

# Особенности статического временного анализа и разработки файлов SDC для комплексной системы на кристалле с множественными асинхронными тактовыми доменами

П.Н. Дергачев, И.П. Филимонова, П.А. Шевченко

ЗАО НТЦ «Модуль»,

[petrd@module.ru](mailto:petrd@module.ru), [ipf@module.ru](mailto:ipf@module.ru), [psench@module.ru](mailto:psench@module.ru)

**Аннотация** — Современные микроэлектронные системы это сложные, комплексные устройства, состоящие из большого числа IP блоков различных разработчиков. Основная задача разработчика подобных систем – обеспечение достижения заданных параметров системы. Корректность STA (Static Timing Analysis) определяется корректностью построения стандартного файла временных ограничений (SDC – Synopsys Design Constraints) проекта. В случае разработки комплексного проекта, использующего множество асинхронных доменов тактовых сигналов, разработка и тестирование такого SDC файла – сложная инженерная задача. В докладе приводится метод построения и методика тестирования и отладки SDC файла для СБИС цифровой обработки телевизионных сигналов.

**Ключевые слова** — статический временной анализ (STA), файл временных ограничений (SDC), верификация, система на кристалле, СБИС ЦОТС (СБИС Цифровой Обработки Телевизионного Сигнала), IP блок, маршрут проектирования.

## I. ВВЕДЕНИЕ

Современные микроэлектронные системы это сложные, комплексные устройства, состоящие из большого числа IP блоков различных разработчиков. Основная задача разработчика подобных систем – обеспечение достижения заданных параметров системы [1]. В современных маршрутах проектирования СБИС для определения соответствия системы заданным временным параметрам на этапе проектирования применяются методы статического временного анализа (STA – Static Timing Analysis). Корректность STA определяется корректностью построения стандартного файла временных ограничений (SDC – Synopsys Design Constraints) проекта. В случае разработки комплексного проекта, использующего множество асинхронных доменов тактовых сигналов, разработка и тестирования такого SDC файла – сложная инженерная задача.

## II. ОСОБЕННОСТИ РАЗРАБОТКИ СБИС ЦОТС

Система на кристалле СБИС ЦОТС предназначена для приема и обработки цифровых телевизионных сигналов от различных источников, решения задач обработки аудио и видео потоков, взаимодействия с окружающими устройствами и управления устройствами на основе СБИС ЦОТС.

Особенностями данной системы на кристалле является ее сложность и необходимость поддержки взаимодействия блоков системы, разнородных по составу, исполнению и происхождению. В системе создано несколько доменов тактовых сигналов, асинхронных между собой, см. рис. 1.

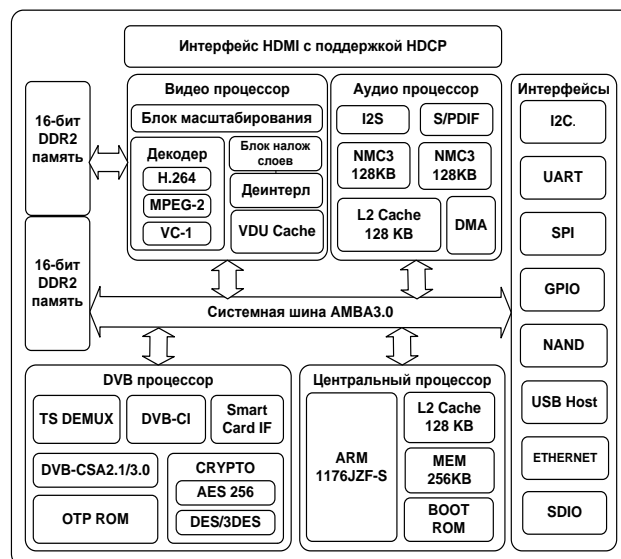


Рис. 1. Структура СБИС ЦОТС

### III. МАРШРУТ ПРОЕКТИРОВАНИЯ СОВРЕМЕННОЙ СИСТЕМЫ НА КРИСТАЛЛЕ

Формирование SDC файла начинается на этапе подготовки логического синтеза системы. В зависимости от выбранного метода проектирования, восходящего или нисходящего, синтез может осуществляться по блоку или для всей системы сразу, но в любом случае, требуется сформировать SDC файл верхнего уровня для всей системы в целом.

