описанные ранее [5–12] системы HLCCAD/IEESD-2000. Они позволяют визуально вводить, редактировать и симулировать аппаратное обеспечение. Для отлаженного проекта, представляющего иерархическую композицию из элементов библиотеки стандартных компонентов, поставляемых с системами HLCCAD/IEESD, может быть автоматически сгенерировано синтезируемое VHDL-описание. Системы являются открытыми, и потому при разработке проекта можно несколькими способами добавить новые синтезируемые компоненты. Кроме того, в качестве компонентов проекта могут использоваться высокоуровневые модели (на языках программирования типа Object Pascal и C++) для обеспечения верификации проектируемой цифровой системы, состоящей из компонентов различного уровня готовности.

В систему HLCCAD/IEESD интегрирован программно-аппаратный комплекс UniICS, первоначальная цель которого заключалась в обеспечении корректной трансляции сигналов между HLCCAD/IEESD и имеющейся аппаратной реализацией (всего проекта цифровой системы или его части).



В случае, если аппаратная реализация включает весь проект, разработчик может использовать для его тестирования систему HLCCAD/IEESD как среду верификации, повторно используя при этом все разработанные ранее тесты. Более того, множество тестов может пополняться и модифицироваться, при этом разработчик находится в привычной ему среде разработки. А все вновь созданные тесты можно будет использовать и при чисто программной симуляции проекта.

Заметим, что использование связки HLC-CAD/IEESD и UniICS эффективно, даже если сам проект разрабатывался и изготовлен без их использования, поскольку HLCCAD/IEESD обеспечивает широкий спектр средств генерации входных воздействий и анализа реакций устройства, а UniICS обеспечивает адекватную трансляцию сигналов от HLCCAD/IEESD к прототипируемой системе и обратно.

Но особенно эффективно постоянное применение HLCCAD/IEESD и UniICS на всем протяжении разработки цифровой системы. Дело в том, что аппаратная часть UniICS была реализована на ПЛИС для сокращения временного цикла исправления ошибок и упрощения развития системы. Более того, сама схема UniICS разрабатывалась посредством HLCCAD/IEESD. Когда выяснилось, что на ПЛИС, используемой для реализации функций UniICS, остается достаточно много свободных ячеек (а саму ПЛИС принципиально не сложно менять на более емкую), возникла идея, а потом и реализация использования свободного места на ПЛИС для отладки проектируемого устройства или его части. В результате теперь на любом этапе проектирования разработчик для любого компонента проекта может указать флаг «эмулировать», в результате чего перед началом симуляции всей системы этот компонент загружается в ПЛИС, содержащую UniICS, и UniICS автоматически настраивается на трансляцию сигналов между той частью проекта, которая симулируется в HLC-CAD/IEESD, и той частью проекта, которая загружена в ПЛИС.

Это позволяет начинать решение проблем «последней мили» задолго до того, как это делается при обычном проектировании. Подчеркнем, что при такой работе HLC-CAD/IEESD обеспечивает привычную разработчику и весьма удобную визуализацию данных о сигналах с «погруженных в железо» компонентов: навигация по времени, отображение значений непосредственно на схеме, фильтры сигналов, таблицы изменений значений, временные диаграммы и т. д. Можно использовать язык тестовых воздействий и регрессионное тестирование для автоматизации корректности различных компонентов проектируемой цифровой системы, по очереди или вместе «погружаемых в железо». После завершения отладки компонентов их можно продолжать «загружать в железо» для обеспечения «аппаратной акселерации».

Наличие вышеописанной возможности позволяет также использовать готовые IP-компоненты сторонних производителей. Для этого на схеме создается корпус с соответствующи-

ми контактами, который связывается с исходным текстом IP-компонента. При запуске на компиляцию IP-компонент синтезируется и загружается в ПЛИС.

