

Развитие отечественных САПР проектирования микроэлектроники на платформе Delta Design

Н. Малышев¹

УДК 621.3 | ВАК 2.2.2

Уход с российского рынка крупных международных игроков в области проектирования радиоэлектроники стал большим толчком к развитию и узнаваемости отечественных систем. САПР Delta Design российской компании «ЭРЕМЕКС» уже больше 10 лет присутствовала на рынке и давала возможность «бесшовного» перехода с продуктов компаний Altium и Mentor Graphics при разработке печатных плат без потери функциональности. Кроме того, в пакете предусмотрены собственные уникальные конкурентные решения для моделирования и трассировки. На платформе Delta Design разрабатывались также инструменты, на основе которых появились продукты DeltaCAM [1], SimOne [2], Delta Design Simtera [3], Delta Design Simtera IC. В статье рассмотрено развитие последнего продукта из этого списка – программного пакета Delta Design Simtera IC.

Развитие продукта Delta Design Simtera началось в начале 2010-х годов и на тот момент он представлял собой отдельный программный пакет по моделированию VHDL-кода. Основным функционалом, дополнительно к моделированию, была генерация нетлиста по электрической схеме. При разработке схемотехнического описания использовались инструменты ведения библиотек, создания компонентов – их условно-графического описания (УГО), создания модели высокоуровневого описания и генерации его шаблона на основе УГО. На рис. 1 представлено окно создания условно-графического обозначения D-триггера и его наполнение моделью высокоуровневого описания на языке Verilog с помощью генератора кода. Далее, на основе базы компонентов, разработчику предлагается перейти к схемотехническому проектированию, генерации нетлиста на выбранном языке описания аппаратуры и его моделированию (рис. 2). Стоит отметить, что генерация нетлиста и моделирование производятся в несколько нажатий кнопок мыши – система автоматически производит все необходимые операции и выдает результаты моделирования в виде временных меандров и сообщений (assertions).

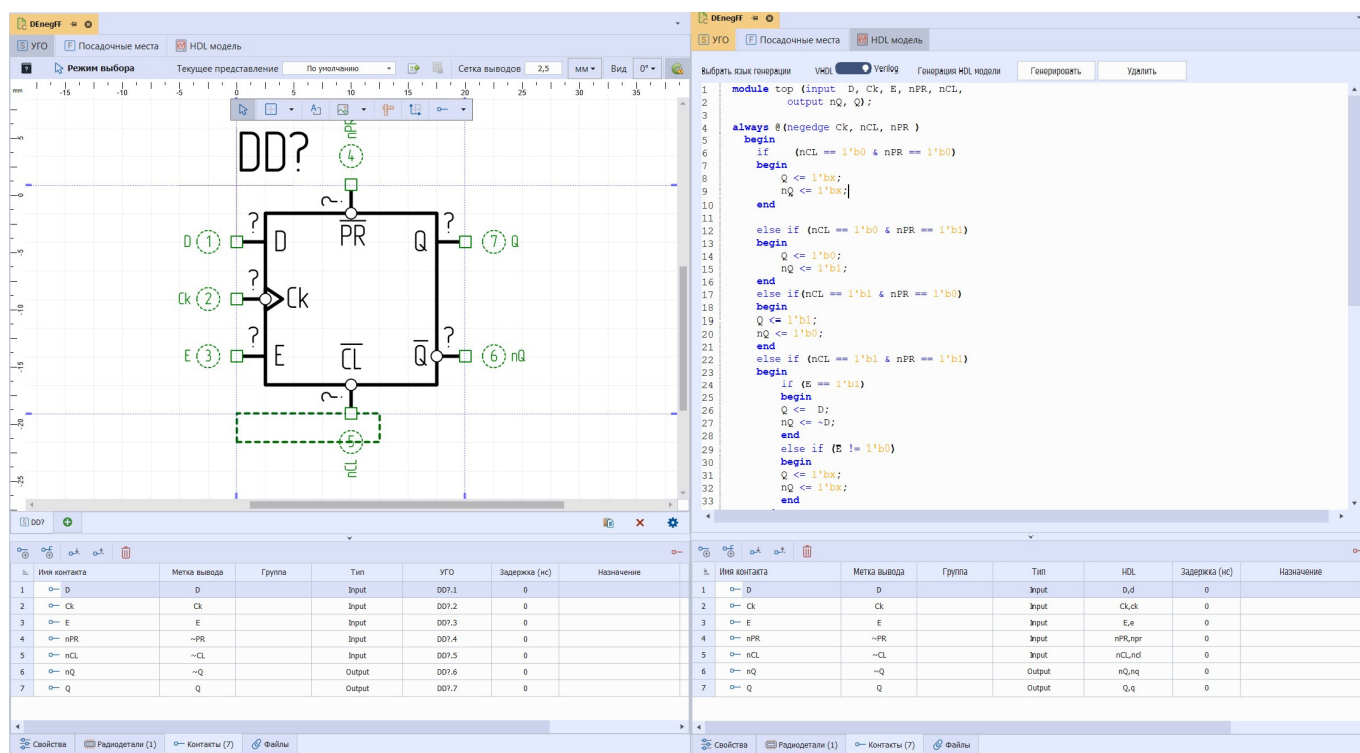
Ввиду спроса и внешних обстоятельств позиционирование продукта изменилось в сторону расширения функциональности. Так, появилась поддержка языков