Этот файл используется для процесса логического синтеза, физического синтеза и верификации временных характеристик системы от начала логического синтеза до отправки проекта на производство, см. рис. 2.

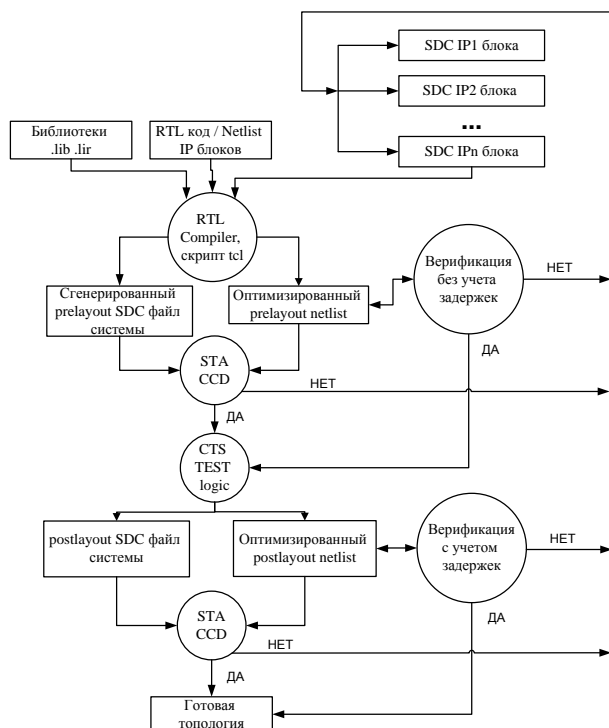


Рис. 2. Маршрут проектирования современной системы на кристалле

Отладка SDC файла включает в себя три этапа.

Во-первых, это формальная проверка. Необходимо проверить синтаксис языковых конструкций, корректность имен идентификаторов и тому подобное.

Во-вторых, это проверка полноты формирования ограничений. Необходимо убедиться, что все тактовые сигналы системы описаны и для всех путей системы определены ограничения.

Данные проверки хорошо формализуются и могут быть выполнены в автоматизированном режиме. Например, командами в САПР Cadence RTL Compiler или при помощи специализированного САПР Cadence CCD.

Третьим этапом отладки является проверка правильности спецификации временных параметров. Данная проверка является наиболее сложной, так как не может быть выполнена формальным образом. Например, если какой-то из реальных путей будет обозначен как несуществующий конструкцией false\_path, формальные проверки будут пройдены, а устройство неработоспособно.

В то же время, обозначение несуществующего пути как синхронного может привести к созданию слишком жесткого временного ограничения, физически нереализуемого в процессе синтеза и разработки топологии СБИС.

Особо следует отметить формирование путей на входы асинхронной установки или сброса регистра, которые нереализуемы без пересинхронизации данных в исходной точке путей.

Основным способом верификации корректности SDC файлов в данном случае будет логическое моделирование системы с учетом задержек элементов. Сложность организации процесса моделирования выражается в следующем: большое число временных путей системы, все должны быть покрыты тестами при моделировании. Разработка подобных тестов – сложная, комплексная задача. В зависимости от реализации логики устройства некоторые пути могут быть недостижимы или труднодостижимы для тестов, например, определенные состояния автомата конечных состояний. В то же время обычно достаточно обеспечить покрытие путей, образующих исключения, в первую очередь, путей между тактовыми доменами системы.

Должны быть полностью сформированы временные параметры системы. В частности, для корректности моделирования с задержками необходимо наличие дерева тактовых сигналов системы. Без дерева тактовых сигналов значительное число путей с исключениями остается непокрытыми тестами.