### 2.1. Конфигурируемая эмуляционная платформа

Конфигурируемая эмуляционная платформа (КЭП) включает физический и логические компоненты. Физическая составляющая КЭП конструктивно представляет собой печатную плату с размещенными на ней микросхемами программируемой логики и загрузчика. На плате имеются также разъем для интерфейса с ПК (используется также для загрузки) и разъем интерфейса с внешней средой функционирования.

Загрузчик — это стандартная схема, используемая для конфигурации ПЛИС данными, которые подаются с инструментального ПК.

Логические компоненты КЭП включают управляющее программное обеспечение на ПК, конфигурируемую синтезируемую RTL-модель внутрисхемного симулятора. Управляющее программное обеспечение обеспечивает конфигурацию необходимой RTL-модели симулятора, загрузку ее в ПЛИС на КЭП, а также взаимодействие пользователя с загруженным в ПЛИС симулятором. Модель внутрисхемного симулятора обеспечивает двустороннюю трансляцию значений сигналов между внешней средой, подключенной к контактам ПЛИС, управляемым внутрисхемным симулятором и верифицируемой системой (часть ее может находиться на ПК и симулироваться, тогда как другая часть может находиться в ПЛИС и эмулироваться).

Симулятор имеет следующие параметры конфигурации:

- количество контактов с внешней средой;
- установка типа каждого контакта (входной, выходной или двунаправленный);
- установка периодичности обновления и опроса информации на контактах симулирующей системой;
- управление оперативной памятью для автономного буферирования значений контактов с последующей блочной передачей информации симулирующей системе.

### 2.2. Базовая эмуляция синтезируемых подсхем

Проект в системе IEESD-2000 представляет собой визуализируемую иерархическую систему компонентов следующих типов:

- Схема или составной компонент — является объединением других схем или монолитных (синтезируемая, DLL или VHDL) компонентов.
- Синтезируемый компонент — имеет симуляционную модель (написанную на языке программирования, допускающем создание DLL — Dynamic Link Library, которая компилируется независимо от компиляции симулирующей системы) и программу генерации синтезируемого VHDL-описания данного компонента. Система IEESD-2000 имеет готовую библиотеку таких компонентов от логических элементов до ОЗУ. Важно заметить, что все эти компоненты параметризованы

по количеству и типам контактов (прямой, инверсный, фронт и др.). Реконфигурация этих компонентов может осуществляться непосредственно на схеме в любой момент проектирования. Автоматически синтезируется VHDL-модель, соответствующая данному способу конфигурации компонента. Для схемы, состоящей из синтезируемых компонентов, автоматически генерируется идентичное синтезируемое VHDL-описание схемы. Технология создания синтезируемых компонентов является открытой и хорошо документированной, что позволяет пользователям разрабатывать собственные специализированные библиотеки синтезируемых компонентов (например фильтры, фреймы БПФ и т. д.).

- DLL-модель — имеет симуляционную модель, но не имеет синтезируемой VHDL-модели. Например, модель внешней среды, или «золотая модель», функционально сложной части проекта.
- VHDL-модель — имеет только синтезируемое VHDL-описание, полученное, например, от стороннего источника (разработчика IP-компонентов).

Каждый структурный компонент (составной или монолитный) имеет флаг «эмулировать», который устанавливается пользователем.

В случае установки такого флага соответствующий составной или монолитный компонент с помощью Max+Plus II переносится в ПЛИС конфигурируемой эмуляционной платформы.

Установка флага «эмулировать» автоматически запрещается для DLL-моделей, а также для схем, содержащих DLL-модели на любом из уровней вложенности.

Таким образом, процесс чистой симуляции проекта конвергирует от симуляции к эмуляции. Часть схемы симулируется, а перенесенная в КЭП часть схемы эмулируется. При этом HLCCAD/IEESD-2000 обеспечивает адекватную трансляцию сигналов между симулируемой и эмулируемой частями проекта.