Verilog и SystemVerilog, вплоть до стандарта 2017 года (IEEE 1800-2017, последнего доступного на текущий момент). Стоит отметить, что продукт Delta Design Simtera постоянно развивается и в нем предполагается поддержка последнего стандарта 2023 года. IEEE, Институт инженеров электротехники и электроники, обновил стандарт SystemVerilog в декабре прошлого года.

Требования к Delta Design Simtera на сегодняшний день предъявляются как к системе симуляции и верификации HDL-проектов. По своей сути, данный продукт способен заменить в работе Questa Sim компании Mentor Graphics (ныне – Siemens). При прямом сравнении двух систем видно, что на текущий момент в Simtera присутствует весь необходимый функционал – интегрированное управление проектом, шаблоны и помощники по исходному коду, поддержка VHDL 1987, 1993, 2002, 2008 (кроме VHPI, PSL), Verilog 1995, 2001, 2005, 2009, платформо-независимая скомпилированная база данных, интерактивная отладка и др. В табл. 1 представлена сводная информация по функциональности систем для сравнения.

Поддержка расширенных инструментов верификации HDL-проектов, таких как OVM/UVM, Verilog PLI/VPI, в пакете моделирования Simtera ожидается в ближайшее время в обновлениях, которые можно получить на сайте компании (eremex.ru) и в Telegram-канале проекта Simtera [4]. Реализация поддержки SystemC, VHDL FLI, VHDL VHPI, отладчика языка C пока не планируется, но, возможно, появится в случае интереса и запроса со стороны клиентов.

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



a)

б)

Рис. 1. Окна создания библиотечного компонента: а – условно-графическое обозначение D-триггера; б – Verilog-модель D-триггера

Дополнительно к наличию требуемого функционала, система моделирования и верификации должна обладать достаточно высокой скоростью работы. Сравнительный анализ показал, что скорость компиляции проектов в Questa Sim и Simtera практически одинакова (последняя несколько выигрывает). По скорости моделирования Simtera отстает от своего конкурента, но это вопрос доработки и оптимизации системы, которые проходят при каждом новом обновлении. Разрыв сокращается, а отставание в скорости моделирования обусловлено юностью отечественной системы, тогда как продукт компании Siemens присутствует на рынке более 30 лет.