Все задержки системы должны быть в пределах ограничений, налагаемых SDC файлом. Это касается как задержек на худшие значения параметров (так называемый SETUP timing check), так и параметров при минимальных возможных задержках (так называемый HOLD timing check). Ограничения этого второго типа обычно приводятся в соответствие с SDC файлом в ручном режиме на финальных этапах формирования топологии СБИС.

Из вышеизложенного, в частности, следует, что для проверки корректности SDC файлов необходимо полностью разработать топологию СБИС согласно сформированному SDC файлу. То есть процесс разработки, однозначно, итерационный со значительным временем и усилиями, прикладываемыми на каждой итерации. Формирование, отладка и коррекция ошибок в SDC файле - важный, а во многих случаях и решающий фактор, влияющий на сроки и качество разработки, а,

следовательно, и на коммерческий успех современных многофункциональных систем на кристалле.

Таким образом, метод, упрощающий корректное формирование SDC файла, является одним из ключевых в маршруте разработки СБИС.

#### IV. МЕТОД ПОСТРОЕНИЯ SDC ФАЙЛА

Традиционно используемая модель системы для статического временного анализа [2] определяет в качестве основного элемента анализа путь прохождения сигнала в системе. Рассматриваются пути: от внешнего входа до внутреннего регистра, от внутреннего регистра до внешнего выхода, между внутренними регистрами, от внутреннего регистра до асинхронных входов установки или сброса триггера.

Корректный SDC файл должен однозначно определять параметры всех путей системы.

В условиях формирования системы из готовых блоков составной блок может оказаться закрытым элементом, без доступа к внутренней логике блока. В этом случае пути от регистра закрытого блока и к регистру закрытого блока преобразуются в пути от внешнего регистра к входу закрытого элемента и с выхода закрытого элемента к внешнему регистру с определением задержек на входах и выходах закрытого элемента.

Обычно SDC файл состоит из следующих частей: спецификации тактовых сигналов системы, спецификации внешних задержек для входов и выходов системы и спецификации исключений для временных путей системы.

В качестве типичных исключений определяется взаимодействие между доменами тактовых сигналов системы. В случае если временные пути идут из одного тактового домена в другой, ограничения на такие пути должны определяться явным образом.

#### V. ПРОБЛЕМЫ ПРИ СТАТИЧЕСКОМ ВРЕМЕННОМ АНАЛИЗЕ

При разработке комплексной системы на кристалле зачастую приходится иметь дело с большим количеством асинхронных тактовых доменов в системе. С одной стороны, это обусловлено общей функциональной сложностью системы (например, кроме частот CPU и DDR2 памяти необходимо поддерживать отдельные тактовые домены для блоков обработки видео и звука). С другой стороны, поддержка глобальных тактовых сигналов неэффективна, так как ведет к увеличению задержки на дереве тактовых сигналов (clock tree skew). Это усложняет балансировку деревьев тактовых сигналов (особенно высокочастотных). В реализуемом проекте содержится более 100 доменов тактовых сигналов, при этом встает вопрос, каким образом задавать

временные ограничения на пути между асинхронными тактовыми доменами.

Для асинхронных тактовых доменов типичным решением является объявление путей между асинхронными доменами несуществующими (false path) и исключение из общего анализа [4]. Однако этот путь, допустимый для старых технологий (>90нм), не приемлем для новейших технологий (<65нм) и современных средств проектирования. Требуется, чтобы временные ограничения задавались на все пути, в том числе на пути между асинхронными тактовыми доменами.

В общем случае для контроля временной сходимости синхронных путей используется команда `set_multicycle_path`, которая задает максимальное `Tsu_lim` и минимальное `Thold_lim` значение задержек линии данных параметром `setup` и `hold` [3], см. рис. 3-5. Путь при этом контролируется средствами EDA, но могут возникнуть проблемы со сведением временных параметров для нетлиста со встроенными задержками деревьев тактовых сигналов и тестирующей логики (postlayout нетлист). Очевидно, что сведение временных путей между асинхронными доменами может потребовать дополнительных усилий и временных затрат, при этом может оказаться, что при текущих параметрах команды «`set_multicycle_path`» путь свести невозможно и надо вносить изменения в SDC файл.

