

Delta Design Simtera. Новости первой российской САПР проектирования, моделирования, а также синтеза RTL для конфигурирования ПЛИС

Никита Малышев, Алексей Ерёмин, Артем Варганов, Сергей Ярцев

Сегодня отечественный рынок ПЛИС может предложить аналоги западных микросхем и даже наметилась тенденция по созданию своих продуктов, касательно систем разработки – всё намного сложнее. Российским разработчикам ПЛИС приходится использовать либо системы проектирования западных вендоров, либо системы с открытым исходным кодом. Такая ситуация сохранялась довольно долго, пока компания ЭРЕМЕКС не выпустила программный пакет проектирования конфигурации ПЛИС Delta Design Simtera. В статье рассмотрим последние изменения за год, которые должны появиться в ближайшем релизе продукта.

Введение

На сегодняшний день разработка проектов для конфигурации ПЛИС (программируемых логических интегральных схем) является достаточно востребованной тематикой. За счёт скорости работы, возможности многократной реконфигурации и простоты разработки ПЛИС находят применение в различных сферах – в телекоммуникации, космонавтике, а также в других устройствах военного и гражданского применения.

Важнейшее значение при разработке интегральных схем занимает процесс высокоуровневого описания схемы на языках разработки аппаратуры (HDL), логическая верификация проекта и «кремниевая компиляция» (синтез). К сожалению, ввиду отставания отечественной микроэлектроники от западной наметился значительный разрыв и в области систем автоматизированного проектирования. В современных

реалиях довольно сложно спроектировать конфигурацию логической схемы, состоящей из десятков и сотен миллионов логических элементов в ручном режиме. А в приемлемо короткий промежуток времени – нереально.

Маршрут проектирования. Разработка высокоуровневого описания

Маршрут проектирования конфигурации программируемых логических интегральных схем всем знаком. Он представляет из себя итерационный процесс и его схему можно представить следующим образом – см. рис. 1. В статье разберём шаги высокоуровневого описания, логической верификации, синтеза схем, реализованных в Delta Design Simtera.

Начнём с первого этапа – разработки высокоуровневого описания. Вести разработку можно в схематехническом, а также в текстовом виде. При

работе с последним довольно важным является процесс взаимодействия системы и разработчика. Современный разработчик привык к интерактивным системам, дающим информацию по ошибкам, «узким местам» разрабатываемого кода, а также показывает их и даёт рекомендации. Тем самым процесс создания HDL-кода становится быстрее и, как правило, лишён грубых ошибок. В системе Delta Design Simtera реализовано несколько инструментов в помощь программистам ПЛИС – это визуальные и интерактивные инструменты. К визуальным можно отнести редактор настройки подсветки синтаксиса, отвечающий за настройку цвета, шрифта, размера и других параметров ключевых слов для HDL-языков, использующихся в проекте. Присутствует в системе и настройка цветовых схем редакторов. К интерактивным инструментам относятся автоподсказки, автодополнения кода, система распознавания лексических, синтаксических и смысловых (семантических) ошибок кода.

Технология умного ведения кода проекта предлагает варианты при написании кода, уменьшает время написания кода и возможность совершения синтаксических ошибок (рис. 2). В зависимости от контекста система предлагает варианты дальнейшего описания по стандартам на языки IEEE 1800-2005 (SystemVerilog), 1364-2005 (Verilog), IEEE 1076-2008 (VHDL) (рис. 3).

Моделирование и верификация HDL

Верификация и моделирование являются одними из этапов проектирования конфигурации ПЛИС. Моделирование осуществляется на основе скомпилированного HDL-проекта и только в том случае, если он не содержит ошибок. Моделирование происходит при нажатии на кнопку «Собрать проект и запустить симуляцию», показанную на рис. 4. В панель «Журналы» выводится информация об этапах и шагах моделирования, а также выводится отладочная информация разработчика ПЛИС (рис. 5).