HDL-языки, и язык 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, возможно не всегда. Логический синтез – перевод поведенческой модели в набор цифровых компонентов – может быть выполнен при использовании лишь синтезируемого подмножества языка (Synthesizable Verilog/VHDL). Вполне корректный с точки зрения моделирования код может содержать

несинтезируемые конструкции. Логический синтезатор должен предупредить пользователя о наличии нереализуемых для создания схемы соединений (нетлиста) языковых конструкций. Основная задача логического синтеза – создать схему по высокоуровневому описанию на HDL-языке с использованием технологических компонентов ПЛИС или СБИС, которые могут быть представлены, в том числе, в формате Liberty. На рис. 3 показан результат синтеза 4-битного арифметико-логического устройства 74181 по его Verilog-описанию [5], с использованием Liberty-библиотеки. В следующем листинге представлено сокращенное содержимое Liberty-библиотеки с единственным функциональным полем function, описывающим логическое поведение выходного пина компонента:

```

library(Nanocron) {
    cell(AN2) {
        area: 5.488;
        pin(A) { direction: input; }
        pin(B) { direction: input; }
        pin(Z) { direction: output;
            function: «A&B»; }
    }
}

```

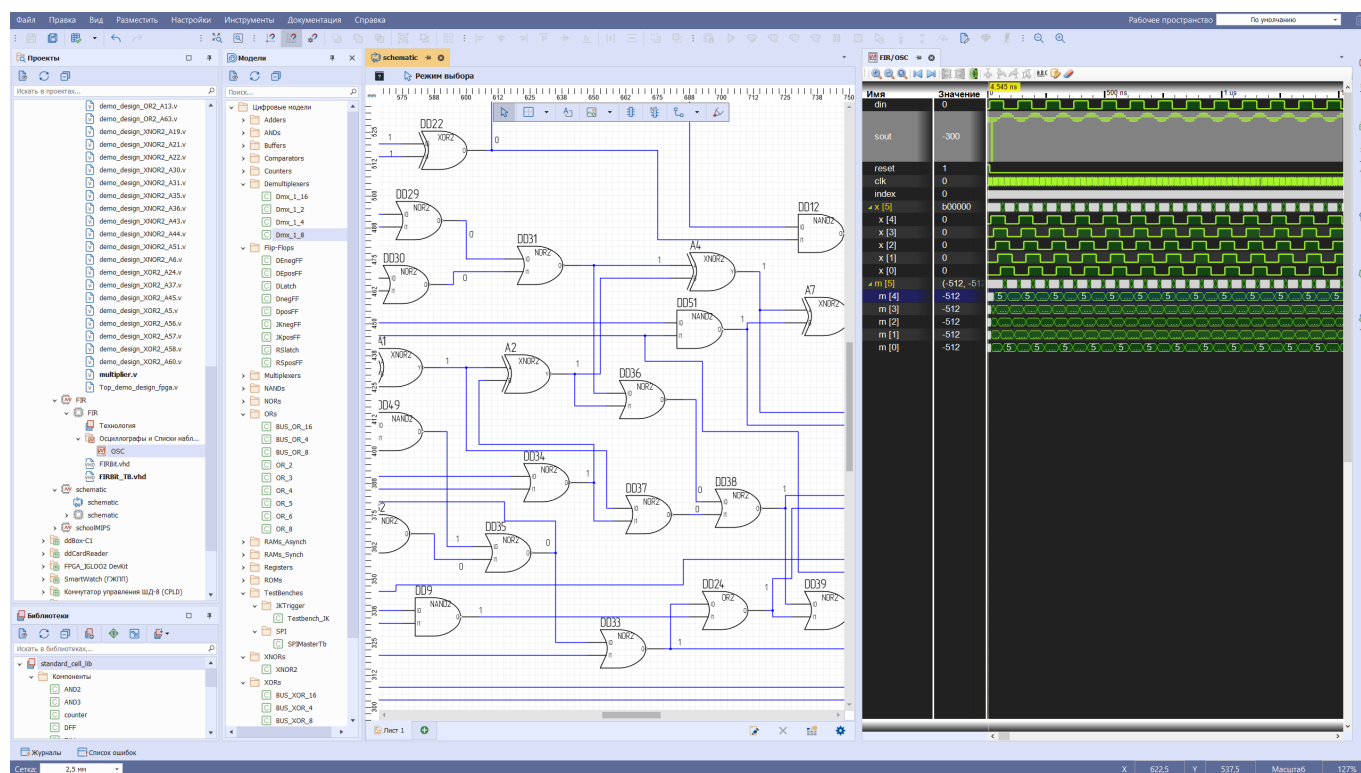


Рис. 2. Рабочее пространство для работы с цифровым проектом, представленным в схематехническом виде.

Слева направо – панель проектов, встроенная библиотека компонентов, схематехнический редактор, осциллограф с результатами моделирования проекта

```
cell(OR2) {
    area: 5.488;
    pin(A) { direction: input; }
    pin(B) { direction: input; }
    pin(Z) { direction: output; function: «A|B»; }
}
...
```

