

Разработка программного средства для автоматизированной генерации временных ограничений в маршруте проектирования схем в базе программируемой логической интегральной схемы

Н.С. Дьяченко, И.В. Тиунов, С.В. Гаврилов

Национальный исследовательский университет «Московский институт электронной техники»

Аннотация: Статья посвящена разработке инструмента автоматизированной генерации временных ограничений в контексте разработки схем в базе программируемой логической интегральной схемы (ПЛИС). В работе проанализированы актуальные решения в области интерфейсных средств генерации проектных ограничений. Разработана структура данных для средства генерации проектных ограничений, алгоритмы чтения и записи файлов формата проектных ограничений Synopsys (Synopsys Design Constraints). На основе разработанных структур и алгоритмов реализован программный модуль, который впоследствии внедрен в маршрут проектирования схем в базе ПЛИС системы автоматизированного проектирования (САПР) X-CAD.

Ключевые слова: система автоматизированного проектирования, программируемая логическая интегральная схема, автоматизация, проектные ограничения, разработка, маршрут проектирования, алгоритм, статический временной анализ

Введение

Задача автоматизации стала неотъемлемой частью всех отраслей. В микроэлектронике разработчикам, чтобы схемы оставались актуальными, нужно не только совершенствовать технологии, но и сделать процесс проектирования более оптимальным, стараясь его автоматизировать. Именно поэтому важной частью проектирования интегральных схем является использование систем автоматизированного проектирования (САПР).

Для проектирования интегральных схем в России используются в основном зарубежные САПР, но в связи с политикой импортозамещения возрос спрос на отечественные решения, возникла необходимость разработки отечественных решений.

При проектировании схем, на этапе составления технического задания определяются требования к их характеристикам. Далее, в процессе проектирования, на каждом из этапов проводится проверка соответствия

схемы техническому заданию [1]. Проверка осуществляется посредством специализированных инструментов анализа. Для установления соответствия схемы требованиям ТЗ инструментам необходимо передать информацию о проектных ограничениях, используя совместимые форматы данных. Одним из наиболее широко используемых среди производителей САПР является формат проектных ограничений Synopsys (Synopsys Design Constraints — SDC) [2].

SDC — это формат файла на основе языка команд инструмента (Tool Command Language — Tcl), предназначенный для задания проектных ограничений. Ограничения описываются последовательно с помощью набора специальных команд и их параметров [3]. Файлы формата SDC используются инструментами анализа характеристик схем на различных этапах проектирования. Используя данные об ограничениях из файла, инструмент может проверить соответствуют ли характеристики схемы на данном этапе заявленным требованиям. SDC позволяет задавать ограничения различных типов: от общих ограничений проекта, до временных и физических. Специфика работы в маршруте проектирования схем в базе программируемых логических интегральных схем (ПЛИС) [4,5] и реконфигурируемых систем на кристалле (РСК)[6] подразумевает, что из всего спектра возможностей формата SDC необходимы только те, которые отвечают за временные ограничения, поскольку конечное устройство уже имеет физическую реализацию, и в процессе проектирования определяется только логика его работы.

Для удобства формирования проектных ограничений в современных САПР предусмотрены специализированные интерфейсные средства. В САПР Vivado [7] от компании AMD используется модуль Timing Constraints (рисунок 1). Так же САПР использует формат XDC [8], являющийся расширением SDC. Инструмент имеет удобно организованный интерфейс

основного окна настроек с навигацией по категориям ограничений, и поэтому был взят за основу при разработке главного окна собственного программного модуля генерации файлов проектных ограничений.

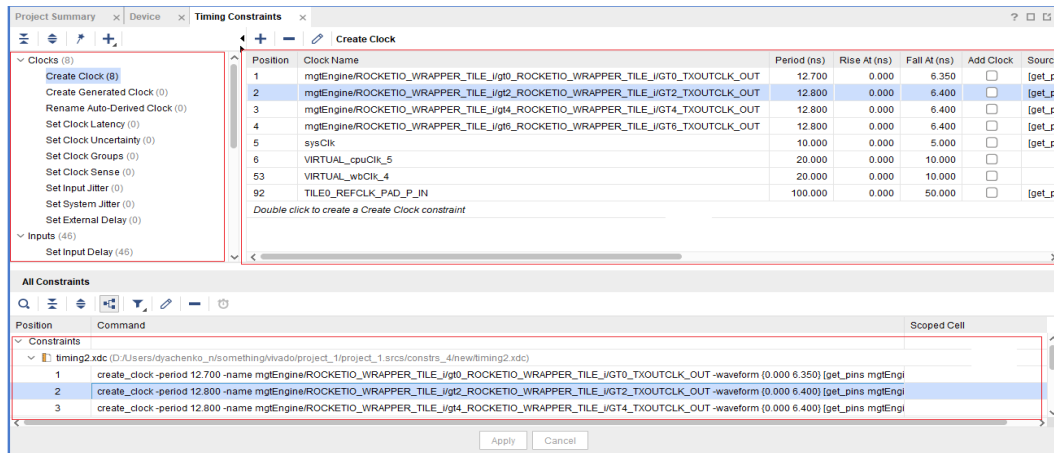


Рис. 1. – Окно инструмента Timing Constraints в САПР Vivado

В Quartus II [9] от компании Intel задание временных ограничений выполняется с помощью инструмента Time Quest Timing Analyzer, представленного на рисунке 2. По сравнению с Vivado, оно выделяется наличием области визуализации формы сигнала. Эта идея взята за основу при разработке соответствующего элемента собственного модуля генерации проектных ограничений.