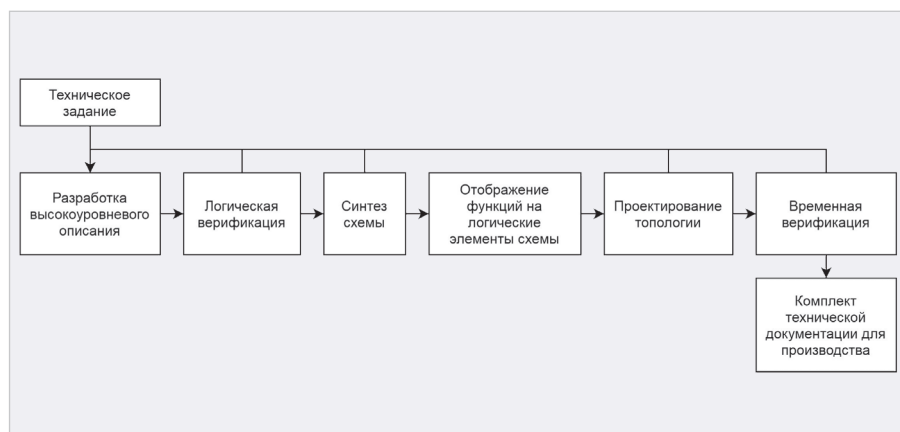


Рис. 1. Маршрут разработки проекта ПЛИС

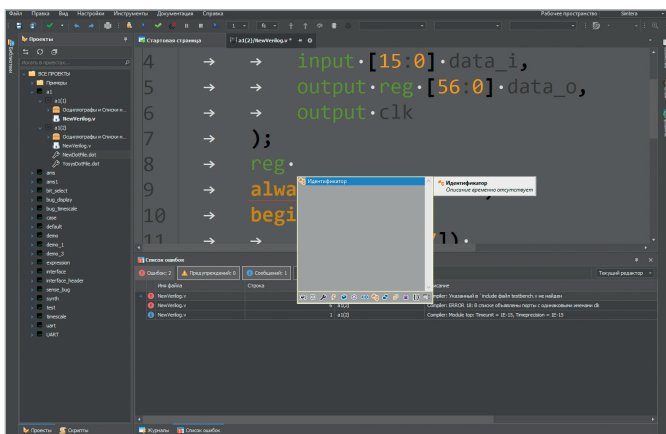


Рис. 2. Автодополнение Verilog-кода в зависимости от контекста

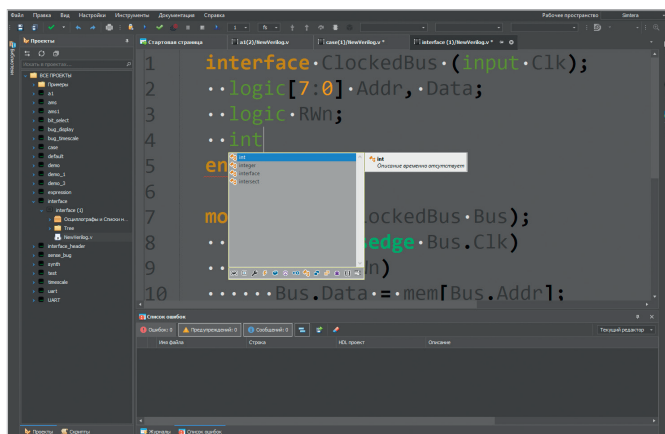


Рис. 3. Дополнение кода в зависимости от введённых символов

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, а в верхней части – абсолютное значение времени (рис. 6).

Поведенческий синтез. Проверка на синтезируемость

HDL-язык, и язык Verilog в частности, используется для описания поведения схемы во времени и запуска соответствующего моделирования. Однако воплощение реализующей такое поведение цифровой аппаратуры (т.е. логических элементов и соединяющих их проводников) возможно не всегда. Логический синтез – перевод поведенческой модели в набор цифровых компонентов – может быть выполнен при использовании некоторого подмножества языка Verilog, который называется Synthesizable Verilog (синтезируемый).

