

Российская САПР Delta Design Simtera

Н. Малышев¹

УДК 621.3 | ВАК 05.13.12

Сегодня отечественный рынок программируемых логических интегральных схем (ПЛИС) может предложить аналоги западных микросхем и даже наметилась тенденция своих, уникальных решений в этой области. Примером тому служит ПЛИС МЗ компании АО «ПКК Миландр». Что же касается систем разработки конфигурации ПЛИС и архитектуры больших и сверхбольших интегральных схем (СБИС), здесь все значительно сложнее. Российским разработчикам приходится ориентироваться на западные решения ввиду отсутствия отечественных. Нужно признать, что некоторые действия в этом направлении предпринимались, но они, к сожалению, оказались недостаточными. Это связано с тем, что разрабатывались лишь отдельные части сквозного цикла проектирования, и при этом все равно оставалась зависимость от западных решений. Но так было до недавнего времени, пока компания «ЭРЕМЕКС» не выпустила программный пакет сквозного проектирования конфигурации ПЛИС и СБИС Delta Design Simtera.

На сегодняшний день разработка проектов для конфигурации ПЛИС является востребованной тематикой. За счет скорости работы, возможности многократной реконфигурации и простоты разработки ПЛИС находят применение в различных сферах электроники: телекоммуникациях, космонавтике, а также в других продуктах военного и гражданского применения, и даже в добыче (майнинге) криптовалют.

Успехи в области интегральной микроэлектроники, с одной стороны, дают возможность реализовывать на одном «кристалле» архитектуры проектов, состоящих из десятков и сотен миллионов вентиляей, но с другой – требуют использования более «продвинутых» автоматизированных систем проектирования, так как стандартные решения либо невозможны, либо требуют неоправданно большего количества времени, что замедляет выход продукта на рынок. На помощь разработчикам интегральных схем пришли языки описания аппаратуры (HDL), способные описывать абстрактные модели логических схем, а также «кремниевые компиляторы» – программные пакеты, способные перевести (синтезировать) абстрактные модели в список соединений, состоящих из логических вентиляей.

В статье проведем обзор текущих возможностей и решений, которые дизайн-центры могут применять уже сейчас, используя Delta Design Simtera. Кроме того,

расскажем о разрабатываемых модулях, которые могут дополнительно ускорить процесс перехода с западных аналогичных систем разработки на отечественные за минимально короткое время.

ИСТОРИЯ ПРОЕКТА

Delta Design Simtera в самом начале пути своего развития позиционировался как модуль цифрового моделирования в составе программного пакета Delta Design. Этот пакет воплощает в себе множество отдельных модулей для разработки печатных плат, начиная с ведения библиотек и заканчивая подготовкой данных для производства. Развитие программного пакета Delta Design Simtera началось еще в начале 2010-х годов (самого Delta Design и топологического роутера TopoR – еще десятилетием ранее), до первых санкций, и в то время он представлял собой отдельный программный пакет по моделированию кода на языке VHDL. С тех пор позиционирование продукта изменилось, ввиду спроса и внешних обстоятельств, в сторону расширения функциональности. Теперь использовать Delta Design Simtera можно как отдельно, если мы говорим о проектировании конфигурации и архитектуры микросхем, так и в составе Delta Design – здесь модуль можно применять для моделирования цифровых частей и блоков в составе проектируемого радиоэлектронного изделия.

Забегая вперед, хочется сказать, что санкции и ограничения пошли только на пользу развитию отечественной системы – предприятия начали задумываться

¹ Компания «ЭРЕМЕКС», ведущий разработчик, malyshev.n@eremex.ru.