### 2.3. Эмуляция взаимодействия с внешней средой

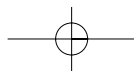
КЭП имеет набор контактов для взаимодействия с внешней средой и частью реально функционирующей схемы. Кроме того, HLCCAD/IEESD обеспечивает управление конфигурированием этих контактов (ввод, вывод, двунаправленный) и может присоединить к ним любой из контактов проекта. Во время верификации HLCCAD/IEESD-2000 обеспечит соответствующую трансляцию информации между внешней средой и схемой.

## 3. Описание реализации UniICS

### 3.1. Состав, конструкция и подключение

Универсальный внутрисхемный симулятор UniICS представляет собой аппаратно-программную систему для проектирования и отладки цифровых устройств произвольной сложности (в том числе и многопроцессорных) в реальной среде функционирования.

В качестве программного обеспечения для симулятора используется среда моделирова-



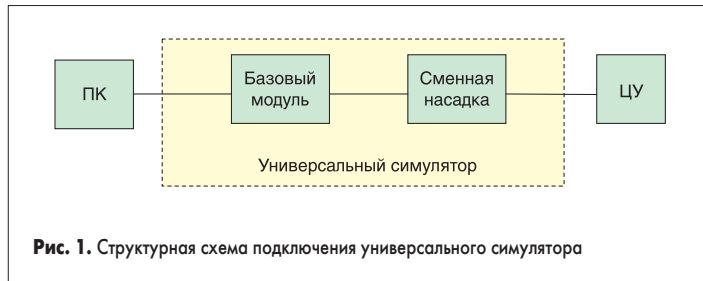


Рис. 1. Структурная схема подключения универсального симулятора

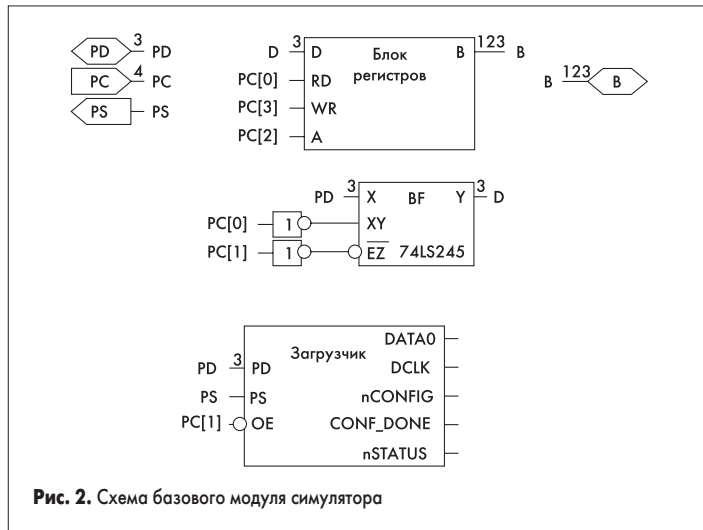


Рис. 2. Схема базового модуля симулятора

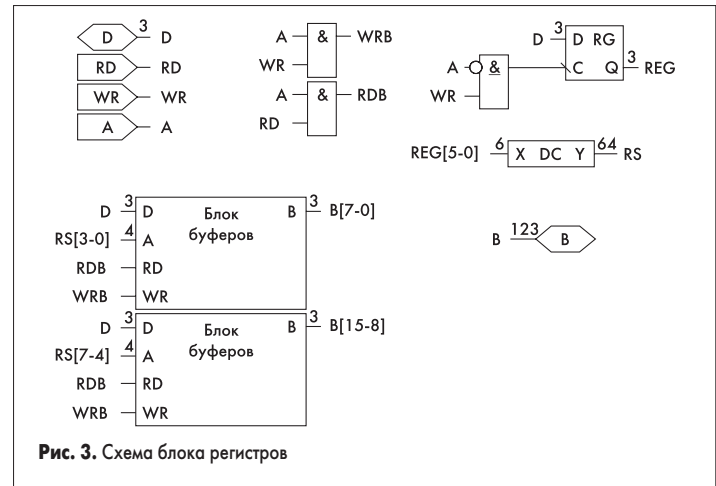


Рис. 3. Схема блока регистров

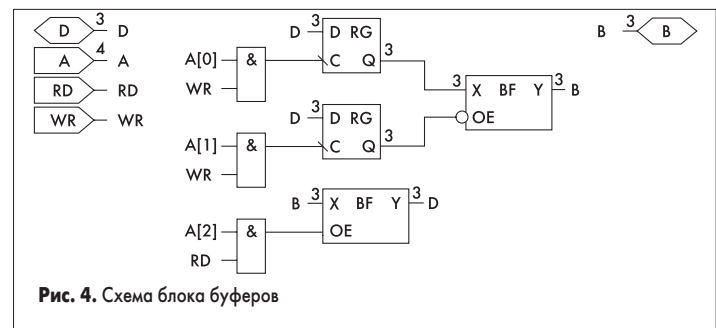


Рис. 4. Схема блока буферов

ния HLCCAD/IEESD, которая обладает соответствующими функциональными возможностями, в том числе удобными средствами ввода и редактирования схем, анализа результатов моделирования.

Программно-аппаратный комплекс на основе универсального симулятора состоит из следующих составных частей: персонального компьютера (ПК), базового модуля универсального симулятора и сменной насадки. ПК обеспечивает интерфейс комплекса с пользователем и выполнение рабочей программы, управляющей симулятором. В базовом модуле сосредоточены основные узлы симулятора. Конструктивно он представляет собой печатную плату с установленными на ней интегральными схемами и разъемами. Один из разъемов предназначен для соединения с управляющим ПК, другой — для соединения со сменной насадкой. Соединение с ПК осуществляется при помощи параллельного интерфейса. Этот интерфейс должен поддерживать режим двунаправленной передачи данных.