Учитывать задержку на дереве асинхронных тактовых сигналов при процедуре STA бессмысленно в связи с тем, что сдвиг фазы для асинхронных тактовых доменов не определен, а для удовлетворения параметра `hold` может потребоваться перебалансировка деревьев асинхронных тактовых сигналов, что, в свою очередь, может ухудшить временную сходимость синхронных путей, см. рис. 6.

Особо стоит отметить тот факт, что SDC файл верхнего уровня должен содержать SDC всех блоков, входящих в систему, многие из которых поставляются сторонними разработчиками. Получается, что для корректного задания временных ограничений между асинхронными доменами инженер должен знать о каждом таком пересечении и умело пользоваться командами «`report_timing`» и «`get_timing_path`». Однако такой подход явно не универсален, так как требует детального изучения и значительно увеличивает время разработки и анализа системы при статическом временном анализе. Итоговый SDC файл системы собирается средствами EDA и может содержать десятки тысяч строк, поэтому эффективная обработка и внесение изменений в такой SDC файл требует автоматизации.

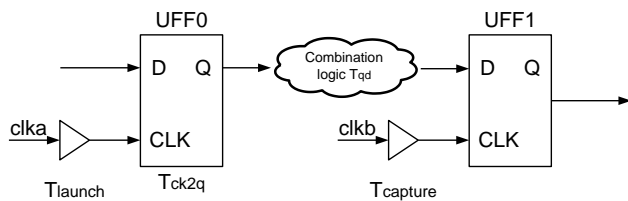


Рис. 3. Пересечение асинхронных доменов тактовых сигналов clka и clkb

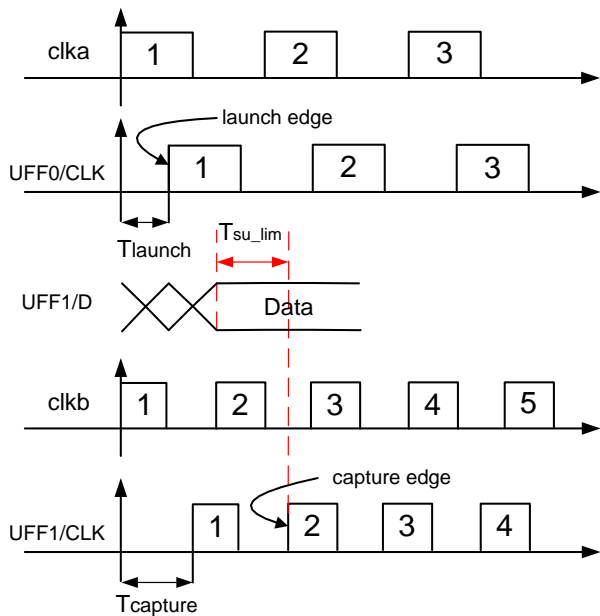


Рис. 4. Временная проверка по максимальному пути (параметр setup)

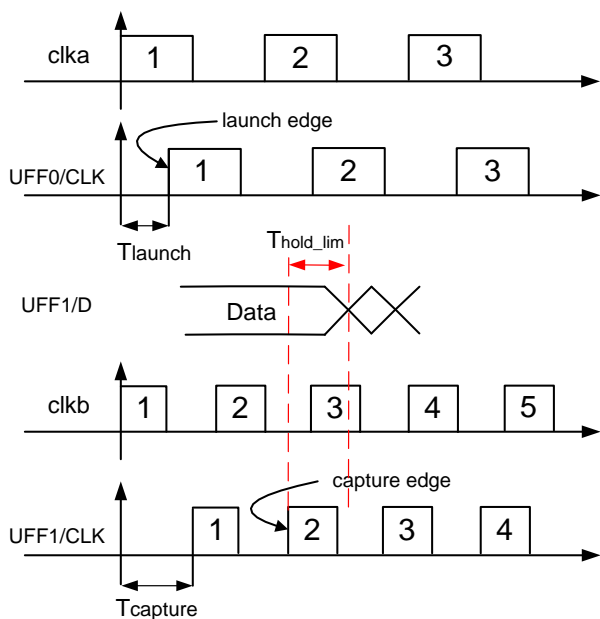


Рис. 5. Временная проверка по максимальному пути (параметр hold)