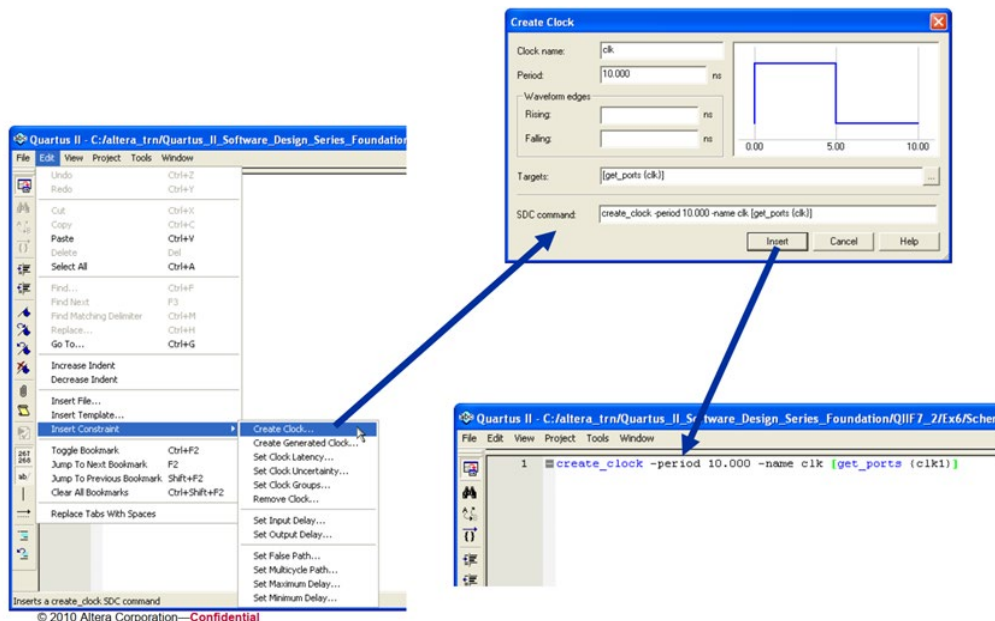


Рис. 2. – Задание проектных ограничений в САПР Quartus II

Одним из отечественных программных комплексов для разработки схем для ПЛИС и РСнК является САПР X-CAD [10,11]. Данный программный комплекс активно развивается и дополняется, расширяя функциональность, но в нем отсутствует инструмент, позволяющий генерировать временные ограничения. В связи с этим, было принято решение разработать соответствующее программное средство с графическим интерфейсом и внедрить его в САПР X-CAD.

Рассмотрим маршрут проектирования X-CAD продемонстрированный на рисунке 3 [12]. Маршрут основан, в большей степени, на собственных разработках Института проблем проектирования в микроэлектронике РАН (ныне НИЦ «Курчатовский институт»), но некоторые этапы (логический синтез, статический временной анализ и функциональное моделирование) реализованы с помощью открытых инструментов. Файл формата SDC используется в данном маршруте на этапе статического временного анализа, который осуществляется с помощью инструмента OpenSTA [13]. Следовательно, инструмент задания проектных ограничений и генерации файлов формата SDC должен применяться перед вызовом OpenSTA.

