

V. N. Gridin, Professor, Director, ingo@ditc.ras.ru,  
Center for Information Technology in the Design of the RAS,  
V. I. Anisimov, Professor, vianisimov@inbox.ru, G. D. Dmitrevich, Professor,  
A. I. Laristov, Ph. D., Associate Professor, ailaristov@inbox.ru,  
Ya. M. A. Al-Shameri, Graduate Student, saprfkti@mail.ru,  
St. Petersburg Electrotechnical University

## Organization of Information Support of Web-Based Circuit Design CAD

*The questions of the organization of information support of web-based circuit design CAD on the basis of database technology. Analyzes the composition of the circuit design data CAD. It is proposed to form a distributed structure of information support of web-based CAD systems, which is based on the use of a single design space on a web server of Internet resources CAD. User interaction with a web server of Internet resources CAD systems is possible via the built-in browser and server-side web applications to access databases. Discusses the composition of the databases included in a single information space of the web server, and methods for implementation of its components.*

**Keywords:** Web-based circuit design CAD, information CAD software, database technology, Web-server Internet resources CAD

### References

1. **Voikov D.** Tendentsii na rynke IT glazami S APR-kompanii, *PC Week RE*, 2011, no. 16 (766). (in Russian).
2. **Luzin S. Yu., Lyachek Yu. T., Petrosyan G. S., Polubasov O. B.** *Modeli i algoritmy avtomatizirovannogo proektirovaniya radioelektronnoi apparatury*. Saint Petersburg, BKhV-Petersburg, 2010, 224 p. (in Russian).
3. **Anisimov V. I., Gridin V. N.** Metody postroeniya sistem avtomatizirovannogo proektirovaniya na osnove Internet-tekhnologii i kompaktnoi obrabotki razrezhennykh matrits, *Informatsionnye tekhnologii v proektirovanii i proizvodstve*, 2009. no 1. (in Russian).
4. **Tochilkin S.** Perevod graficheskikh prilozhenii v oblaka, *Otkrytye sistemy*, SUBD, 2012, no. 3, pp. 21–27 (in Russian).
5. **Altium Designer Schematic Capture and PCB Editing training**, 2006, Altium Limited, 248 p.
6. **Date C. J.** SQL and Relational Theory: How to Write Accurate SQL Code (2nd edition), Published by O'Reilly Media, Inc., 2012, 446 p.
7. **Laristov A. I., Lyachek Yu. T., Abu Sara M. R.** Integrirovannyye bazy dannykh v programmnykh sistemakh proektirovaniya elektronnykh skhem, *Informatsionno-upravlyayushchie sistemy*, 2009, no. 3 (40), pp. 69–71. (in Russian).
8. **Heineman R.** *PSPICE Einfuhrung in die Elektroniksimulation*. Carl Hanser Verlag, München, 2011, 400 p.
9. **Gridin V. N., Anisimov V. I., Kamen'kov D. V.** Razrabotka tipovogo vstroennogo modulya dokumentooborota v sistemakh upravleniya dannymi ob izdelii. *Informatsionnye tekhnologii v proektirovanii i proizvodstve*, 2011, no. 2, pp. 18–21. (in Russian).
10. **Gridin V. N., Dmitrevich G. D., Anisimov D. A.** Postroenie veb-servisov sistem avtomatizatsii skhemotekhnicheskogo proektirovaniya na osnove metodov szhatiya dannykh. *Informatsionnye tekhnologii i vychislitel'nye sistemy*, 2012, no. 4, pp. 79–84. (in Russian).