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 трудоёмкий процесс логического синтеза. Это выполняется путём добавления к проверке семантики дополнительных правил, следование которым проверяется сразу же при запуске синтеза, и в случае их нарушения выдаётся ошибка, а процесс синтеза не стартует.

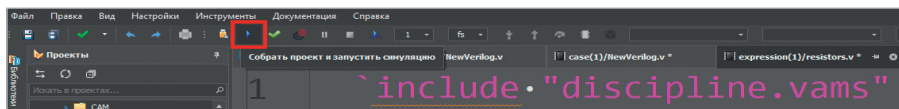


Рис. 4. Запуск моделирования HDL-проекта

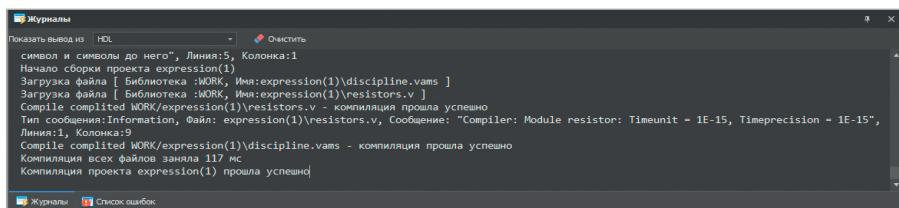


Рис. 5. Панель «Журналы» с отладочной информацией

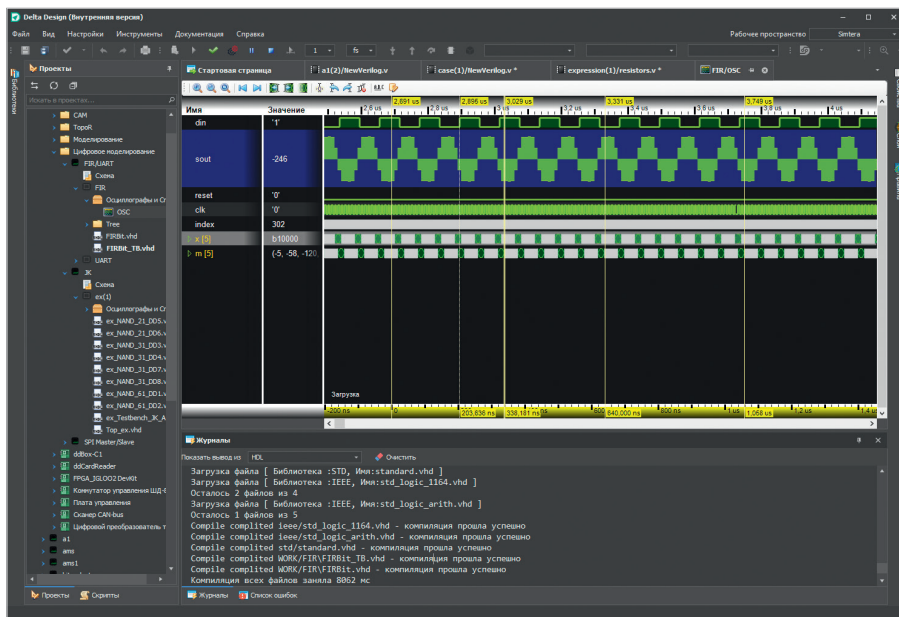


Рис. 6. Осциллограф Delta Design Simtera

В качестве примера работы системы на проверку синтеза рассмотрим Verilog-код, в котором используется несинтезируемая конструкция **event**. Verilog-код:

```

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