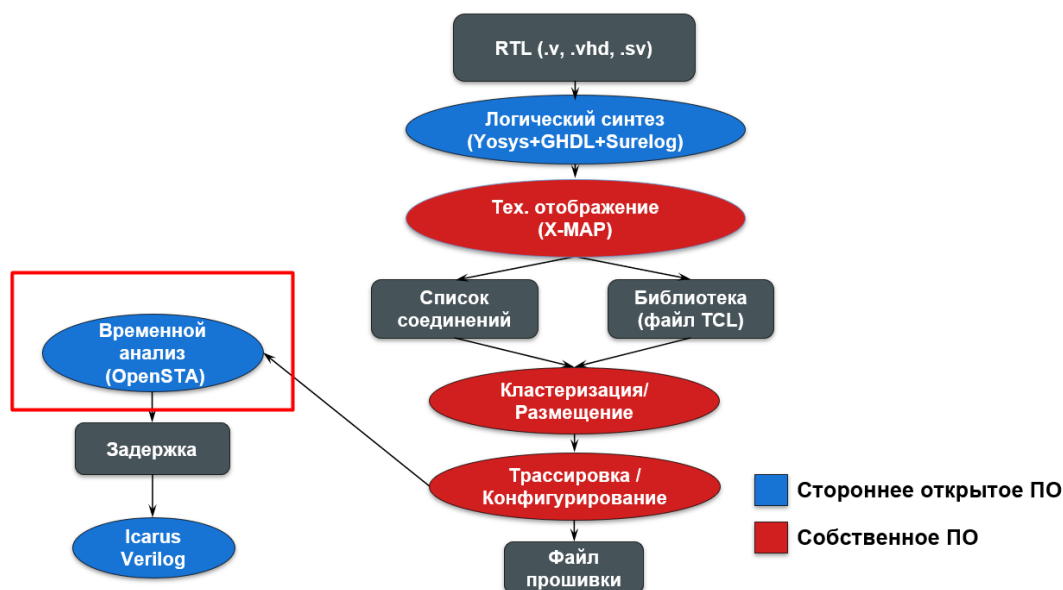


Рис. 3. – Маршрут проектирования в САПР X-CAD

В результате анализа аналогичных инструментов зарубежных САПР был определен визуальный облик инструмента. Также была разработана UML-диаграмма классов и методов, в соответствии с которой он должен быть реализован.

Разработка UML диаграммы и структур данных

При создании UML диаграммы были разработаны: класс главного окна («SdcEditor»), абстрактный класс редактора ограничений («ConstraintEditor»), набор классов редакторов для каждого из ограничений («...Editor»). Для редактора синхросигналов «ClockEditor» разработан виджет для отображения формы тактового сигнала (класс «WaveFormView»). Также для удобства пользователя добавлен интерактивный справочник с информацией о командах SDC, по аналогии с САПР Vivado, в виде отдельного окна, открывающегося по нажатию кнопки во всех окнах модуля (класс «ReferenceBook»). С помощью данной функции можно изучать информацию о разных SDC командах прямо в приложении. Все эти классы, их методы и связи между ними реализованы в виде UML-диаграммы представлены на рисунке 4.

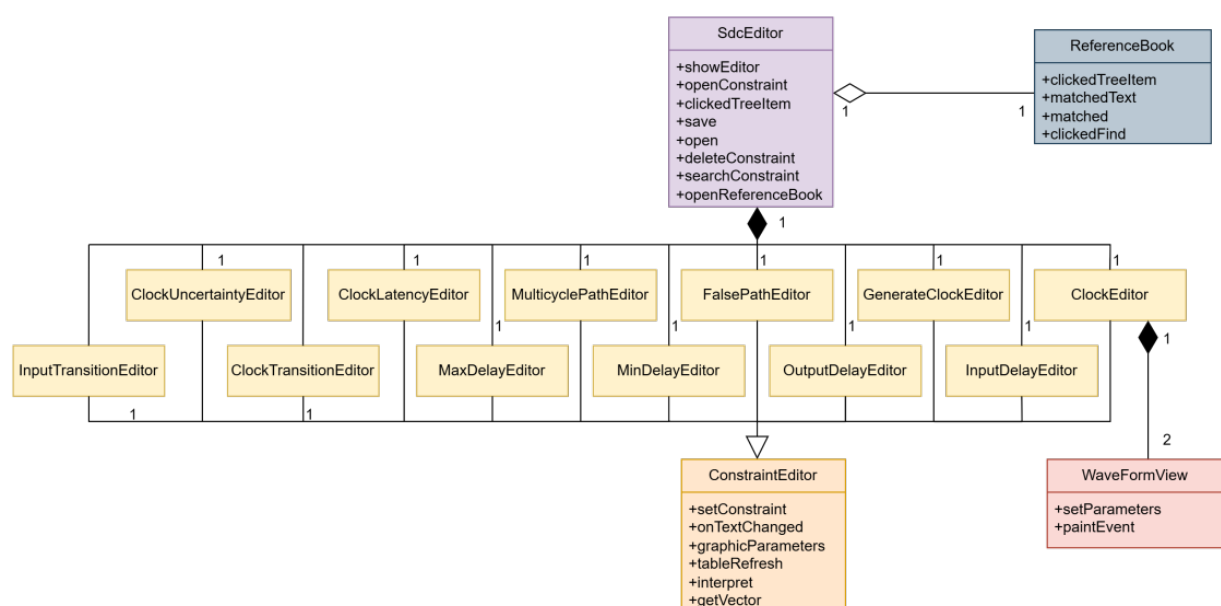


Рис. 4. – UML-диаграмма организации структур программного модуля

Разработав UML-диаграмму с классами и методами, перейдем к алгоритму работы с файлами формата SDC.

Разработка алгоритмов чтения и записи файлов формата SDC

Алгоритм чтения файла SDC (рисунок 5(а)) начинается с передачи на вход программы файла формата SDC. После этого начинается построчное считывание команд, пока не будет достигнут конец файла. Каждая строка представляет собой команду формата SDC, которая передается методу «parseSdc».