Логический синтезатор должен анализировать и оптимизировать нетлист проекта согласно различным характеристикам, заложенным в его HDL-описании, с учетом требований, которые указываются в качестве атрибутов и ограничений. С другой стороны, синтезатор может использовать лишь те компоненты, которые разработчик указывает в качестве доступных. Так, в Liberty-библиотеке присутствуют дополнительные поля с указанием временных и мощностных характеристик компонентов, а к проекту прикладываются SDC-ограничения. Кроме того, задача логического синтезатора – подобрать оптимальную схему с наименьшей площадью, то есть сократить количество компонентов для удешевления стоимости производимого изделия. Если говорить о синтезаторах, то собственных решений на российском рынке нет.

В рамках работы по расширению функциональности и созданию системы полного цикла проектирования командой компании «ЭРЕМЕКС», работающей над системой цифрового моделирования, были проведены работы по созданию собственного логического синтезатора. На основе компилятора системы верификации были созданы новые пакеты, а именно: пакет проверки синтезируемости Verilog-проектов, высокоуровневый синтезатор, логический синтезатор, оптимизатор, системы распознавания атрибутов синтеза и файлов ограничения стандарта Synopsys (Synopsys Design Constraint), а также Liberty-анализатор. Были заложены инструменты статического временного анализа и проверки логической эквивалентности. О каждом из инструментов поговорим подробнее. Следует отметить, что приведенные результаты работ, о которых будет идти речь в статье, получены в рамках работы с пре-альфа версией системы и имеют больше исследовательский эффект, нежели технический, который уже сейчас позволил бы разработчикам применять данную систему на практике в больших проектах. Под большими проектами следует понимать дизайны, рассчитанные на сотни тысяч, миллионы и более вентилей. Именно такие проекты реализуются в рамках