```

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

Рис. 7. Предупреждение системы проверки на синтезируемость

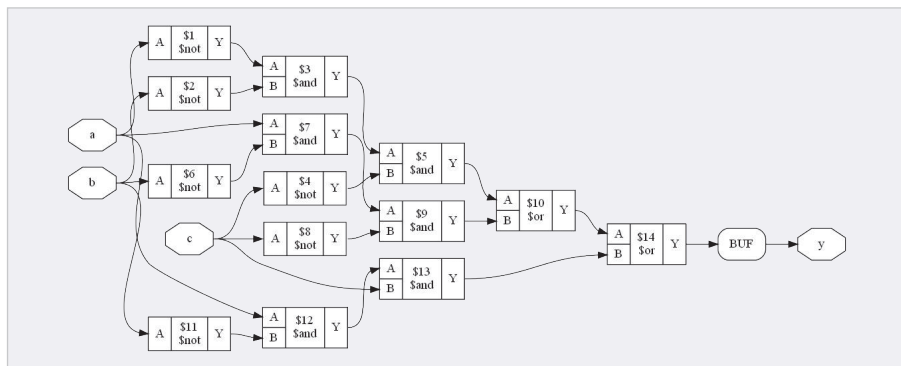


Рис. 8. Результаты поведенческого синтеза в Yosys

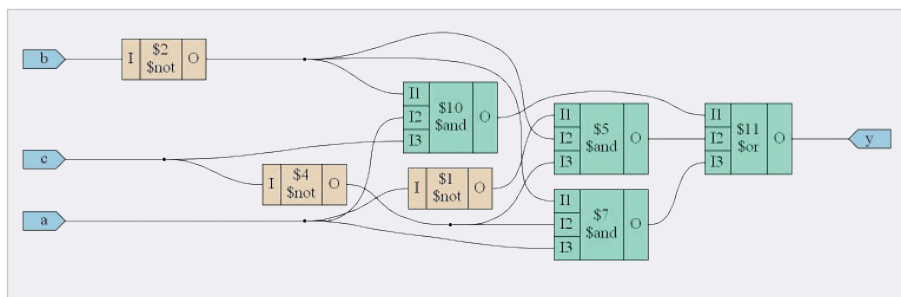


Рис. 9. Результаты поведенческого синтеза в Delta Design Simtera

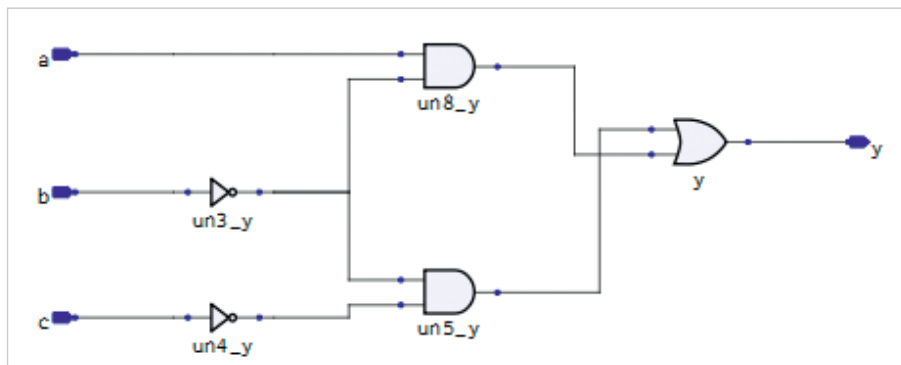


Рис. 10. Результаты поведенческого синтеза в Synopsys Synplify

В нём присутствует несинтезируемая конструкция *событие*.

В Delta Design Simtera компиляция проходит без ошибок, но при проверке синтезируемости кода также выдаётся ошибка – рис. 7.

Поведенческий синтез в Delta Design Simtera

Синтез RTL (Register transfer level) по коду Verilog реализован в наиболее простом варианте. В первом приближении для реализации синтеза в системе присутствует построение схем связей высокоуровневых элементов: переменных (портов, локальных переменных, чисел, параметров); бло-

ков выполнения унарных, бинарных и имеющих большее число аргументов операций; блоков осуществления комбинационной и синтезируемой последовательной логики (защёлок, мультиплексов, D-триггеров и их разновидностей, шифраторов/дешифраторов), а также, если переменная представляет собой массив, обеспечение работы с его частями. При этом оптимизация выполняется только над используемыми высокоуровневыми элементами, то есть, например, сумматор в текущем варианте представляет собой «чёрный ящик», и его внутренние элементы И/ИЛИ/НЕ не являются компонентами синтезиро-

ванной схемы и не участвуют в оптимизации.