Метод «parseSdc» класса SdcEditor, основываясь на первом слове строки, которое является именем команды SDC, передает строку в метод «interpret» (рисунок 5(б)) редактора, соответствующего данному ограничению, где она разбивается на части (токены) и заполняет структуру данных своего класса.

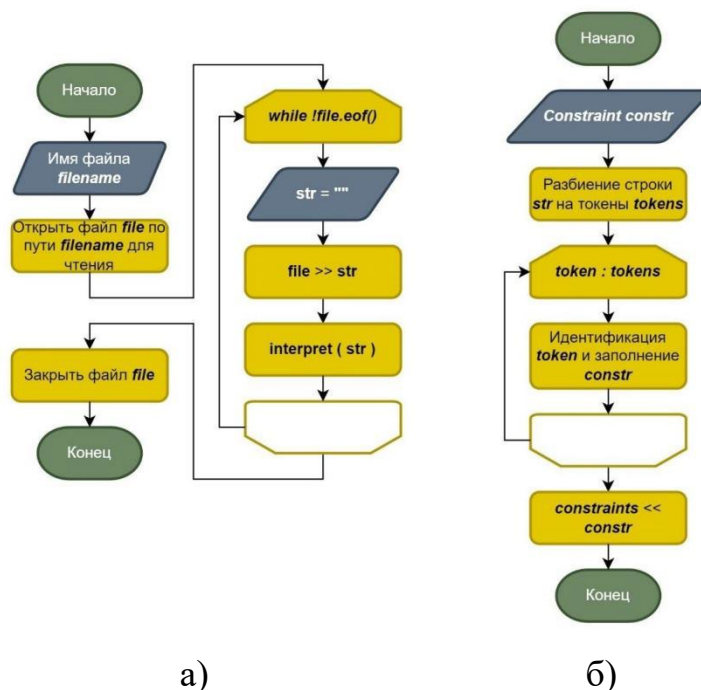


Рис. 5. – Блок-схемы алгоритма чтения файлов (а) и алгоритма работы метода «interpret» (б)

Алгоритм генерации файла, рисунок 6(а), начинается с открытия файла для записи. Далее, модуль обращается к каждому из объектов классов редакторов, чтобы получить список из всех заданных ограничений данного типа. Из каждого объекта этого списка модуль получает сформированную строку команды в формате SDC и записывает ее в файл. По окончании алгоритма, работа с файлом завершается.

Алгоритм для создания строки формата SDC (рис. 6(б)) считывает данные полей редактора и на их основе заполняет структуру данных. Команда формата SDC генерируется из значений данной структуры.

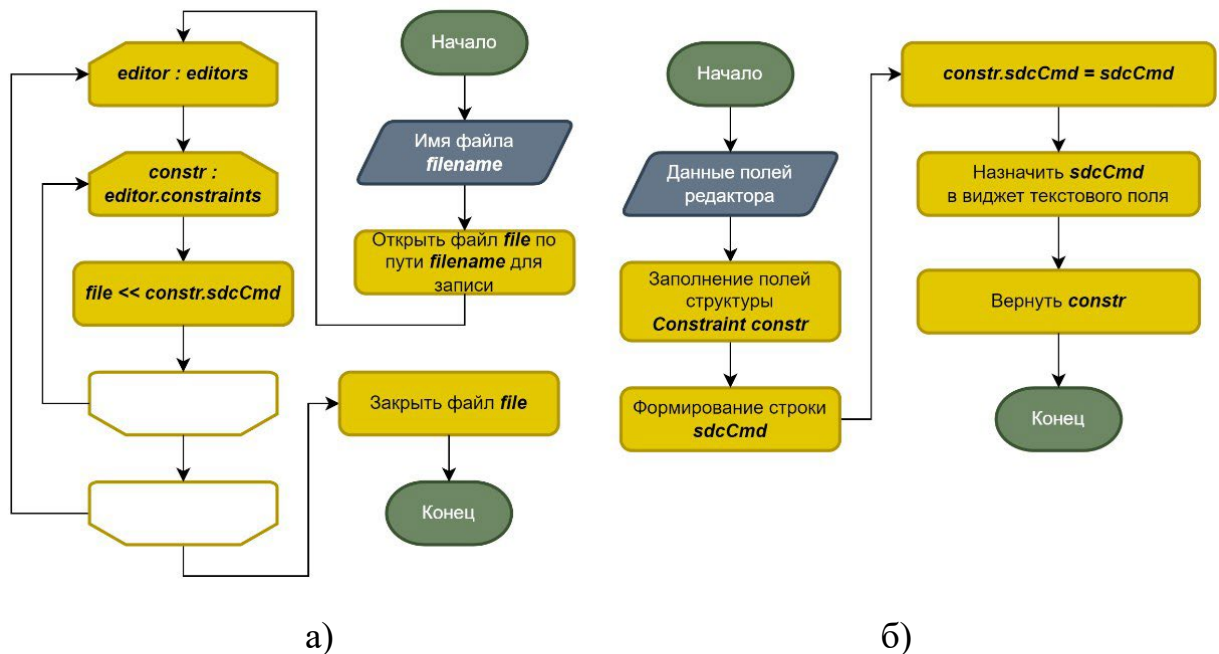


Рис. 6. – Блок-схемы алгоритмов генерации файла (а) и формирования команды SDC (б)

Реализация программного решения

Перейдем к реализации программного модуля, в основу которого будут положены разработанные структуры данных и алгоритмы.