УДК 004.3

**А. В. Сурков**, ст. науч. сотрудник, e-mail: surkov@cs.niisi.ras.ru  
Федеральное государственное учреждение науки  
Научно-исследовательский институт системных исследований РАН, г. Москва

## Использование статического временного анализа для проектирования самосинхронных схем

*Рассматривается вопрос адаптации существующего маршрута проектирования синхронных схем для работы с самосинхронными схемами. Современные САПР используют статический временной анализ в качестве промежуточного этапа для большинства операций со схемой. Основным требованием для работы статического временного анализа является отсутствие в схеме обратных связей и асинхронных элементов, что делает невозможным анализ самосинхронных схем. Предложенная методика изменения Liberty-моделей элементов позволяет устранить обратные связи в самосинхронных схемах с волновой синхронизацией, и проводить статический временной анализ. Изменения касаются только описания функций элементов с сохранением значений задержек, полученных при характеристизации. В результате открывается возможность автоматизированного проектирования самосинхронных конвейерных схем с использованием современных САПР.*

**Ключевые слова:** асинхронные, самосинхронные, временной анализ, САПР

## Введение

Существует ряд задач, где применение синхронных схем не всегда удобно или доступно. К примеру, в смешанных аналого-цифровых схемах синхронная логика является источником помех, распространяющихся по линиям питания и земли в аналоговую часть схемы. Существуют применения в условиях нестабильного питания и экстремальных температур, при которых использование синхронных схем ограничено ввиду накладываемых ограничений на время распространения сигнала внутри схемы. Известны также и применения, работающие при напряжении питания около порога срабатывания транзисторов, где синхронная логика не используется ввиду большого разброса задержек переключения. Все перечисленные свойства выгодно отличают *самосинхронный* подход к проектированию: низкие шумы при работе; работа при напряжениях питания около порога срабатывания транзисторов, а также инвариантность к задержкам при любых вариациях питания и температуры, при которых сохраняются переключательные свойства транзисторов. При этом асинхронность может являться и недостатком: самосинхронные схемы работают в логических отсчетах времени, которые не являются постоянными в физическом течении времени [1]. Неспособность работать в физических отсчетах времени в ряде случаев затрудняет взаимодействие самосинхронных схем с элементами синхронной среды. Тем не менее, самосинхронные схемы востребованы, и основным препятствием к их использованию является отсутствие специализированных средств автоматизации проектирования, в результате чего до недавнего времени самосинхронные схемы проектировали вручную.

Исследования В. И. Варшавского [2] показали возможность синтеза самосинхронной схемы путем преобразования синхронной схемы-прототипа. Использование уже разработанных синхронных блоков позволяет значительно сокращать время разработки их самосинхронных аналогов, поскольку экономится время на тестировании и выявлении ошибок. Также, в целях сокращения времени проектирования, в качестве синхронного прототипа удобно брать не исходную *RTL-модель*, а уже синтезированную, с использованием элементов библиотеки, схему. В связи с этим представляется наиболее удобным адаптировать современную методику разработки микросхем для работы с самосинхронной логикой, поскольку это существенно сократит время разработки.

### 1. Обзор маршрутов проектирования микросхем

Все этапы проектирования заказных микросхем (ASIC) условно можно разбить на две части: маршрут аналогового проектирования и маршрут цифрового проектирования. Оба маршрута полностью автоматизированы за счет использования САПР, при этом

для аналогового проектирования используют одни САПР, а для цифрового проектирования — другие.

Процесс автоматизации проектирования включает определенную формализацию входных данных для уменьшения сложности используемых алгоритмов в САПР. В частности, программы для аналогового проектирования работают со *Spice-моделями* транзисторов, емкостей, сопротивлений и индуктивностей, которые предоставляет фабрика. Замена физической топологии моделями позволяет существенно упростить, и как следствие, ускорить расчеты. А программы цифрового маршрута проектирования используют уже модели целых элементов, не опускаясь до уровня их транзисторной схемы. В этом случае, как правило, используют *Liberty-модели* элементов, которые описывают характер и значения задержек между входами и выходами элемента, а также упрощенно описывают функцию выходов. *Liberty-модель* элемента создают с помощью процедуры, называемой *характеризацией* на основе вариативного *Spice-моделирования* транзисторной схемы элемента, с учетом экстрагированных из топологии параметров *spice-моделей* сопротивлений проводов, паразитных емкостей и индуктивностей.