Синтезатор Synplify, как наиболее известный по оптимизации ресурсов «кремниевый компилятор», осуществляет довольно глубокую оптимизацию уже на уровне создания принципиальной схемы. И, зачастую, даже в представлении схемы с высокоуровневыми компонентами проведена оптимизация. В ней явно видимые из кода компоненты заменены другими – обеспечивающими более оптимальное представление всей схемы на уровне И/ИЛИ/НЕ (соответственно и в RTL).

Delta Design Simtera по функциональным возможностям можно сравнить с инструментом синтеза с открытым исходным кодом Yosys. Ниже приведены сравнительные результаты работы поведенческого синтеза систем Delta Design Simtera, Yosys и Synopsys Synplify.

Примеры работы Булева функция

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
    
```

Результат работы Yosys представляет собой все операции в графической форме. 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) (рис. 8). В Delta Design Simtera на настоящий момент реализована процедура повторного использования уже созданных блоков, что приводит лишь к однократному появлению отрицаний каждой из переменных. Кроме того, блоки И и ИЛИ на схеме изображаются от трёх операндов, как и записано в коде (рис. 9). Никакой минимизации здесь не проводится: лишь повторное использование уже созданных блоков. Для примера покажем работу Synplify, который путём минимизации записанной функции снизил число операторов до 5 (отрицание a, например, вовсе не понадобилось) (рис. 10).

Защёлки и мультиплексы

Комбинационная логика, описываемая условными операторами, реализуется с помощью мультиплексов (осу-

шествяющих выбор одного из двух входящих сигналов в зависимости от контрольного сигнала) и защёлки (пропускающих или нет входящий сигнал в зависимости от контрольного сигнала). В примере ниже сигнал clk отвечает за выбор сигнала d0 или d1, передаваемого в биты [1:0] выходного порта q. Обратите внимание, что в части else присвоение происходит во все четыре бита q[3:0], а в части if — только в первые два q[1:0]. Вследствие этого для битов q[1:0] создаётся мультиплексор, который по значению clk==0 пропускает d0[1:0] (выбор двух нижних битов d0 производится автоматически в соответствии с тем, что именно попадает в q[1:0] в части else), а по clk==1 — d1. Для битов q[1:0] создаётся защёлка, работающая при clk==0, то есть при ~clk==1.

```
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 выдаёт в качестве результата ссылку на некую процедуру, которую он не может представить в виде цифровых компонентов (рис. 11), на рис. 12 Simtera реализует описанную выше логику, такой же результат выдаёт Synplify – рис. 13.