В качестве языка программирования выбран язык C++ из-за его высокой производительности и ввиду того, что САПР X-CAD разработан с применением этого языка. Для разработки графического интерфейса

программы выбрана библиотека Qt, так как интерфейс X-CAD также реализован с ее помощью.

Главное окно разработанного программного модуля представлено на рисунке 7. Внешний вид данного окна, как было сказано раньше во многом основан на таковом в САПР Vivado. В окне присутствует область навигации (на рисунке в левой части), область со списком ограничений выбранного типа, и панель инструментов для добавления, удаления и поиска ограничений, а также для загрузки, и сохранения ограничений в формате SDC. В программном модуле реализованы редакторы для двенадцати команд формата, которые разделены на четыре категории: «Тактовые сигналы» («create_clock» «create_generated_clock» «set_clock_latency» «set_clock_uncertainty» «set_clock_transition», «set_input_transition»), «Входы» («set_input_delay»), «Выходы» («set_output_delay») и «Прочее» («set_false_path», «set_multicycle_path», «set_max_delay», «set_min_delay»).

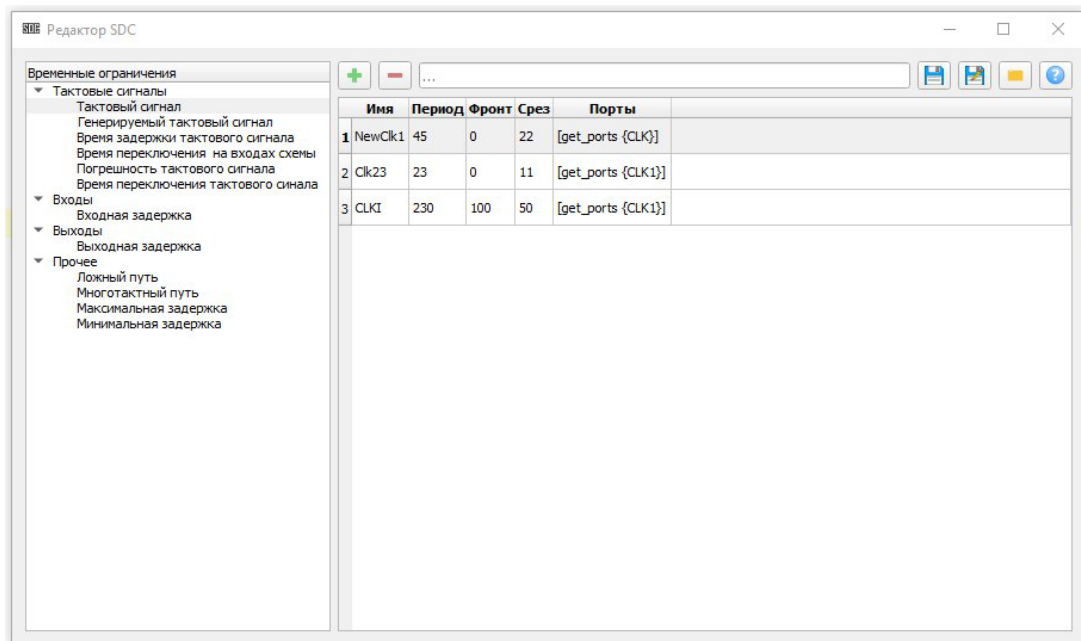


Рис. 7. Интерфейс главного окна, разработанного программного модуля

Информацию о поддерживаемых командах формата SDC можно прочитать во встроенном интерактивном справочнике программного модуля,

ссылки на который в виде кнопки расположены в каждом из окон редактора. Внешний вид справочника представлен на рисунке 8.