Вариативность моделирования заключается в дискретном переборе таких параметров как питающее напряжение, температура, крутизна переходных характеристик сигналов на входах элемента, и подключенной емкостной нагрузки к его выходам. В результате полученные значения заносятся в таблицы и сохраняются в виде библиотеки в формате *Liberty*. Полученную библиотеку вследствие табличного задания значений также называют точечной (*dot-lib*). В дальнейшем САПР всегда может восстановить поведение элемента и переходную характеристику его выходов, взяв соответствующие заданным параметрам значения из таблиц *Liberty-модели* элемента и проведя требуемую интерполяцию по ближайшим полученным из таблицы точкам.

Существует и более точная модель элементов (*CCS — Composite Current Source*), учитывающая эффект Миллера при переключении входных каскадов элементов, а также более близко описывающая поведение выхода элемента при большом сопротивлении нагрузки. Для расчета задержек с использованием этой модели выход элемента представляется как источник тока, который зависит от крутизны переходной характеристики сигнала на входе, емкости выходной нагрузки, времени, и задается в виде четырех таблиц — при переключении из логического 0 в логическую 1 и обратно, до и после порога переключения.

Емкость входа элемента также задается в виде четырех таблиц и зависит от крутизны подаваемого сигнала, но при расчете задержек может учитывать и емкость нагрузки на выходе элемента. Использование источников тока при описании выхода в модели элемента удобно при анализе шумов и на-

водов (*crosstalk*), а также при анализе просадки питания (*IR-drop*).

Описанная модель с использованием источников тока является значительно более сложной с точки зрения характеристики и использования, и поэтому для работы с самосинхронными элементами на данный момент не рассматривается.

## 2. Синтез самосинхронных схем

Маршрут проектирования цифровой части микросхем условно можно разделить на два этапа: создание RTL-модели схемы и ее синтез с применением библиотек элементов, а затем проектирование топологии схемы с использованием результатов синтеза. RTL-модель синхронной схемы создается путем описания функциональных узлов на языках описания аппаратуры, таких как Verilog или VHDL.

Для функциональной верификации модели используют специальные программы для моделирования, позволяющие провести верификацию с проверкой покрытия тестами. В НИИСИ для моделирования используют САПР Cadence Incisive. Синтез RTL-модели проводится в два этапа: вначале выполняется логический синтез схемы во внутренний формат САПР, а затем с использованием библиотек элементов в формате *Liberty* проводится трансформация синтезированной модели в схему, состоящую из элементов библиотеки и их межсоединений, и называемую *нетлистом*. В НИИСИ для логического синтеза используют программу *Synopsys Design Compiler*. Полученный нетлист в дальнейшем можно моделировать с использованием подключаемых *Verilog*-моделей библиотечных элементов, которые получают на этапе характеристики. При этом существует возможность рассчитать реальные задержки элементов, исходя из нагруженности проводов, путем использования инструментов САПР. Рассчитанные задержки записывают в отдельный файл (формат SDF), который используется при моделировании нетлиста. Следует помнить, что на этапе синтеза САПР не может учитывать все паразитные задержки в схеме, поскольку информация о них появится только после проектирования топологии.

Разрабатываемый в НИИСИ маршрут проектирования самосинхронных схем не предусматривает создания модели самосинхронных схем с использованием языков HDL. Вместо этого используется методика [3] автоматического преобразования нетлиста синхронной схемы, являющейся прототипом. Преобразование проводится с помощью специального скрипта, написанного на языке Perl. Полученный таким образом нетлист самосинхронной схемы также можно моделировать с экстрагированными из библиотеки задержками. Для экстракции задержек, полученных в результате трансформации, нетлист необходимо загрузить в САПР вместе с *Liberty*-библиотеками элементов (включая и самосинхронные элементы), и выполнить экс-

тракцию в формат SDF. Таким образом, САПР синтеза используют для создания самосинхронной схемы лишь косвенным образом: на первом этапе для синтеза прототипа, и на завершающем этапе — для экстракции задержек, используемых при верификации функции самосинхронной схемы с помощью моделирования.