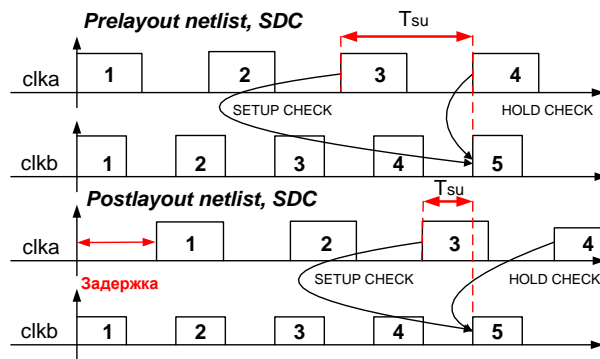


Рис. 6. Временная проверка путей асинхронных доменов по максимальному пути с учетом задержек тактовых сигналов

## VI. МЕТОД ЗАДАНИЯ ВРЕМЕННЫХ ОГРАНИЧЕНИЙ НА АСИНХРОННЫЕ ДОМЕНЫ ТАКТОВЫХ СИГНАЛОВ

### A. Цель метода

Очевидно, возникает необходимость в поиске универсального метода, который удовлетворял бы следующим условиям:

- 1) Позволял бы задавать временные ограничения на пути между асинхронными тактовыми доменами.
- 2) Исключил бы задержку на дереве тактовых сигналов из временного анализа путей между асинхронными доменами.
- 3) Метод должен быть максимально автоматизирован, т.е. по возможности исключать внесение изменений в результирующий SDC вручную.

### B. Основная идея метода

Для решения вышеперечисленных задач нами был разработан метод задания временных ограничений на множественные асинхронные домены, который был применен при разработке системы на кристалле СБИС ЦОТС. Ниже приводится описание этого метода.

Основная идея метода заключается в следующем:

- 1) Для каждого реального тактового сигнала создается дубликат тактового сигнала с постфиксом «cdc», для которого не учитывается задержка на дереве тактовых сигналов, см. рис.7.

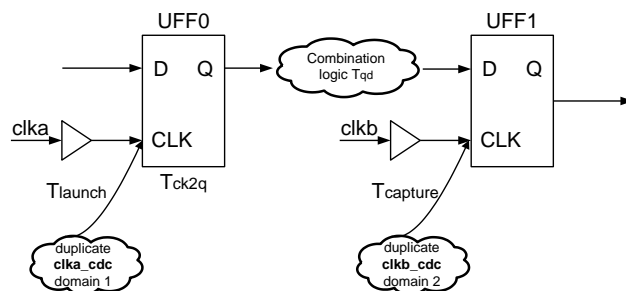
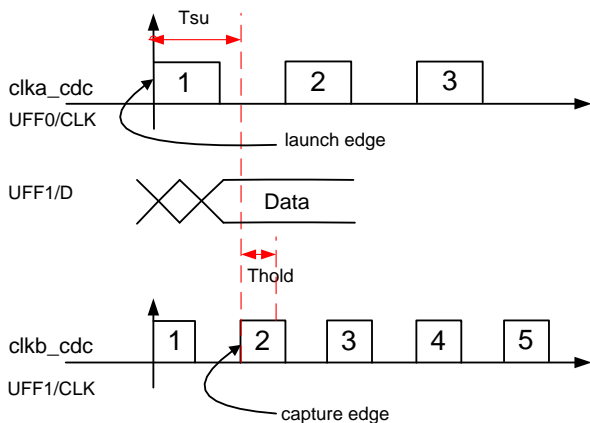


Рис. 7. Пересечение асинхронных доменов дубликатов тактовых сигналов clka\_cdc и clkb\_cdc

- 2) Для задания временных ограничений на пути между асинхронными тактовыми доменами используется команда «set\_multicycle\_path», см. рис. 8.
- 3) Обеспечивается необходимый контроль путей между асинхронными тактовыми доменами исключительно с помощью идеальных тактовых сигналов (с постфиксом «cdc»), а для контроля путей синхронных тактовых доменов используются исключительно реальные тактовые сигналы.