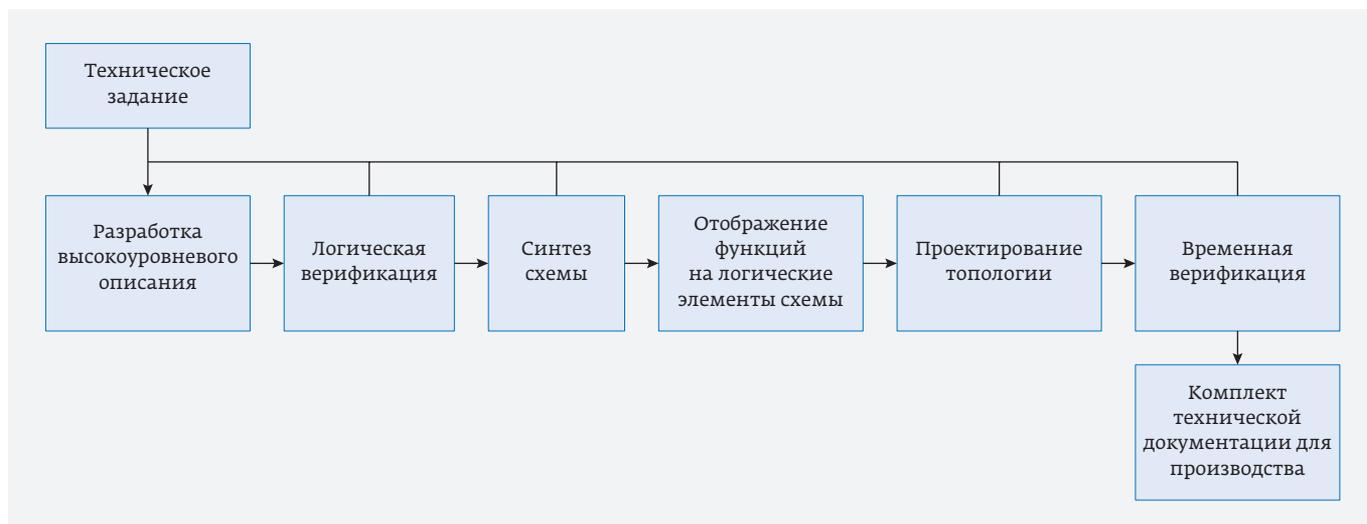


Рис. 1. Маршрут разработки интегральной схемы (упрощенный)

о возможном отключении их от западных программных комплексов, встал вопрос о наличии «закладок» в разрабатываемых микросхемах ввиду закрытости автоматизированных систем. Также разрабатывались собственные уникальные решения в микроэлектронике. Текущие события показывают, что развитие собственных чипов, в том числе широкого спроса, необходимо, а вместе с ним необходимым становится и развитие отечественных систем проектирования. Нужно отдать должное: в последние годы государство активно помогает и мотивирует отрасль развивать конкурентоспособные решения, действуя не путем ограничения использования зарубежных продуктов, а путем поддержки российских предприятий, интересуясь при этом эффективностью, в том числе коммерческой.

МАРШРУТ ПРОЕКТИРОВАНИЯ

Итерационный процесс разработки интегральных схем можно изобразить графиком (рис. 1). Некоторые из шагов на графике маршрута упростим, например процесс упаковки, расстановки и трассировки назовем «проектирование топологии». Каждый из этапов предполагает множество подэтапов, не будем их рассматривать, дабы сократить план и отобразить основные шаги. В статье разберем шаги разработки высокоуровневого описания, логической верификации, поведенческого и логического синтеза и сопутствующие им инструменты разработки на каждом из этапов проектирования.

РАЗРАБОТКА ВЫСОКОУРОВНЕВОГО ОПИСАНИЯ

Вести разработку архитектуры и конфигурации интегральных схем можно как в схемотехническом, так и в «текстовом» виде, то есть путем написания исходного кода, описывающего функциональность проектируемой микросхемы. При разработке схемотехнического описания используются инструменты ведения библиотек, создания компонентов – их условных графических обозначений (УГО), создания модели высокоуровневого описания и генерации его шаблона на основе УГО (рис. 2, 3).

Уже на основе базы компонентов можно перейти к схемотехническому проектированию и далее в системе Delta Design Simtera предусмотрен цикл разработки вплоть до создания файла конфигурации. На рис. 4 представлен проект последовательного периферийного интерфейса