## 3. Статический временной анализ

Программа автоматического синтеза выполняет две основные задачи: цифровой синтез с использованием булевых преобразований, трансформация синтезированной модели в схему из стандартных библиотечных элементов. Эти и другие операции учитывают взятые из *Liberty*-моделей внутренние задержки и емкости выводов элементов, что позволяет проводить оптимизацию схемы по таким параметрам, как площадь, потребление и производительность. Все работающие в этих операциях алгоритмы используют особую методику расчета задержек, называемую статическим временным анализом (*STA — Static Timing Analysis*). Аналогичная методика применяется и в САПР по проектированию топологии, поскольку позволяет оценивать правильность размещения элементов, расположения соединений с использованием металлов и т. д. В первую очередь STA является методикой оценки временных характеристик схемы, а поскольку все используемые в САПР алгоритмы обязаны вести контроль над результатами проводимых операций, то без STA полноценная работа САПР невозможна.

В статическом временном анализе используется графовая модель соединений между элементами: элементы являются вершинами графа, а связи — соединениями. При этом используется такой термин, как путь (*path*) сигнала, представляющий собой траекторию распространения сигнала, проходящую по соединениям (ребрам графа) через элементы логики (вершины графа). Пути сигналов можно разделить на две группы — синхронные и асинхронные.

Синхронные пути сигналов всегда конечны: они начинаются на тактовом входе синхронного триггера-источника сигнала, проходят внутри модели триггера на выход, затем распространяются по проводам через модели элементов логики и заканчиваются на входе данных синхронного триггера-приемника сигнала. Элементы логики, включенные между триггерами, являются частью пути сигнала вместе с соединяющими их проводами. Пример пути сигнала в синхронной схеме показан пунктиром на рис. 1.

Для работы САПР используют в основном синхронные пути сигналов, поскольку они наиболее удобны для расчетов ввиду своей конечности. Однако в некоторых случаях пути сигналов могут замыкаться в петлю, что нарушает работу САПР. Например, асинхронные входы синхронных элементов, такие как асинхронный сброс у триггера, также мо-

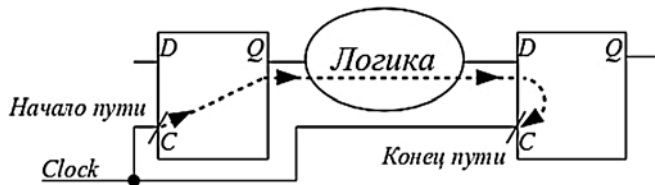


Рис. 1. Путь распространения сигнала в синхронной схеме

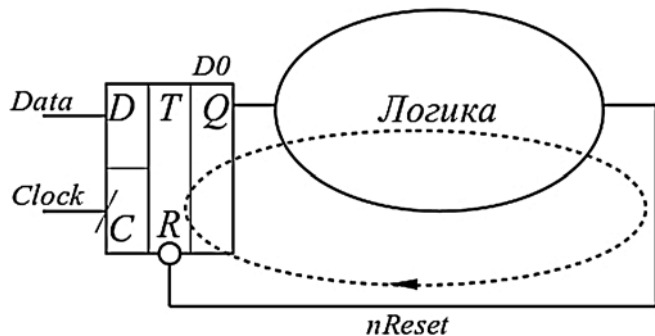


Рис. 2. Пример петли обратной связи в синхронной схеме

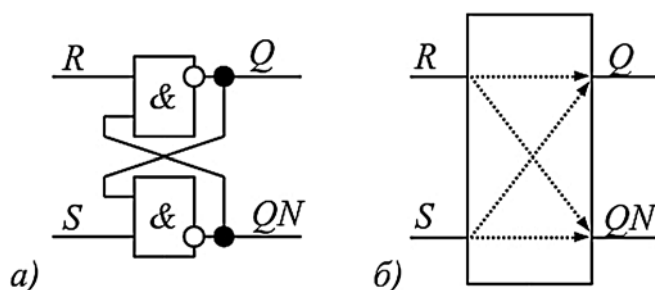


Рис. 3. Принципиальная схема RS-зашелки (а), и асинхронные связи между входами и выходами внутри Liberty-модели RS-зашелки (б)

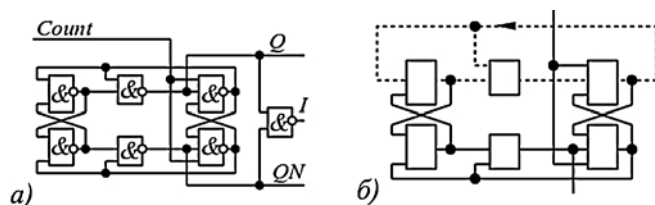


Рис. 4. Схема самосинхронного счетного триггера (а), и пример обратной связи внутри самосинхронной схемы (б)

гут стать частью пути сигнала и быть использованы при расчете. В результате, траектория сигнала замыкается через обратную связь, образовав петлю. Схема с приведенной петлей обратной связи показана на рис. 2.