Сменная насадка обеспечивает электрическое и логическое согласование симулятора с целевым устройством. Применение сменной насадки обусловлено необходимостью обеспечивать симуляцию того или иного устройства в различных вариантах исполнения корпусов. Непосредственный контакт с целевым устройством обеспечивается вилкой, количество и расположение выводов которой соответствует количеству и расположению выводов корпуса симулируемого устройства. Вилка вставляется в розетку на целевой плате. Иногда необходимо подключаться к разрозненным контрольным точкам в целевом устройстве. В этом случае сменная насадка представляет собой набор проводов с клеммами на концах.

Структурная схема подключения универсального симулятора приведена на рис. 1. В состав базового модуля входят следующие конструктивные элементы: разъем интерфейса с ПК, ПЛИС, загрузчик и разъем интерфейса со сменной насадкой. Внутри ПЛИС реализована вся логическая структура симулятора. Загрузчик предназначен для программирования ПЛИС в момент включения. При этом данные для загрузки поступают из ПК. Также он может использоваться как самостоятельное устройство и выполнять функции интерфейса JTAG. Это может найти применение при симуляции тех устройств, доступ к внутренним цепям которых затруднен.

### 3.2. Структурно-функциональная схема универсального симулятора и его функционирование

Симулятор представляет собой иерархическую структуру следующего вида:

- Симулятор:
  - Базовый модуль
  - Загрузчик
  - Блок регистров
  - Блок буферов 1
  - .....
  - Блок буферов N

Рассмотрим схему универсального симулятора и ее функционирование в соответствии с приведенной иерархией. На рис. 2 представлена схема базового модуля симулятора. В ее состав входят блок регистров, загрузчик и согласующий буфер. Блок регистров реализован в ПЛИС.

Перечислим назначение каждого из основных структурных элементов, входящих в состав базового модуля симулятора:

- Блок регистров (рис. 3) содержит набор буферных блоков и систему адресации регистров. Для адресации используются адрес-

ный регистр и дешифратор. Адресация организована таким образом, что на каждый блок буферов приходится 4 адреса. Поскольку разрядность адресного регистра равна 8, общее число адресуемых блоков составляет 64.