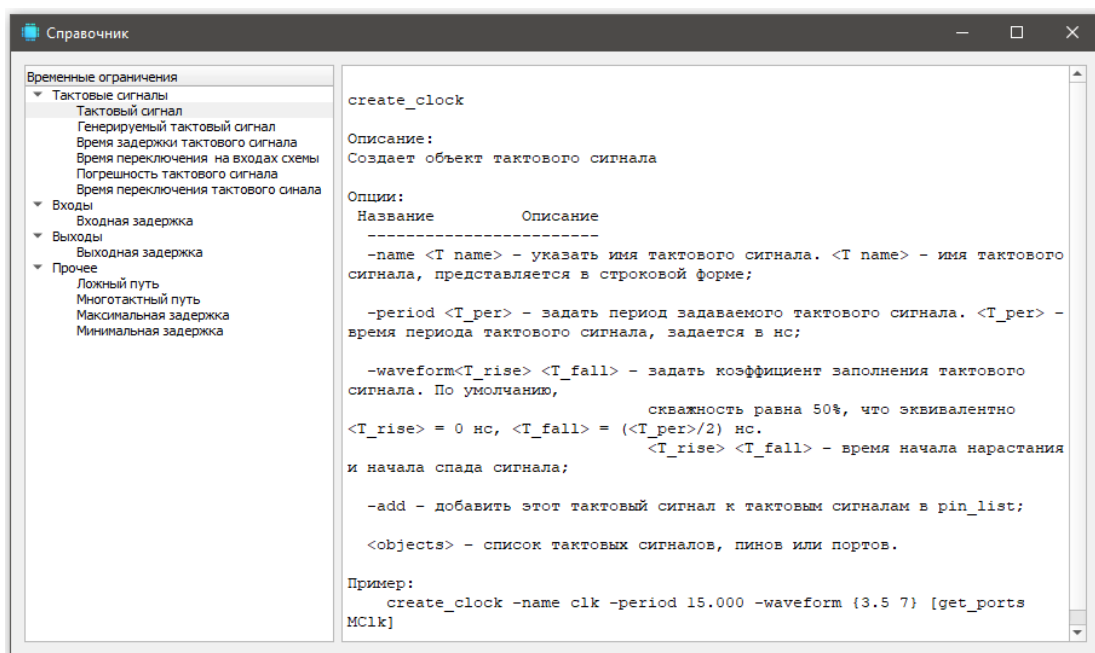


Рис. 8. Интерактивный справочник по поддерживаемым командам формата SDC

Пример работы приложения

Для создания файла ограничений, необходимо в области навигации выбрать тип ограничения, который требуется задать (в левой части рисунка 9), после чего либо нажать на нее двойным кликом левой кнопкой мыши, либо нажать на кнопку со знаком «+». После этого откроется окно для задания параметров выбранного ограничения (в данном примере рассматривается окно редактирования команды «create_clock», которая задает параметры синхросигнала). В открывшемся окне пользователь заполняет поля нужными значениями. Часть этих полей являются обязательными, а часть – опциональными, и если какие-либо необходимые поля не были заданы пользователем, то, при попытке добавить ограничение, программный модуль выведет соответствующее сообщение. По окончании ввода параметров, если пользователь желает сохранить данные настройки, он

должен нажать на кнопку «Добавить», после чего введенные значения сохраняются и добавляются в таблицу в главном окне программного модуля.

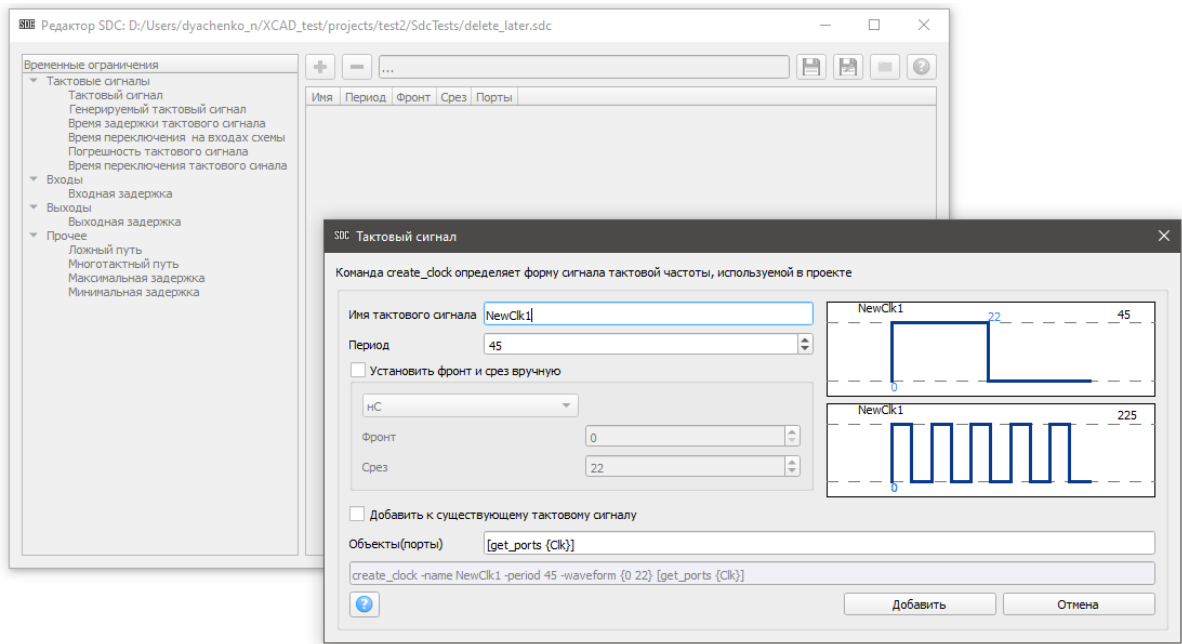


Рис. 9. – Пример задания команды программного модуля

При необходимости, пользователь может отредактировать ранее заданные значения путем двойного клика левой кнопкой мыши на нужном элементе списка, после чего откроется окно с параметрами данного ограничения, куда пользователь может внести правки. Таким же образом пользователь может задать и ограничения других типов.