Схема соединений на рис. 2, считается непригодной к использованию. При обнаружении замкнутого пути сигнала, обозначенного на рис. 2 пунктиром, САПР попытается искусственно разорвать эту связь, удалив информацию о соединении в некоторой точке петли. Что, в свою очередь, приводит к ошибке при расчете задержки прохождения сигнала по указанному пути.

Тем не менее, существуют и асинхронные пути сигналов, для активации расчета которых в САПР

необходимо использовать специальные ключи. Асинхронные пути сигналов проходят только через входы полностью асинхронных элементов, таких как RS-зашелка, показанная на рис. 3.

Асинхронные пути сигналов также могут быть использованы в расчетах, но только в том случае, если они не замыкаются через обратную связь. Все известные САПР цифрового маршрута проектирования работают исключительно с конечными путями сигналов — как синхронными, так и асинхронными. Но поскольку в подавляющем большинстве проектируемых схем асинхронных элементов нет, то по умолчанию при расчетах используют только синхронные пути сигналов.

Невозможность работать с замкнутыми путями сигналов является основной преградой к использованию современного цифрового маршрута для проектирования самосинхронных схем. В качестве примера рассмотрим схему самосинхронного счетного триггера (рис. 4) и путей сигналов внутри.

На рис. 4, а показана принципиальная схема самосинхронного счетного триггера со счетным входом (сигнал *Count*), парафазным выходом (сигналы *Q* и *QN*), и выходом индикации окончания переходных процессов (сигнал *I*). На рис. 4, б в качестве примера замкнутой обратной связи пунктиром обозначена траектория сигнала через три элемента И-НЕ; функции элементов скрыты для наглядности рисунка. Ввиду множества обратных связей в самосинхронных схемах алгоритмы STA не работают, а значит, и не работают такие функции САПР, как размещение элементов, трассировка, различные оптимизации скорости, площади схемы, потребления питания и т. д. Поэтому для использования САПР при проектировании самосинхронных схем, необходимо избавиться от обратных связей, препятствующих работе статического временного анализа.

В ходе исследований было найдено единственное на данный момент решение задачи о ликвидации обратных связей в схеме, заключающееся в изменении описания Liberty-моделей. Асинхронные элементы с помощью моделей удобно представлять как элементы логики, либо как синхронные триггеры, в результате чего асинхронные пути сигналов становятся конечными и могут быть описаны как синхронные. В результате изменения моделей элементов САПР может работать с самосинхронной схемой так, как если бы она являлась синхронной. С помощью САПР становится возможным автоматически размещать элементы, проводить трассировку и оптимизации.

#### 4. Проектирование и характеристика самосинхронных элементов

На данный момент для проектирования самосинхронных конвейеров потребовалось создать всего два типа самосинхронных элементов, отсутствующих в стандартной библиотеке (TSMC 65 нм). Первый тип элементов — С-элемент Маллера спро-

ектирован с использованием статической схемы [4] для двух входов (рис. 5) и с использованием мостиковой схемы [4] для трех входов (рис. 6).

Также для схемы синхростратума для волновой обработки данных [1] потребовалось спроектировать двухвходовой *C*-элемент со сбросом, который используется для начальной установки схемы. Второй тип элементов — это RS-зашелка с дополнительным сбросом, используемая для хранения данных. RS-зашелку можно построить и из двух библиотечных элементов И-НЕ (см. рис. 3, а), однако в этом случае мы получаем две замкнутые обратные связи. Поэтому оказалось целесообразным спроектировать RS-зашелку в виде отдельного элемента библиотеки: в этом случае обратные связи оказываются скрытыми внутри *Liberty*-модели.