- Блок буферов (рис. 4) является нижней иерархической единицей симулятора. Реализован блок буферов, который обеспечивает работу с 8 линиями ввода-вывода, каждая из которых может независимо конфигурироваться на ввод или вывод информации. Для этого используются входной и выходной буферы. Информационные и управляющие сигналы формируются на выходах регистров и управляют входными данными и разрешением Z-состояния выходного буфера соответственно.

### 4. Библиотека подпрограмм для независимой работы с универсальным внутрисхемным симулятором UniICS

Эта информация может быть использована для интеграции UniICS со сторонним программным обеспечением.

Обмен данными с аппаратной частью проекта осуществляется посредством вызова нескольких функций.

```
function InitSimulator(path: string) : boolean;
```

Функция инициализации симулятора. В качестве параметра для этой функции передается путь к файлу, содержащему прошивку текущего проекта в бинарном виде (генерируется программой MaxPlus+II).

```
procedure SetHardLine(LineID: integer; Value: char);
```





### 5.3. Удаленная работа с UniICS в среде HLCCAD/IEESD-2000

В системах HLCCAD и IEESD-2000 реализована возможность для работы с UniICS удаленно, используя подключение по локальной сети или через Интернет. При этом клиентская часть запускается на машине пользователя. Обмен данными происходит с сервером, к которому подключен UniICS. Это позволяет работать нескольким пользователям с одним прибором. Для предотвращения возможных коллизий реализован механизм очереди.

Порядок работы с удаленным UniICS в целом схож с описанным в предыдущем разделе порядком работы с локальным прибором. При генерации файла прошивки нужно выбрать скрипт «NewIT UniICS compile (remote)» (рис. 6), при этом дополнительно происходит вызов параметра «Remote Synthesis», который производит подключение к удаленному серверу синтеза, передает полученный VHDL-файл и получает файл прошивки (параметры удаленного сервера находятся в файле «Projects\UX\UX.ini»).

Если сервер синтеза занят, то запрос будет поставлен в очередь и будет получено сообщение «Server busy (N). Waiting...», где N — это порядковый номер в этой очереди.

Кроме того, следует указать «rUniICS» при выборе названия модели (рис. 8). Это позволит подключиться к удаленному серверу модели. Если сервер модели занят, то запрос будет поставлен в очередь и будет получено сообщение «Server busy (N). Waiting...», где N — это порядковый номер в этой очереди.

Удаленная работа с UniICS использовалась для проведения студенческих лабораторных и курсовых работ в Гомельском государственном университете им. Ф. Скорины.

### 5.4. Использование UniICS в качестве встроенного средства отладки

Хотелось бы остановиться еще на одной возможности применения UniICS — в качестве IP-компонента, встраиваемого в пользовательский проект. Это позволило бы получить возможность доступа к внутренним компонентам проекта, что значительно облегчило бы отладку проекта при разработке, а также обеспечило бы удобство тестирования и диагностики готового прибора при его эксплуатации. При этом интерфейсная часть UniICS могла бы подстраиваться под используемые в проекте протоколы обмена. Поскольку затраты на внесение в пользовательский проект как аппаратной, так и программной частей UniICS минимальны, такое решение представляется весьма эффективным и перспективным.

### 5.5. Отладка проекта системы на кристалле на базе связи IEESD-2000+UniICS

Хотелось бы обратить внимание на весьма важную возможность применения внутри-схемного симулятора в связке с IEESD-2000 — в качестве системы отладки систем на кристалле (процессора и логики, расположенных на одном кристалле). При этом микропроцессорное ядро будет симулироваться средой IEESD, а UniICS будет исполнять роль массива программируемой логики, а также набора

всех остальных необходимых для работы компонентов.

Проект в IEESD-2000 на верхнем уровне будет содержать процессорное ядро (в нашем примере i8051) и систему логики, которая затем будет погружена в UniICS (рис. 9).