Заключение

На текущий момент на российском рынке САПР Delta Design Simtera явля-

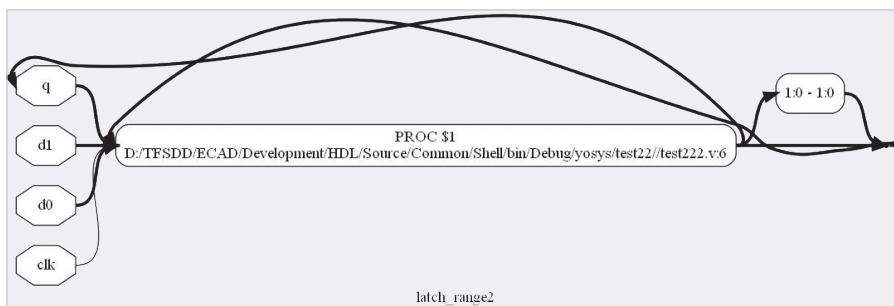


Рис. 11. Yosys. Поведенческий синтез

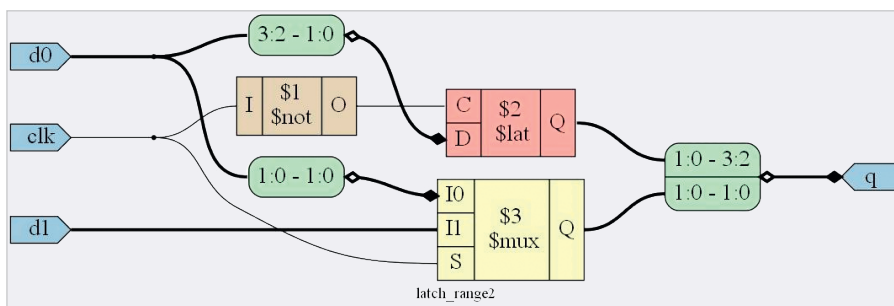


Рис. 12. Delta Design Simtera. Поведенческий синтез

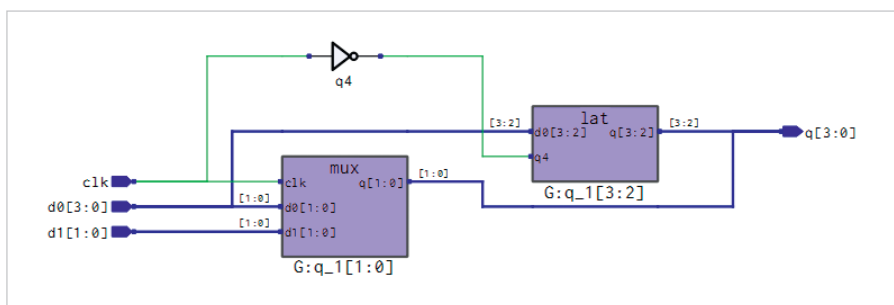


Рис. 13. Synopsys Synplify. Поведенческий синтез

ется хорошей альтернативой западным системам по проектированию конфигурации ПЛИС и может закрыть потребности разработчиков на таких этапах, как проектирование, симуляция и пове-

денческий синтез. При этом система постоянно модернизируется и дополняется новым функционалом, предоставляется техническая поддержка и обучение.



НОВОСТИ МИРА

Техпроцесс TSMC N4X ПРЕНАЗНАЧЕН ДЛЯ ВЫПУСКА МИКРОСХЕМ, ОТ КОТОРЫХ ТРЕБУЕТСЯ МАКСИМАЛЬНАЯ ПРОИЗВОДИТЕЛЬНОСТЬ

Компания TSMC представила техпроцесс N4X, оптимизированный по критерию производительности изготавливаемых микросхем. Это первый техпроцесс такого рода среди техпроцессов TSMC 5-нанометрового поколения. Изготовленные с его применением микросхемы рассчитаны на наибольшие тактовые частоты среди микросхем, изготавливаемых по техпроцессам этого поколения. Техпроцессы TSMC, оптимизированные по критерию производи-

тельности, можно узнать по букве «X» в обозначении.

Оптимизация затрагивает не только транзисторы, которые рассчитаны на повышенные токи и частоты, но и внутренний металлический стек. Кроме того, новый техпроцесс позволяет формировать конденсаторы сверхвысокой ёмкости со структурой «металл-изолятор-металл».

Как утверждается, всё это позволяет получить выигрыш по производительности до 15% по сравнению с N5 или до 4% по сравнению с N4P при напряжении питания 1,2 В. Дополнительный потенциал увеличения производительности обеспечен возможностью использования управляющих напряжений более 1,2 В. Заказчики могут ускорить раз-



работку микросхем для N4X, воспользовавшись тем, что для этого техпроцесса подходят общие правила проектирования для техпроцесса N5. В TSMC ожидают, что рискованное производство с применением N4X начнётся в первой половине 2023 года.

ixbt.com