### Prelayout, postlayout netlist u SDC



**Рис. 8. Временная проверка путей асинхронных доменов по максимальному и минимальному пути без учета задержек тактовых сигналов**

### C. Алгоритм реализации

Для реализации этого метода был использован следующий алгоритм и SDC команды:

- 1) Заданы реальные тактовые сигналы при временном статическом анализе (1)-(2):

```
create_clock -name clkb -period 10.0 [get_ports clka], (1)
```

```
create_clock -name clkb -period 5.0 [get_ports clkb], (2)
```

- 2) Определены группы синхронных и асинхронных доменов тактовых сигналов SDC командой (3):

```
set_clock_groups -async -group {clka} -group {clkb}, (3)
```

- 3) Заданы дубликаты тактовых сигналов (4)-(5), задержка деревьев которых отменена SDC командой (6) при временном статическом анализе:

```
create_clock -name clka_cdc -period 10.0 [get_ports clka], (4)
```

```
create_clock -name clkb_cdc -period 5.0 [get_ports clkb], (5)
```

```
remove_propagated_clock [get_clocks *_cdc], (6)
```

- 4) Запрещено анализировать внутренние пути одного асинхронного домена (7):

```
foreach_in_collection cdcclk [get_clocks *_cdc] {set_false_path -from [get_clock $cdcclk] -to [get_clock $cdcclk]}, (7)
```

- 5) Устранено влияние доменов идеальных тактовых сигналов «cdc» на домены реальных тактовых сигналов (8):

```
set_clock_groups -physically exclusive -group [remove_from_collection [get_clocks *] [get_clocks *_cdc]] -group [get_clock *_cdc], (8)
```

- 6) Заданы временные ограничения на пути между асинхронными доменами дубликатов тактовых сигналов (9)-(10):

```
set_multicycle_path -from [get_clock clka_cdc] -to [get_clock clkb] -setup -end 2, (9)
```

```
set_multicycle_path -from [get_clock clka_cdc] -to [get_clock clkb] -hold -end 1. (10)
```

### D. Реализация автоматического написания файлов SDC

Чтобы упростить получение написания SDC файлов, был создан PERL скрипт. Для запуска скрипта необходимо на вход подать SDC файл, полученный на этапе логического синтеза, и таблицу асинхронности доменов тактовых сигналов (11). Главной задачей PERL скрипта является получение двух вспомогательных SDC файлов, впоследствии используемых для статического временного анализа на этапе физического синтеза, см. рис. 9.

```
{clka clkb} {clkc clkd} ... {...}, (11)
```

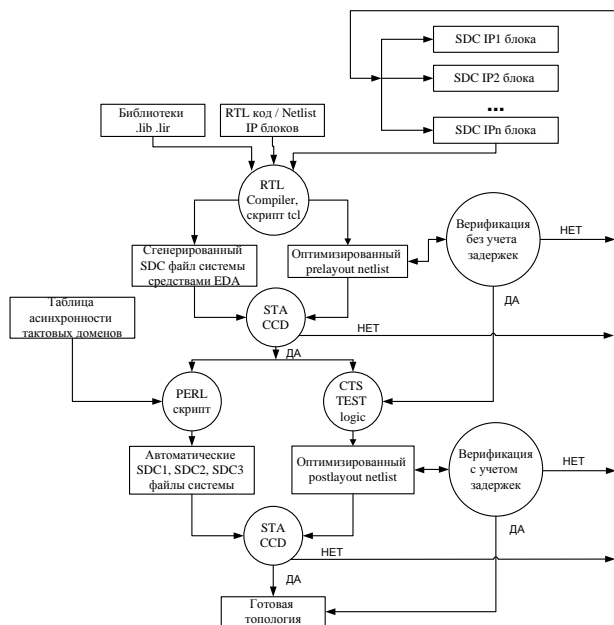
где тактовый сигнал clka (clkc) синхронен тактовому сигналу clkb (clkd), а группа синхронных тактовых сигналов clka и clkb асинхронна группе синхронных тактовых сигналов clkc и clkd.