Рис. 9. Верхний уровень проекта системы на кристалле в среде IEESD-2000

На корпусе FPGA можно создать контакты для подключения к соответствующим контактам процессора. Все необходимые для полнофункциональной работы системы компоненты (генераторы, источники прерываний, таймеры и т. п.) могут быть реализованы использованием возможностей среды IEESD-2000. При необходимости они могут быть также реализованы внутри FPGA и соединены с процессором соответствующими контактами. В нашем примере, как показано на рисунке, с каждым контактом процессора соединен соответствующий контакт на FPGA. Таким образом, верхняя часть проекта останется неизменной при дальнейшем его использовании.

Внутри себя FPGA может содержать любые компоненты, для которых может быть синтезировано VHDL-описание. Кроме стандартных, необходимых для системы модулей, внутри FPGA будут содержаться схемы, «погружаемые» в массив программируемой логики разрабатываемой SoC.

Как показано на рис. 10, внутри схема FPGA содержит два устройства. Первое — System Control — содержит устройства управления системой на кристалле (таймеры, генераторы и т. п.). Выходы этого устройства соединены с соответствующими выходами схемы FPGA, которые, в свою очередь, соединены с управляющими входами процессора. Второе устройство — System Logic — будет содержать систему логики. Его входы через соответствующие входы схемы FPGA соединены с информационными контактами процессора.

При необходимости, между этими двумя устройствами могут быть созданы дополнительные служебные контакты. Они могут по-

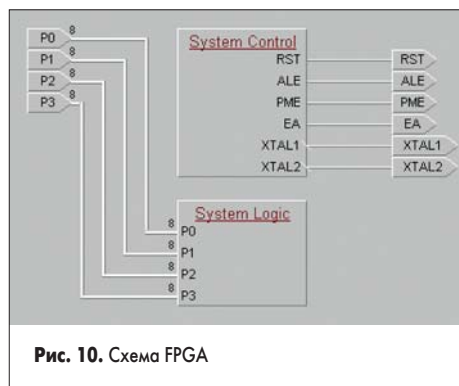


Рис. 10. Схема FPGA

надобиться для управления работой проекта из системы логики.

Для запуска проекта необходимо создать файл прошивки для UniICS, запустив соответствующий скрипт IEESD-2000, после чего схема FPGA помечается флагом для погружения в симулятор (этот процесс был описан выше). Работа с проектом будет вестись в основном внутри устройства System Logic, где будет описываться система логики. Схема устройства System Control меняться не будет, изменения в этой части проекта будут затрагивать только настройку соответствующих компонентов (частота генератора, режимы работы таймеров и т. п.). После настройки и разработки схемы System Logic необходимо снова выполнить скрипт генерации файла прошивки UniICS и запустить проект на моделирование.

В настоящее время необходимость использования систем на кристалле при создании сложных цифровых устройств очевидна. С другой стороны, приобретение полнофункциональной системы на кристалле во многих случаях может быть проблематичным. Кроме того, на этапе отладки может оказаться, что приобретенная SoC не удовлетворяет всем потребностям разработчика. В этом случае, при использовании вышеописанного подхода не возникает потребности в приобретении нового устройства, поскольку все необходимые доработки можно будет сделать в уже имеющейся системе. В частности, возможности среды IEESD-2000 позволяют подключать к проекту собственные процессоры, то есть для каждой конкретной задачи может быть разработан и подключен специфический процессор.