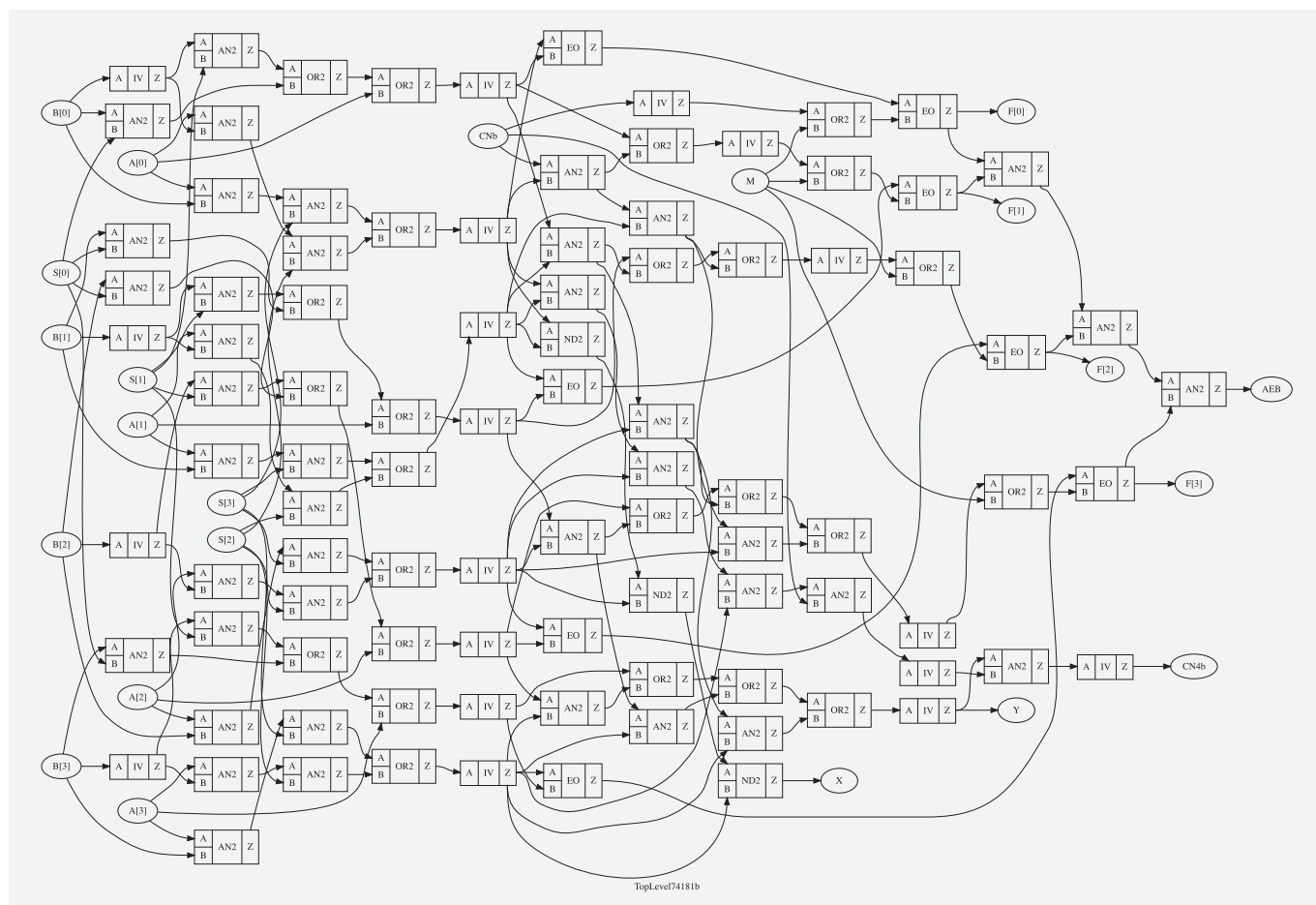


Рис. 3. Результат, полученный при синтезе Verilog-проекта по Liberty-библиотеке в Delta Design Simtera IC

работ над созданием собственных уникальных по функциональности сверхбольших интегральных схем и конфигураций программируемых интегральных схем. Для небольших проектов, которые можно имплементировать в изделия, содержащие до двух тысяч вентилей, например в ПЛИС М3 компании «Миландр» [6] или подобные – например, Altera Cyclone II (EP2C5) – можно использовать систему Simtera IC в качестве логического синтезатора.

Обнаружение несинтезируемых конструкций и предупреждение о них следует выполнять раньше, чем будет запущен трудоемкий процесс логического синтеза. Для этого был разработан модуль проверки синтезируемости Verilog-проектов. В приведенном ниже листинге представлен Verilog-код, содержащий несинтезируемую конструкцию event:

```
module dff(clk,r_data,q);
input clk;
input r_data;
```

❗ Ошибок: 1 ⚠ Предупреждений: 0 ⓘ Сообщений: 1			
...	Имя файла	Строка	HDL проект
❗	file.v	15	schematic(1)
	Описание		
	Ошибка 6903: Синтез событий (event) не поддерживается		

Рис. 4. Предупреждение системы проверки на синтезируемость

```
output reg q;
event e_data;

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

always @(e_data)
q = r_data;

endmodule
```

На рис. 4 представлено сообщение (ошибка) системы при запуске логического синтеза.