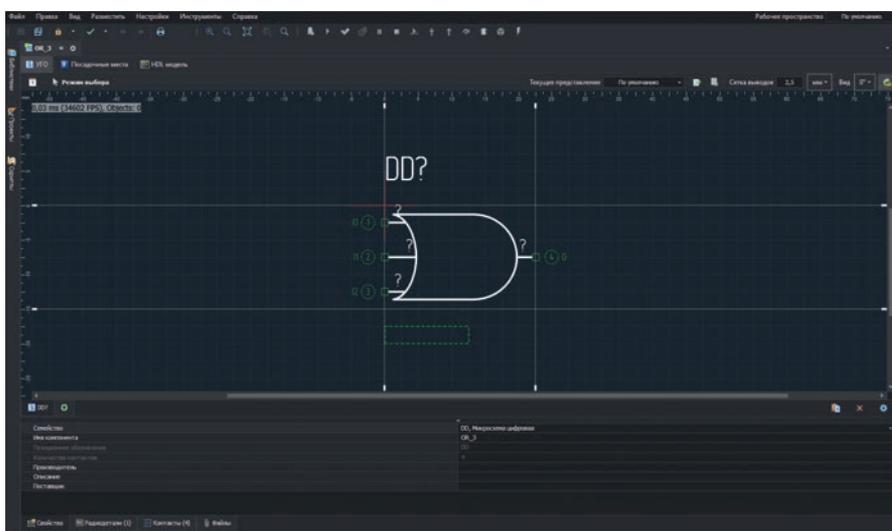


Рис. 2. Создание условного графического обозначения компонента

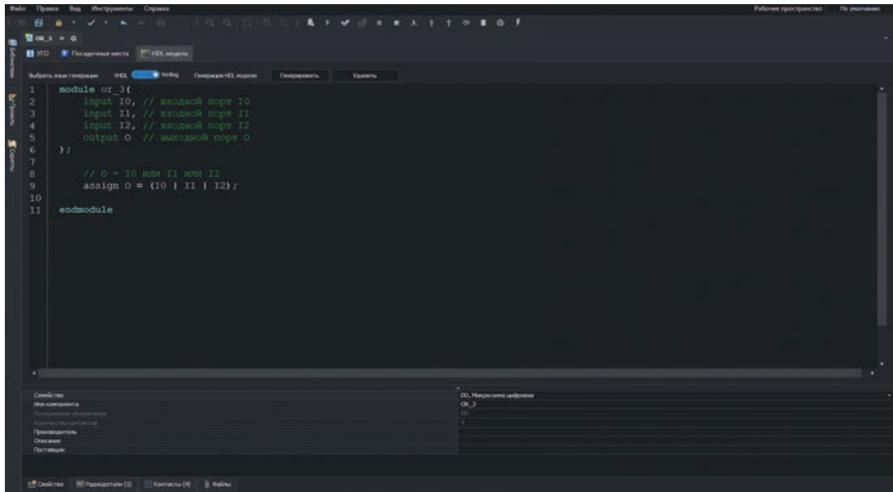


Рис. 3. Создание поведенческой модели на языке описания аппаратуры на основе условного графического обозначения

(SPI) с ведущей и ведомой частями, соединенными с тестовым блоком (тестбенчем).

При работе с проектом, представленным в виде исходного кода на HDL-языках, важным является процесс взаимодействия системы и разработчика. Современный разработчик уже привык к интерактивным системам, дающим ему информацию об ошибках, «узких местах» разрабатываемого кода, а также к тому, что система показывает их и дает свои рекомендации.

За счет интерактивных инструментов разработки процесс создания HDL-кода становится быстрее и, как правило, лишен грубых ошибок. В системе Delta Design Simtera реализован следующий инструментарий в помощь программистам ПЛИС и СБИС: визуальные средства и «инструменты обратной связи». К визуальным средствам можно отнести редактор настройки подсветки синтаксиса, отвечающий за настройку стилей: цвета и размера шрифта и других параметров ключевых слов для HDL-языков; настройка цветовых схем редакторов также важна и присутствует в системе. В качестве «инструментов обратной связи» выступают средства автоподсказок, автодополнений кода, распознавания лексических, синтаксических и смысловых (семантических) ошибок кода. Технология умного ведения кода проекта предлагает варианты для написания кода, уменьшает время на его создание и сокращает возможность совершения синтаксических ошибок. В качестве иллюстрации работа системы приведена рис. 5 и 6.

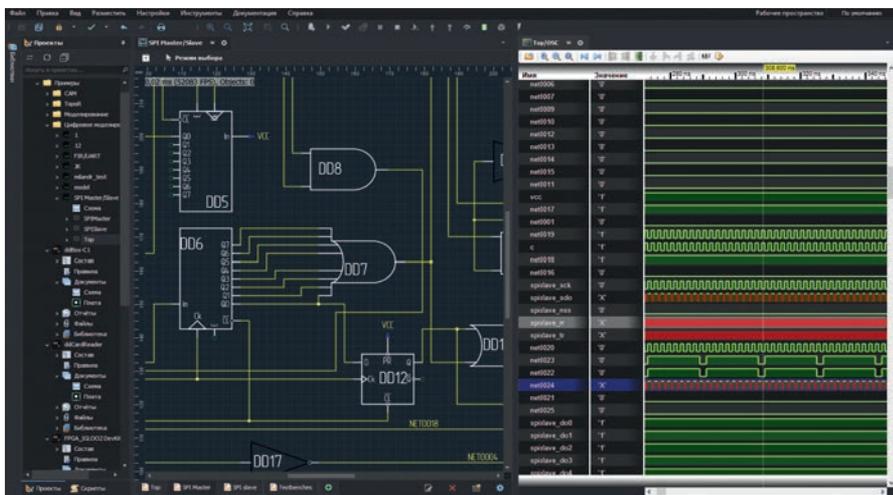


Рис. 4. Схематехнический проект последовательного периферийного интерфейса с результатами моделирования