После того, как пользователь внес все необходимые данные, он может сохранить их в виде файла формата SDC. Для этого необходимо нажать на соответствующую кнопку «Сохранить» с иконкой в виде синей дискеты. Пример полученного файла с набором команд продемонстрирован на рисунке 10.

```
*newConstraints.sdc X
1 create_clock -name NewClk1 -period 45 -waveform {0 22} [get_ports {Clk}]
2 create_generated_clock -name gclk1 -source clk1 -divide_by 4 r1/Q
3 set_multicycle_path 18 -end -reset_path -rise_through Dt1
4 set_clock_latency 3.23 -fall -min -clock clk [get_pins Rst1]
5 set_input_transition 4.2 -fall -max [all_outputs]
6 set_clock_uncertainty 23.3 -setup -fall -from_clock Dt2
```

Рис. 10. Сгенерированный файл ограничений формата SDC

Заключение

В рамках проведенной работы проведен анализ актуальных решений в области интерфейсных средств генерации проектных ограничений. Также разработана структура данных для средства генерации проектных ограничений и алгоритмы чтения и записи файла проектных ограничений. На основе разработанных структур и алгоритмов реализован программный модуль с графическим интерфейсом для автоматизированной генерации файлов проектных ограничений. Программный модуль внедрен в маршрут отечественной САПР X-CAD.

В перспективе планируется провести доработки программного модуля, расширив набор поддерживаемых команд формата SDC не только для маршрутов проектирования схем в базисе ПЛИС и РСнК, но и для маршрутов заказного проектирования. Кроме того, планируется доработать взаимодействие разработанного программного модуля и САПР. Например, для получения списка выводов схемы, из которого можно было бы выбрать нужный порт, для которого ограничение задается, вместо того, чтобы задавать имена сигналов вручную.

Данная работа выполнена при финансовой поддержке Минобрнауки в рамках государственного задания на выполнение научно-исследовательской работы по Соглашению №075-03-2024-061/5 от 21.05.2024г. «Разработка методики прототипирования электронной компонентной базы на отечественных микроэлектронных производствах на основе сервиса MPW».

Литература

1. Казенов Г. Г. Основы проектирования интегральных схем и систем. — М.: БИНОМ. Лаборатория знаний, 2005. — 295 с.
2. Synopsys Documentation: офиц. сайт. — URL: synopsys.com/support/licensing-installation-computeplatforms/synopsys-documentation.html (дата обращения: 01.12.2024).

3. Design Constraints User Guide. Libero SoC v11.8 SP1 and SP2. Microsemi, 2017. — 168с.

4. Гаврилов С. В., Хватов В. М. Преимущества специализированных ресурсов программируемых логических интегральных схем для гибких сложно-функциональных блоков // Инженерный вестник Дона, 2021, №5. URL: ivdon.ru/ru/magazine/archive/n5y2021/6997. — ISSN 2073-8633.

5. Тельпухов Д.В., Рухлов В.С., Иванова Г.А., Рыжова Д.И., Надоленко В.В., Деменева А.И. Исследование вариантов частичного резервирования при проектировании сбоеустойчивых логических блоков ПЛИС // Инженерный вестник Дона, 2018, №1. URL: ivdon.ru/ru/magazine/archive/n1y2018/4681. — ISSN 2073-8633.

6. Хватов В. М. Теоретико-множественное представление сложно-функциональных блоков в системе автоматизированного проектирования интегральных схем на основе реконфигурируемых систем на кристалле // Инженерный вестник Дона, 2023, №8. URL: ivdon.ru/ru/magazine/archive/n8y2023/8616. — ISSN 2073-8633.

7. AMD Vivado Design Suite: офиц. сайт. — URL: amd.com/en/products/software/adaptive-socs-and-fpgas/vivado.html (дата обращения: 01.12.2024).

8. Документация по формату XDC: сайт. — URL: docs.amd.com/r/en-US/ug903-vivado-using-constraints/About-XDC-Constraints (дата обращения: 09.12.2024).

9. Quartus II HandBook Version 10.0 Volume 1: Design and Synthesis. Altera Corporation, 2010. 1821с.

10. Гаврилов С.В., Железников Д.А., Хватов В.М. [и др.]. Программа автоматизации этапов логического синтеза, размещения элементов и трассировки межсоединений микросхем на базе ПЛИС X-CAD. Свидетельство о государственной регистрации программы для ЭВМ №

2024661879.