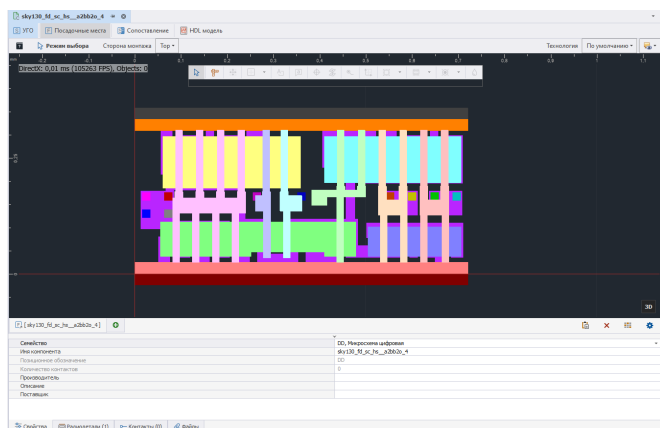


Рис. 5. Отображение GDSII-компонента SkyWater PDK в Delta Design Simera IC

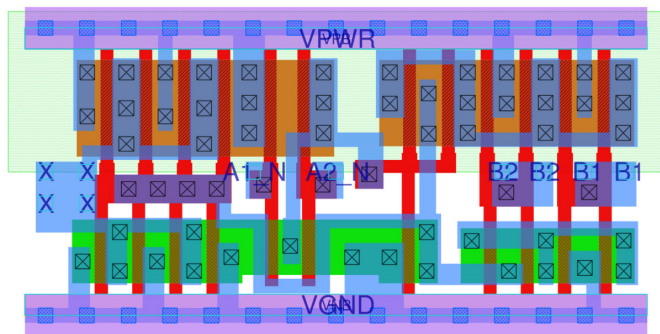


Рис. 6. Отображение GDSII-компонента SkyWater PDK в OpenLane

состоящая из 2-выходов примитивов. При наличии Liberty-библиотеки в проекте запускается ее анализ, расставляются весовые критерии, то есть коэффициенты приоритета использования в конечном нетлисте в зависимости от требований к исходному проекту. По созданной на предыдущем шаге функциональной схеме составляется неоптимизированный технологический нетлист. Следующий этап синтеза – оптимизация технологического нетлиста. Теоретической основой для оптимизации логической схемы является задача минимизации логических функций. Она заключается в приведении логических функций к такой форме, которой соответствует минимальное число операций над логическими переменными. За счет того, что логические (булевы) функции могут быть представлены в различных формах: в виде таблиц, бинарной диаграммы решений, графа и пр. – выделяют алгоритмы, работающие на каждом из видов представлений. Каждый из видов представлений и алгоритмов минимизации в них обладает рядом преимуществ и недостатков. При работе с табличным представлением получение множества всех простых импликант булевой функции уже для числа независимых переменных $n > 10$ может потребовать значительной памяти и большого времени работы ЭВМ [7]. Одним из наиболее интересных с точки зрения минимизации функций для большого количества независимых переменных является подход с использованием ориентированного ациклического графа [8].