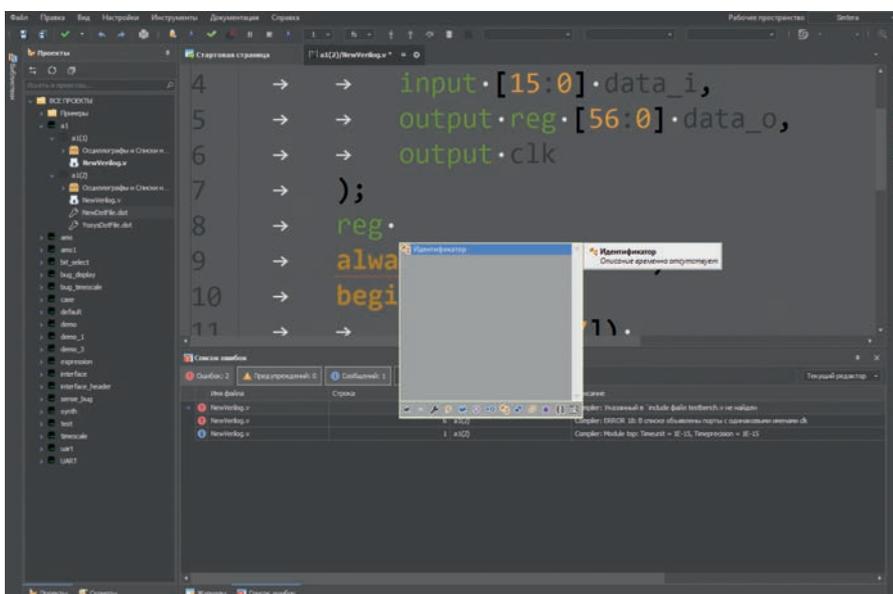


Рис. 5. Автодополнение Verilog-кода в зависимости от контекста

В зависимости от контекста система предлагает варианты дальнейшего описания по стандартам для языков: IEEE 1800-2005 (SystemVerilog), IEEE 1364-2001 (Verilog), IEEE 1076-2008 (VHDL).

МОДЕЛИРОВАНИЕ И ВЕРИФИКАЦИЯ HDL

Моделирование осуществляется на основе скомпилированного HDL-проекта или схмотехнического проекта, компоненты которого также собираются (компилируются). При сборке проекта система проектирования проверяет, были ли сделаны ошибки. В случае их отсутствия строится внутренняя модель, по которой как раз и проводится функциональная проверка работы проекта. Существует несколько вариантов проверки: поведенческое моделирование (построение временных диаграмм и их отображение в осциллографе), пошаговая отладка с размещением точек установки в тестируемых частях проекта, верификация с помощью утверждений (assertion-based verification) и т.д. Все эти инструменты можно использовать в Delta Design Simtera в полной мере.

Отрисовка поведенческого моделирования во временной области доступна в осциллографе и в списке наблюдения. Данные могут быть отображены в цифровом и аналоговом видах, во временной области доступна расстановка курсоров, в нижней части курсора отображается дельта времени между текущим и «нулевым» курсорами, в верхней части – абсолютное значение времени (рис. 7). Верификация и поведенческое моделирование HDL-проекта предполагает полную поддержку VHDL, поддержку Verilog и SystemVerilog. Также при верификации можно промоделировать аппаратные средства совместно со встраиваемым программным обеспечением – модель устройства может быть промоделирована совместно с программной прошивкой этого же устройства. При этом обеспечивается единый стек – единый показ значений сигналов и переменных программы прошивки. Контроль изменения сигналов происходит параллельно

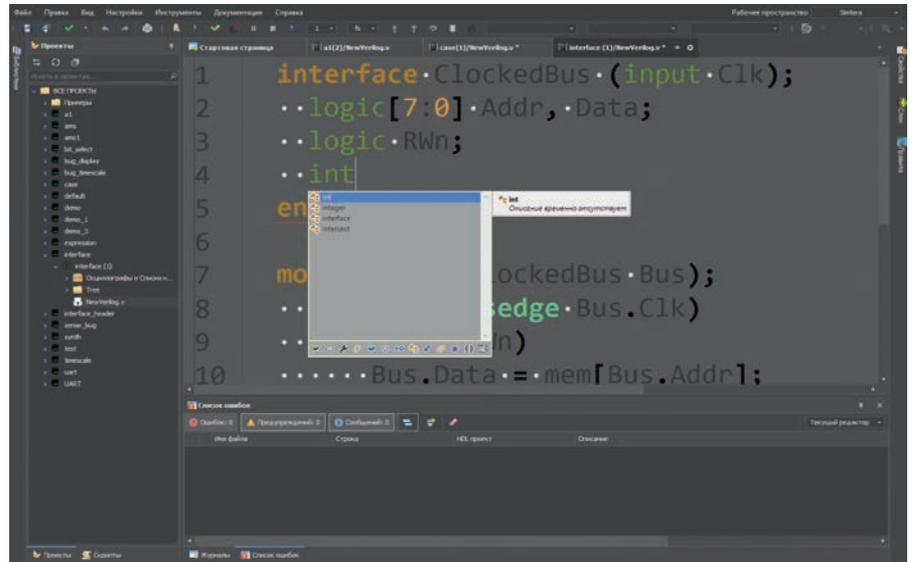


Рис. 6. Дополнение кода в зависимости от введенных символов

с отладкой модели и осуществляется в отдельном окне виртуального осциллографа. Во время отладки доступен просмотр как стека вызовов, так и значений локальных переменных.