В итоге мы получаем три SDC:

- 1) SDC1 файл определяет временные ограничения для всех путей синхронных доменов, проверяются пути между реальными тактовыми сигналами. Является файлом задания временных ограничений, созданным на этапе логического синтеза средствами EDA.

- 2) SDC2 файл определяет временные ограничения для всех асинхронных доменов, проверяются пути между идеальными «cdc» тактовыми сигналами. Файл автоматически получается на базе SDC1 файла путем переименования реальных тактовых сигналов.

- 3) SDC3 файл определяет на базе таблицы асинхронности отношения между множеством доменов тактовых сигналов системы, содержит необходимую информацию о дубликатах «cdc» тактовых сигналов.



**Рис. 9. Маршрут проектирования современной системы на кристалле с примененной технологией задания временных ограничений между асинхронными доменами и автоматическим созданием SDC файлов**

## VII. ОСНОВНЫЕ ДОСТИГНУТЫЕ РЕЗУЛЬТАТЫ

Преимуществом данной методики, которая позволяет формировать ограничения временных параметров системы, заданных в SDC файлах, является возможность контролировать пути асинхронных тактовых доменов, исключать из статического временного анализа задержки деревьев тактовых сигналов и получать максимально автоматизированный процесс написания SDC файлов. Нами предложен один из способов задания временных ограничений на пути между асинхронными тактовыми доменами при решении задач проектирования комплексных систем на кристалле, который был эффективно применен в маршруте разработки СБИС ЦОТС.

SDC файл СБИС ЦОТС содержит 119 доменов тактовых сигналов, из которых 44 домена тактовых сигналов являются асинхронными.

Общий объем SDC1 и SDC2 файла – 2,83 Мб.  
Общий объем SDC3 файла – 92,5 Кб.

Разработанный метод формирования SDC файла позволяет сократить время разработки сложных СБИС с множеством асинхронных доменов тактирующих сигналов за счет сокращения итераций внесения изменений в sdc файл и упрощения работы инженеров на стадии back-end. Для проектов, схожих по сложности с СБИС ЦОТС, время разработки топологии при использовании описанного метода может быть сокращено на 3-6 месяцев.

## VIII. ВЫВОД

В докладе описан как традиционный подход к формированию SDC файлов, используемый в проектах по проектированию сложных СБИС с множественными доменами тактовых сигналов, так и метод, разработанный для упрощения задания временных ограничений между асинхронными тактовыми доменами, позволяющий обеспечить разработку топологии СБИС с верными временными ограничениями, снижая риск ошибок временных параметров и значительно уменьшая сроки и стоимость разработки. Данная техника может быть применена в разработке комплексных систем на кристалле.

## ЛИТЕРАТУРА

- [1] Шевченко П.А. СБИС декодера цифрового телевизионного сигнала. Технология разработки // IV Всероссийская научно-техническая конференция «Проблемы разработки перспективных микро- и нанoeлектронных систем – 2010». Сб. трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИПИМ РАН, 2010. С. 320-325.
- [2] Бочаров Ю.И., Гуменюк А.С., Симаков А.Б., Шевченко П.А. Проектирование БИС класса «Система на кристалле»: Учебное пособие. М.: МИФИ, 2008. 188 с.
- [3] J. Bhasker, R. Chadha Static Timing Analysis for Nanometer Designs: A Practical Approach. Springer Science + Business Media, LLC. 2009. 572 p.
- [4] P Zimmer No Man's Land – Asynchronous Clock Domain Crossing // SNUG San Jose 2013. 2013. P. 5-39.