При разработке инструментов логического синтеза и оптимизации команда компании «ЭРЕМЕКС» ориентировалась на результаты, полученные с применением

Классическим подходом к задаче реализации синтезатора является «послойное» построение нетлиста по исходному высокоуровневому описанию на HDL-языке. На первой итерации происходит построение схемы с использованием крупных функциональных блоков. Уже на данном этапе происходит определение конечных автоматов, частей комбинаторной и последовательной логики, выделение повторяющихся узлов в отдельные блоки. Следующий этап – анализ технологической библиотеки. В Delta Design Simera встроена generic-библиотека, то есть универсальная библиотека логических компонентов,

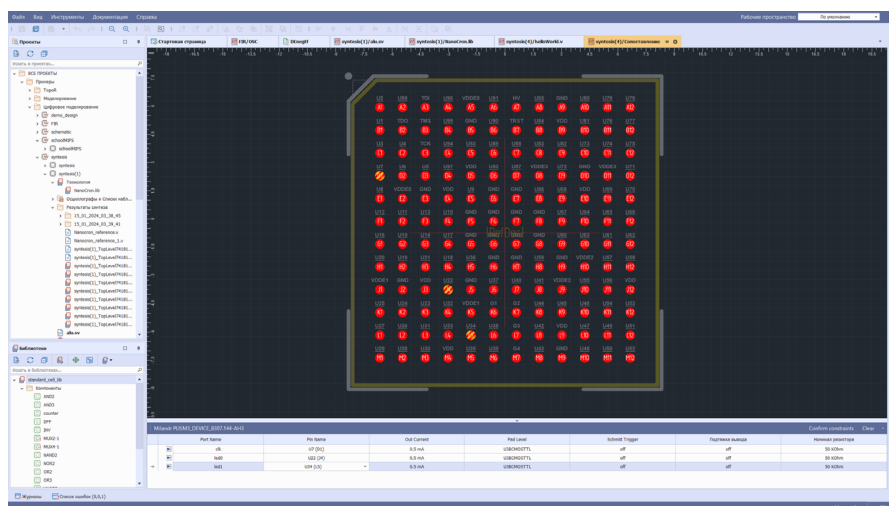


Рис. 7. Графический редактор подготовки файлов ограничений в Delta Design Simera IC

Таблица 1. Основные параметры и их наличие в продуктах

Параметр	Questa Core	Questa Prime	Delta Design Simtera
Редактор HDL	Да	Да	Да
Интегрированное управление проектом	Да	Да	Да
Шаблоны и помощники по исходному коду	Да	Да	Да
VHDL 1987, 1993, 2002, 2008 (кроме VHPI, PSL)	Да	Да	Да
Verilog 1995, 2001, 2005, 2009	Да	Да	Да
SystemVerilog IEEE for Design 2005, 2009, 2012	Да	Да	Да
SystemVerilog IEEE for Verification 2005, 2009, 2012	Только assertions	Да	Assertions – да, полная поддержка в разработке
Платформено-независимая база данных	Да	Да	Да
Инкрементное компилирование	Да	Да	Да
Интерактивная отладка	Да	Да	Да
Отладка после моделирования	Да	Да	Да
Отслеживание причинно-следственных связей	Да	Да	Да
Несколько окон диаграмм сигналов	Да	Да	Да
Просмотр схемы	Да	Да	Да
Единое ядро моделирования	Да	Да	Да
Поддержка VCD и Extended VCD	Да	Да	Да
Смешанный Verilog/VHDL	Опция (MixedHDL)	Да	В разработке, ожидается в ближайших обновлениях
Полная поддержка OVM/UVM	Нет	Да	В разработке, ожидается в ближайших обновлениях
Assertions (PSL для VHDL или Verilog, SVA для Verilog)	Да	Да	SVA – в разработке
Аналоговые/смешанные сигналы	Опция (Questa ADMS)	Опция (Questa ADMS)	Да
Verilog PLI/VPI	Да	Да	В разработке, ожидается в ближайших обновлениях
SystemVerilog Direct Programming Interface (DPI)	Да	Да	В разработке, ожидается в ближайших обновлениях
Оптимизация производительности моделирования и компиляции	Да	Да	Да
SystemC 2.2, 2.3	Опция	Да	Нет
VHDL FLI	Да	Да	Нет
VHDL VHPI	Нет	Нет	Нет
Отладчик C	Да	Да	Нет
Window 64-бит	Win XP, 7, 8.1, 10	Win XP, 7, 8.1, 10	Win 8.1, 10
Linux 64-бит	Да	Да	В разработке, ожидается в ближайших обновлениях