ПОВЕДЕНЧЕСКИЙ СИНТЕЗ. ПРОВЕРКА НА СИНТЕЗИРУЕМОСТЬ

HDL-языки, и Verilog в частности, используются для описания поведения схемы во времени и запуска соответствующего моделирования. Однако воплощение реализующей такое поведение цифровой аппаратуры, то есть логических элементов и соединяющих их проводников,

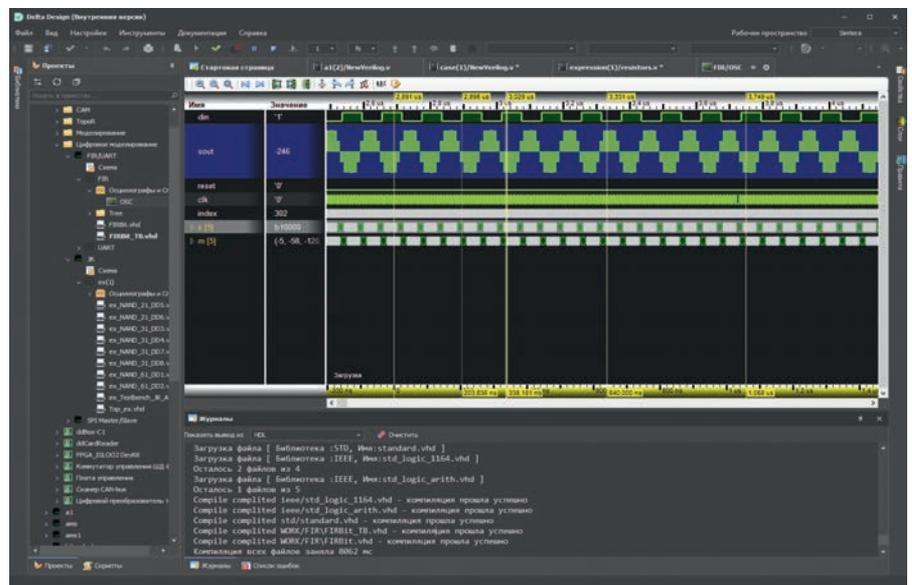


Рис. 7. Осциллограф Delta Design Simtera

возможно не всегда. Логический синтез – перевод поведенческой модели в набор цифровых компонентов – можно выполнить при использовании синтезируемого подмножества языка (Synthesizable Verilog).

Вполне корректный с точки зрения моделирования код может содержать несинтезируемые конструкции, и сообщение об этом должно быть выдано пользователю только при запуске модуля синтеза, а не при общей проверке кода на отсутствие синтаксических или семантических ошибок. При этом обнаружение несинтезируемых конструкций следует выполнять раньше, чем будет запущен трудоемкий процесс логического синтеза. Это реализуется путем добавления к проверке семантики дополнительных правил, следование которым проверяется сразу же при запуске синтеза, и в случае их нарушения выдает ошибку, а процесс синтеза не запускается, что сокращает время на поиск несинтезируемых конструкций.

Рассмотрим Verilog-код, содержащий несинтезируемую конструкцию event:

```
module dff(clk,r_data,q);
input clk;
input r_data;
output reg q;
event e_data;

always@(posedge(clk))
begin
if(clk)->e_data;
end

always @(e_data)
q = r_data;

endmodule
```

В Delta Design Simtera компиляция проходит без ошибок, но при проверке синтезируемости кода выдается ошибка (рис. 8).

Задача поведенческого синтеза – это перевод абстрактной языковой модели в представление на уровне регистровых передач. Задача логического синтеза – оптимизация результатов уровня поведенческого синтеза и реализация схемы списка соединений уже с использованием компонентов выбранной ПЛИС или разработанных библиотек проектируемой СБИС. На текущий момент в системе Delta Design

...	Имя файла	Строка	HDL проект	Описание
▶	file1.v	5	test5	Synthesizer Error 6903: Синтез событий (event) не поддерживается