Топологическое проектирование выполняла САПР Virtuoso; DRC и LVS проведены с помощью САПР Calibre. Проектирование топологии самосинхронных элементов ничем не отличается от проектирования топологии других устройств.

Для характеристики элемента в САПР SiliconSmart требуется *Spice-нетлист*, который представляет собой совокупность *spice*-моделей транзисторов, паразитных емкостей, индуктивностей и сопротивления проводов, а также всех межсоединений. *Spice-нетлист* получается на завершающем этапе разработки топологии элемента с использованием САПР Calibre. Также для характеристики требуется описать функцию устройства, чтобы САПР мог составить список тестов для измерения электрических и временных характеристик элемента, которые впоследствии будут сохранены в табличном виде в описании *Liberty*-модели элемента. Поскольку *C*-элемент Маллера не является распространенным устройством, то его функциональное описание удобно представить для характеристики как RS-зашелку (см. рис. 3). В качестве примера рассмотрим двухвходовой *C*-элемент на рис. 4, поведение выходов которого описывается следующими формулами:

$$R = !A \& !B; \quad (1)$$

$$S = A \& B, \quad (2)$$

где *A* и *B* — входы *C*-элемента, а выход *C*-элемента берется с прямого выхода (*Q*) RS-зашелки.

Полученная в результате характеристики *Liberty*-модель *C*-элемента подходит для экстракции задержек в САПР Synopsys DC, а полученная Verilog-модель может быть использована вместе с экстрагированными задержками в моделировании нетлиста. Однако для статического временного анализа полученная *Liberty*-модель не годится, поскольку содержит только асинхронные пути сигнала, изображенные на рис. 3, б. Чтобы САПР мог считать задержку прохождения сигнала через *C*-элемент, необходимо вручную исправить *Liberty*-модель, изменив функцию устройства с зашелки на элемент логики, такой как И или ИЛИ. В этом случае пути сигнала внутри модели можно считать синхронными

и использовать при расчете задержек в САПР. Изменение функции элемента можно считать корректным, поскольку применяемые сейчас в большинстве САПР алгоритмы STA не используют при расчетах функцию элемента, беря вместо этого наихудшую из задержек сигнала (вход-выход), содержащихся в *Liberty*-модели элемента. Измененную *Liberty*-модель можно использовать только в САПР синтеза и топологии. Для функционального моделирования необходимо использовать оригинальные модели элементов, полученные на этапе характеризации.

Более сложной задачей является изменение *Liberty*-модели RS-зашелки, применяемой для хранения данных на выходах стадий самосинхронного конвейера. Стадия самосинхронного конвейера условно показана на рис. 7.