URL:

fips.ru/ofpstorage/BULLETIN/PrEVM/2024/06/20/INDEX.HTM

11. Гаврилов С.В., Железников Д.А., Хватов В.М. и др. Программа автоматизации этапов логического синтеза, размещения элементов и трассировки межсоединений микросхем на базе ПЛИС X-CAD (для ОС Linux). Свидетельство о государственной регистрации программы для ЭВМ № 2024662730.

URL:

fips.ru/ofpstorage/BULLETIN/PrEVM/2024/06/20/INDEX.HTM

12. Гаврилов С.В., Железников Д.А., Заплетина М.А., Тиунов И.В., Хватов В.М., Чочаев Р.Ж., Шокарев Д.Б. Разработка доверенных средств проектирования ИС в базисе гетерогенных ПЛИС // Труды Института системного программирования РАН — 2023 — Т. 35, № 5 — С. 107-126 — DOI: 10.15514/ISPRAS-2023-35(5)-8.

13. OpenSTA документация — URL: github.com/The-OpenROAD-Project/OpenSTA/blob/master/doc/OpenSTA.pdf (дата обращения: 01.12.2024).

References

1. Kazenov G. G. Osnovy` proektirovaniya integral`ny`x sxem i sistem [Fundamentals of integrated circuits and systems design]. BINOM. Laboratoriya znaniy 2005. 295p.

2. Synopsys Documentation: off. site. URL: synopsys.com/support/licensing-installation-computeplatforms/synopsys-documentation.html (accessed: 01.12.2024).

3. Design Constraints User Guide. Libero SoC v11.8 SP1 and SP2. Microsemi, 2017. 168p.

4. Gavrilov S. V., Khvatov V. M. Inzhenernyj vestnik Dona, 2021, № 5. URL: ivdon.ru/ru/magazine/archive/n5y2021/6997. ISSN 2073-8633.

5. Telpukhov D.V., Rukhlov V.S., Ivanova G.A., Ryzhova D.I., Nadolenko V.V., Demeneva A.I. Inzhenernyj vestnik Dona, 2018, № 1. URL:



ivdon.ru/ru/magazine/archive/n1y2018/4681. ISSN 2073-8633 (online). ISSN 2073-8633.

6. Khvatov V. M. Inzhenernyj vestnik Dona, 2023, № 8. URL: ivdon.ru/ru/magazine/archive/n8y2023/8616. ISSN 2073-8633.

7. AMD Vivado Design Suite: off. site. URL: amd.com/en/products/software/adaptive-socs-and-fpgas/vivado.html (accessed: 01.12.2024).

8. Dokumentaciya po formatu XDC: sayt [XDC format documentation: site]. URL: docs.amd.com/r/en-US/ug903-vivado-using-constraints/About-XDC-Constraints (accessed: 09.12.2024).

9. Quartus II HandBook Version 10.0 Volume 1: Design and Synthesis. Altera Corporation, 2010. 1821p.

10. Gavrilov S.V., Zheleznikov D.A., Khvatov V.M. et al. «Programma avtomatizacii e`tapov logicheskogo sinteza, razmeshheniya e`lementov i trassirovki mezhsjoedinenij mikroshem na baze PLIS X-CAD» [X-CAD - a program for logical synthesis, elements placement, and interconnections routing stages automation for circuits in the FPGA basis]. Svidetel'stvo o gosudarstvennoj registracii programmy dlya EVM № 2024661879. URL: fips.ru/ofpstorage/BULLETIN/PrEVM/2024/06/20/INDEX.HTM

11. Gavrilov S.V., Zheleznikov D.A., Khvatov V.M. et al. «Programma avtomatizacii e`tapov logicheskogo sinteza, razmeshheniya e`lementov i trassirovki mezhsjoedinenij mikroshem na baze PLIS X-CAD (dlya OS Linux)» [X-CAD - a program for logical synthesis, elements placement, and interconnections routing stages automation for circuits in the FPGA basis (for Linux OS)]. Svidetel'stvo o gosudarstvennoj registracii programmy dlya EVM № 2024661879. URL: fips.ru/ofpstorage/BULLETIN/PrEVM/2024/06/20/INDEX.HTM



12. Gavrilov S.V., Zheleznikov D.A., Zapletina M.A., Tiunov I.V., Khvatov V.M., Chochaev R.Z., Shokarev D.B. Trudy Instituta sistemnogo programmirovaniya RAN. 2023. Vol. 35, No. 5. pp. 107-126. DOI: 10.15514/ISPRAS-2023-35(5)-8.

13. OpenSTA dokumentaciya [OpenSTA documentation]. URL: github.com/The-OpenROAD-Project/OpenSTA/blob/master/doc/OpenSTA.pdf (accessed: 01.12.2024).

Дата поступления: 13.12.2024

Дата публикации: 26.01.2025