Рис. 8. Предупреждение системы проверки на синтезируемость

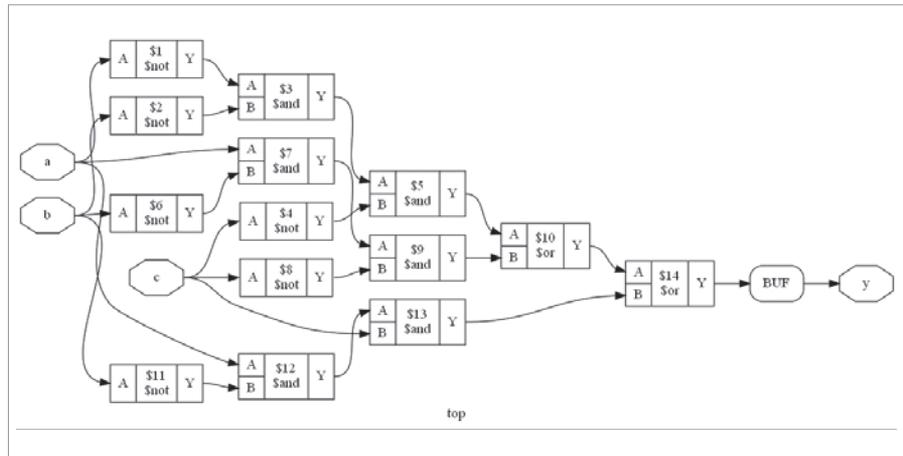


Рис. 9. Результаты поведенческого синтеза в Yosys

Simtera присутствуют инструменты по поведенческому синтезу. Продемонстрируем его работу и сделаем сравнительный анализ с решением с открытым исходным кодом Yosys и коммерческим решением от компании Synopsys – Synplify.

Рассмотрим простые логические операции над булевыми переменными:

```
module top(input logic a, b, c, output logic y);
assign y = (~a & ~b & ~c)
| (a & ~b & ~c)
| (a & ~b & c);
endmodule
```

На результате работы Yosys (рис. 9) все операции показаны в графической форме. На рис. 9 можно увидеть, что все три блока 2, 6 и 11 представляют собой отрицание одной и той же переменной b, 4 и 8 – отрицание c, а 7 и 12 каждый соответствуют (a & ~b).

В Delta Design Simtera на настоящий момент реализована процедура повторного использования уже созданных

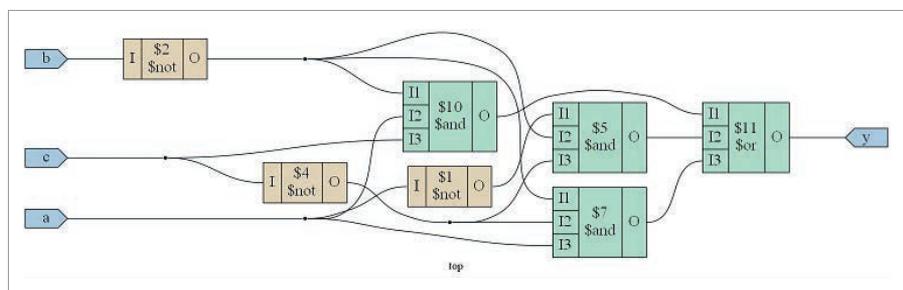


Рис. 10. Результат поведенческого синтеза в Delta Design Simtera

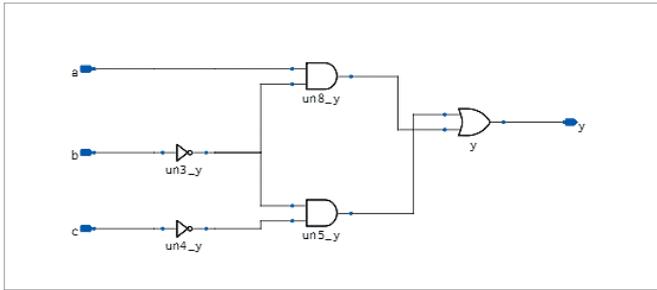


Рис. 11. Результат поведенческого синтеза в Synplify

блоков, что приводит лишь к однократному появлению отрицаний из переменных. Кроме того, в блоках (рис. 10) И и ИЛИ на схеме изображаются три операнда, как и записано в коде.

Никакой минимизации здесь не проводится: лишь повторное использование уже созданных блоков. Для примера покажем работу Synplify, который путем минимизации записанной функции снизил число операторов до 5 (рис. 11).

ЗАЩЕЛКИ И МУЛЬТИПЛЕКСОРЫ

Комбинационная логика, описываемая условными операторами, реализуется с помощью мультиплексоров, осуществляющих выбор одного из множества сигналов в зависимости от контрольного сигнала, и защелок – пропускающих или нет входящий сигнал в зависимости от контрольного сигнала. Рассмотрим следующий пример:

```
module latch_range2(input logic clk,
                   input logic [3:0] d0,
                   input logic [1:0] d1,
                   output logic [3:0] q);
  always_latch
  if (clk) q[1:0] <= d1;
  else q <= d0;
endmodule
```

Yosys выдает в качестве результата ссылку на некую процедуру, которую он не может представить в виде цифровых компонентов (рис. 12), а Delta Design Simtera реализует описанную выше логику (рис. 13), аналогичный результат выдает Synplify (рис. 14).