инструментов с открытым исходным кодом, например Yosys, используемого в OpenLane [9]. OpenLane – это САПР сквозного (RTL to GDSII) проектирования СБИС с открытым исходным кодом, которая включает в себя инструменты Yosys, Magic, Netgen, KLayout и др. В упомянутом выше проекте 4-битного АЛУ серии 74181 команде «ЭРЕМЕКС» удалось получить для отдельных блоков совпадающие результаты по площади и количеству компонентов. За счет извлечения вложенных структур и глобальной оптимизации удалось достигнуть сокращения площади кристалла в сравнении с Yosys. Выигрыш составил 5% (226 против 238 мкм).

Система логического синтеза не может правильно и эффективно работать без инструментов проверки логической эквивалентности и статического временного анализа. Первый – проверяет соответствие полученного на этапе синтеза нетлиста исходному HDL-дизайну, второй – проводит расчет временных параметров схемы. Данные инструменты сейчас находятся в разработке, так же как и часть, связанная с физическим проектированием СБИС. Уже реализованы инструменты импорта технологических библиотек и их отображения. На рис. 5 показано отображение компонента библиотеки PDK SkyWater [10] в Simtera, а на рис. 6 – его представление в OpenLane. Видно, что они идентичны.

* * *

За последние несколько лет за счет разработки собственных решений и сотрудничества с вендорами Simtera и Simtera IC выросли в качестве моделирования и в качестве сопутствующих инструментов, началось развитие собственных инструментов «кремниевой компиляции», защищенной от встраивания «закладок». Сегодня программные пакеты компании «ЭРЕМЕКС» могут обеспечить решение задач, возникающих при разработке конфигурации ПЛИС, особенно российских. В качестве примера на рис. 7

представлен графический редактор подготовки файлов ограничений для ПЛИС МЗ компании «Миландр».

ЛИТЕРАТУРА

1. **Попов С.** DeltaCAM – инструмент подготовки производственных файлов // Управление и производство. 2021. Вып. 9. С. 22–28. <https://sapr.ru/article/26299>
2. **Смирнов А., Гимеин А.** Схемотехническое моделирование в Delta Design SimOne. Часть 2 // Современная Электроника. 2022. Вып. 9. С. 34–37.
3. **Малышев Н., Поляков А.** Библиотеки HDL-тестов для систем моделирования цифровой аппаратуры. Отечественная САПР проектирования микроэлектроники. Ч. 1 // Современная Электроника. 2023. Вып. 9. С. 12–15. <https://t.me/+ZnqV7hCNviAxYzMy>
4. **Hansen M.C.** Verilog Behavioral description of the TI 74181 Circuit. <https://web.eecs.umich.edu/~jhayes/iscas.restore/74181b.v>
5. **Шумилин С.** ПЛИС МЗ, ОКР «Бриллиант». <https://www.milandr.ru/upload/iblock/60c/60cdbff6fae9ad208ada6109a1f3cd9f.pdf>
6. **Лузин С.Ю.** Математическое обеспечение синтеза минимальных форм представления переключательных функций для САПР БИС. Санкт-Петербург, 2001, ОАО «НИИ «ЗВЕЗДА», Диссертация на соискание ученой степени доктора технических наук, на правах рукописи.
7. **Gao M., Jiang J-H., Jiang Y., Li Y., Mishchenko A., Sinha S., Villa T., Brayton R.** Optimization of Multi-Valued Multi-Level Networks, In Proc, 32nd IEEE International Symposium on Multiple-Valued Logic (ISMVL'02), Boston, MA, USA, May 2002. P. 168–177. Доступно по ссылке: <https://people.eecs.berkeley.edu/~alanmi/publications/2002/ismvl02.pdf>
8. <https://github.com/The-OpenROAD-Project/OpenLane>
9. <https://github.com/google/skywater-pdk>