При отладке систем на кристалле наиболее актуальна симуляция системы логики. В описанном выше подходе программно симулировать будет только процессор, а схема самого цифрового устройства будет работать в UniICS. Обмен данными процессора с системой логики и другими компонентами SoC будет обеспечиваться механизмами UniICS, которые могут быть рассмотрены как своеобразная «системная шина». Кроме того, при использовании SoC часто возникают проблемы наблюдения за работой системы. При использовании же вышеописанного проекта появляются все возможности отладки устройства, предоставляемые IEESD-2000, — возможности задания точек останова и пошаговой трассировки проекта, просмотр содержимого регистров, дампов памяти, таблиц и диаграмм истории изменения значения на контактах схемы, «ручное» изменение значений на контактах «на лету» и многое другое. Также разработчик имеет возможность отладки устройства «во времени», то есть задать интервал времени, через который будет происходить обращение к UniICS, что позволяет избавиться от зависимости от тактовой частоты во время отладки проекта. Для этого нужно вызвать локальное меню над корпусом FPGA, выбрать пункт «Задержка» и установить необходимый интервал. Кроме того, в любое время можно вернуть систему в состояние, в котором она

находилась в предыдущие моменты времени работы. При этом моделирование останавливается и, если сохранялась трасса, можно при помощи ползунка вернуть систему в выбранный момент времени.

Таким образом, при использовании связки IEESD-2000+UniICS и вышеописанного проекта для отладки цифровых устройств, мы, с одной стороны, избавляемся от зависимости от конкретной отладочной платформы при проектировании устройства, с другой стороны — получаем полнофункциональную, легко конфигурируемую, «гибкую» и простую в управлении систему на кристалле, в которой разработанные устройства работают в реальной среде функционирования (в «железе»).

#### Заключение

Использование UniICS совместно с системами HLCCAD/IEESD-2000 может в значительной степени сократить сроки и стоимость разработки цифровой системы. Синтезируемое описание UniICS может быть использовано в качестве основы при разработке специализированных средств отладки для сложных систем на кристалле. ■

#### Литература

1. Dolinsky M. S., Ziselman I. M., Fedortsov A. O. In-circuit emulators of microprocessors and microcontrollers // Automatic Control and Computer Sciences. New York. Allerton Press. 1999. Vol. 33. No 1.
2. Долинский М. С., Зисельман И. М., Федорцов А. О. Внутрисхемные эмуляторы микропроцессоров и микроконтроллеров // Автоматика и вычислительная техника. Рига. 1999. No 1.
3. Внутрисхемный эмулятор / А. О. Федорцов, М. С. Долинский (Беларусь). No a19980557. // Изобретения. Полезные модели. Промышленные образцы. 1999. № 4.
4. Федорцов А., Долинский М. Устройство для отладки микропроцессорных систем // Электроника. Минск. 1999. No 1.
5. M. Dolinsky. High-level design of embedded hardware-software systems // Advances in Engineering Software. UK. Oxford. «ELSEVIER». 2000. Vol. 31. No 3.
6. Dolinsky M. S. Integrated Environment IEESD-2000 for embedded system development // Automatic Control and Computer Sciences. New York. Allerton Press. 1999. Vol. 33. No 3.
7. Долинский М. С. Интегрированная среда для разработки встроенных систем IEESD-2000 // Автоматика и вычислительная техника. Рига. 1999. No 3.
8. Долинский М. Концептуальные основы и компонентный состав IEESD-2000 — интегрированной среды сквозной совместной разработки аппаратного и программного обеспечения встроенных цифровых систем // Компоненты и Технологии. 2002. № 8.
9. Долинский М., Литвинов В., Галатин А., Ермолаев И. HLCCAD — среда редактирования, симуляции и отладки аппаратного обеспечения // Компоненты и Технологии. 2003. № 1.
10. Долинский М., Литвинов В., Толкачев А., Корнейчук А. Система высокоуровневого проектирования аппаратного обеспечения HLCCAD: тестирование // Компоненты и Технологии. 2003. № 3.
11. Долинский М., Литвинов В., Галатин А., Шалаханова Н. Система высокоуровневого проектирования аппаратного обеспечения HLCCAD: открытый универсальный интерфейс моделируемых компонентов // Компоненты и Технологии. 2003. № 4.
12. <http://NewIT.gsu.unibel.by>.