РАЗРАБОТКА КОНФИГУРАЦИИ ПОД ОТЕЧЕСТВЕННЫЕ ПЛИС

В предыдущих пунктах мы ознакомились с шагами проектирования, верификации и синтеза, но их недостаточно для подготовки конфигурации для программируемой логической интегральной схемы. В Delta Design Simtera реализованы инструменты подготовки файлов ограничений, упаковки, размещения, трассировки и конфигурации для ПЛИС МЗ компании «Миландр». Тем самым программный комплект решает все необходимые для разработчиков задачи

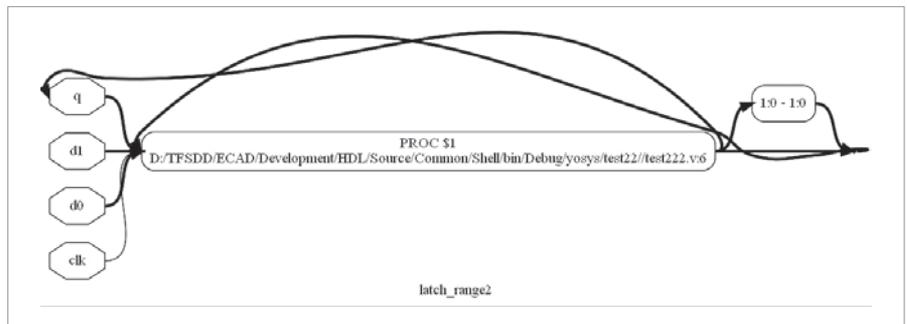


Рис. 12. Поведенческий синтез в Yosys при использовании мультиплексоров и защелок

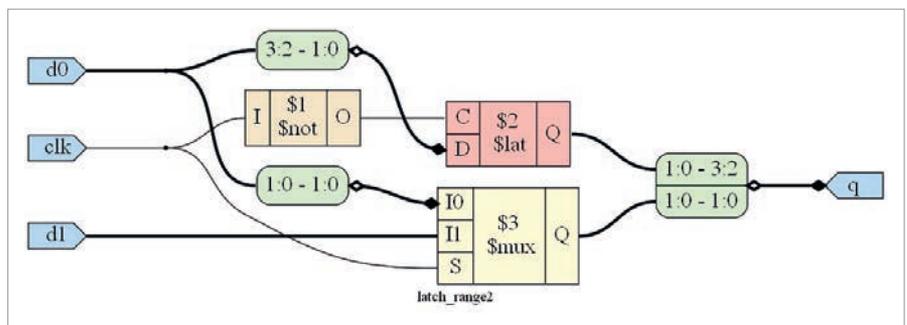


Рис. 13. Поведенческий синтез в Delta Design Simtera при использовании мультиплексоров и защелок

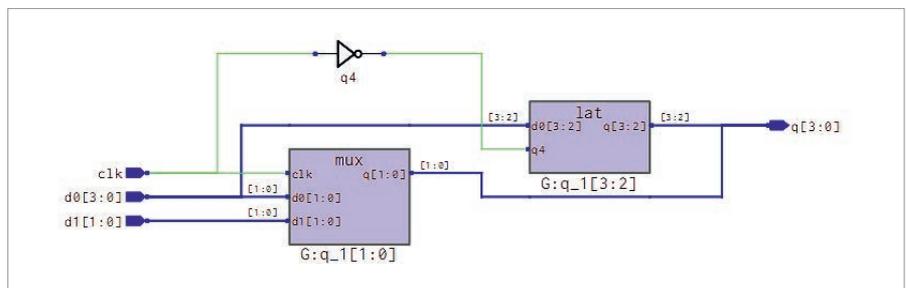


Рис. 14. Поведенческий синтез в Synplify при использовании мультиплексоров и защелок

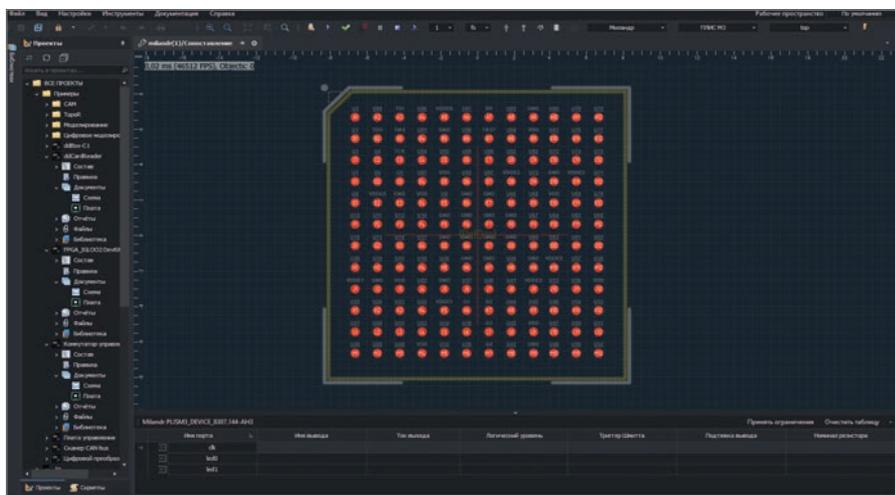


Рис. 15. Графический редактор для подготовки файлов ограничений Delta Design Simtera

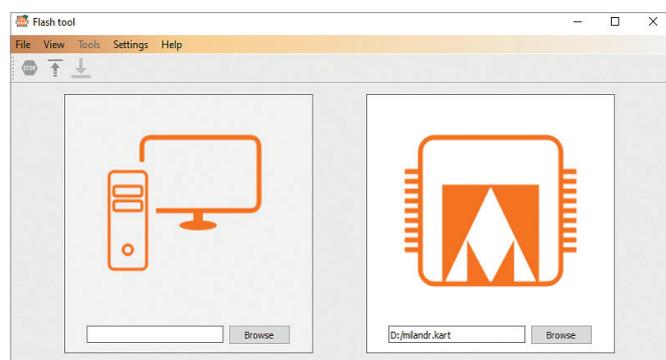


Рис. 16. Программа прошивки конфигурационной программы в чип

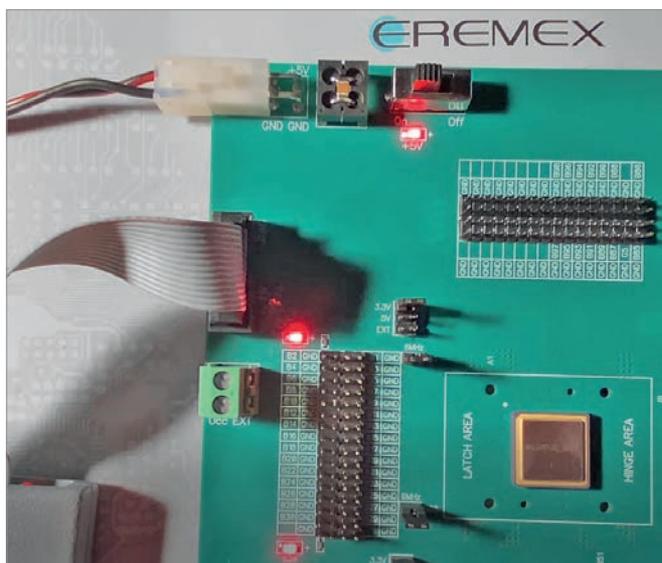


Рис. 17. Работа сконфигурированной отладочной платы

по проектированию конфигурации, без необходимости использовать дополнительные сторонние инструменты. На рис. 15, 16 и 17 представлены снимки инструментов по подготовке файлов ограничения, программы прошивки («прожига») кристалла и работы отладочной платы.

ГОТОВЯЩИЕСЯ К ВЫПУСКУ ИНСТРУМЕНТЫ РАЗРАБОТКИ

С учетом того факта, что разработчики ПЛИС и СБИС используют широкие инструменты по автоматизации, в том числе скрипты TCL, применяют серверные решения для сокращения скорости верификации и синтеза, в разработке сейчас на-

ходятся конвертеры TCL для взаимодействия со средой Delta Design Simtera и ее автоматизации. Также на этапе подготовки находится кроссплатформенное использование среды, в том числе и на отечественных операционных системах и процессорах «Эльбрус». Данная функциональность разрабатывается в том числе благодаря стимулирующим государственным программам развития отечественной микроэлектроники.

На сегодняшний день, несмотря на санкционное давление, а во многом даже благодаря ему, развиваются отечественные системы автоматизированного проектирования. Развитие, особенно САПР микроэлектроники, во многом сопровождается кооперацией производителей микросхем и разработчиков ПО, так как это единственно правильный вектор развития и выпуска конкурентоспособной продукции. Так, например, за последние несколько лет за счет спроса, разработки собственных решений, а также за счет сотрудничества с вендорами Delta Design Simtera выросла в качестве моделирования и в количестве сопутствующих инструментов, началось развитие своих собственных инструментов «кремниевой компиляции», защищенной от встраивания «закладок». При реализации решений для Delta Design Simtera учитывается опыт конкурентов и особенности их инструментов для наиболее простого перехода в отечественную среду разработки. Уже сейчас можно сказать, что программный пакет компании «ЭРЕМЕКС» может закрыть некоторую необходимость в разработке конфигурации ПЛИС, особенно российских, но не только, а также некоторых семейств чипов Intel и Xilinx.