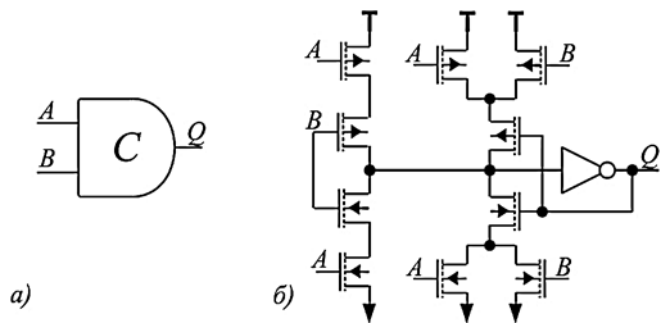


Рис. 5. Статическая схема двухвходового *C*-элемента Маллера

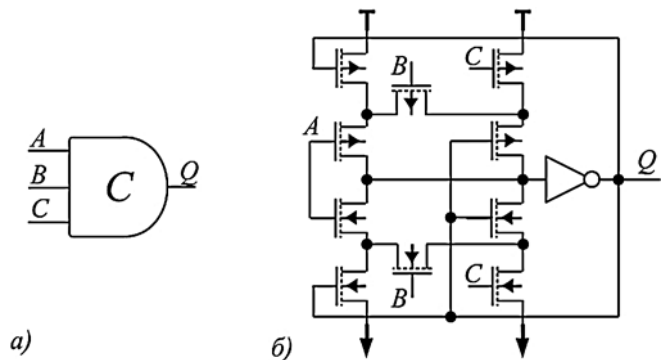


Рис. 6. Мостиковая схема трехвходового *C*-элемента Маллера

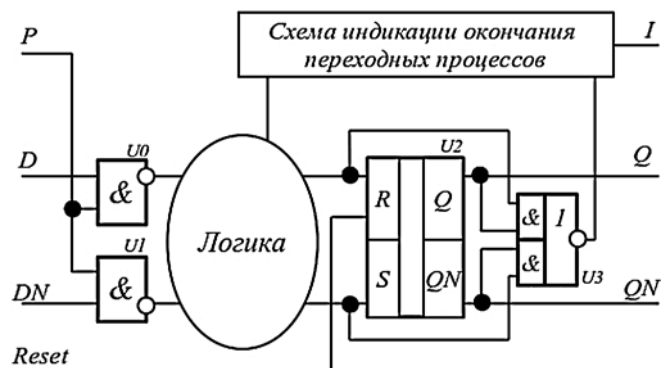


Рис. 7. Стадия самосинхронного конвейера

В целях упрощения рисунка у стадии конвейера изображены только один входной ( $D$ ,  $DN$ ) и один выходной ( $Q$ ,  $QN$ ) парафазные разряды данных, но подразумевается, что схема содержит множество входных и выходных разрядов данных. Входной сигнал  $P$  является фазовым для всей стадии, и управляет элементами блокировки входных сигналов  $U0$  и  $U1$ . Элемент  $U3$  осуществляет индикацию окончания переходных процессов в защелке  $U2$ , формирующей выходы одного разряда данных. Сигнал  $I$  является выходом схемы окончания переходных процессов всей стадии конвейера, включая логику и защелки хранения. Элементы блокировки входных сигналов являются логическими элементами и могут быть использованы при расчете путей сигнала. В то же время защелка  $U2$  является непрозрачной (содержит только асинхронные пути сигнала) для СТА, и ее *Liberty*-модель должна быть изменена. Поскольку в синхронных схемах начало и конец пути сигнала могут находиться только внутри модели  $D$ -триггера, то целесообразно изменить и *Liberty*-модель защелки наподобие синхронного триггера. Пути сигналов в *Liberty*-модели защелки после характеристики и после изменения показаны на рис. 8. Для проектирования самосинхронных конвейеров удобно использовать защелки с двумя сбросовыми входами.

На рис. 8, а, в отличие от рис. 3, б, показаны все существующие пути сигналов внутри модели, включая сквозные пути сигнала, и связь между входами. Сквозные пути учитываются при расчете задержки распространения сигнала, а пути между входами используются лишь для контроля взаимовлияния (*Setup* и *Hold*) входных сигналов во время их установки и снятия.

Модель защелки необходимо изменить таким образом, чтобы асинхронные пути стали синхронными. На рис. 8, б показано, как внутри *Liberty*-модели помещаются два виртуальных синхронных  $D$ -триггера, которые тактируются с входа  $R1$ , на который согласно схеме стадии конвейера (см. рис. 7) заведен сигнал сброса. Сигнал сброса является глобальным для всего конвейера, поэтому с точки зрения САПР может быть объявлен как тактирующий. В результате проведенного изменения *Liberty*-модели схему самосинхронного конвейера на рис. 7 можно рассматривать как синхронную, по-

скольку при расчете задержек сквозные пути со входов ( $D$ ,  $DN$ ) до выходов ( $Q$ ,  $QN$ ) отсутствуют, а в схеме появляются виртуальные синхронные триггеры и схема тактирования.

Прделанные изменения *Liberty*-модели не являются полностью корректными с точки зрения расчета временных параметров схемы, поскольку лишь незначительная часть асинхронных путей сигналов, получившихся на этапе характеристики, наследуется в измененной модели. В частности, задержка между входом  $R0$  и  $Q$  значительно меньше, чем между  $S$  и  $Q$ , что следует из принципиальной схемы защелки (см. рис. 3, а). В связи с этим задержку до выхода  $Q$  в измененной модели целесообразно взять равной асинхронной задержке между  $S$  и  $Q$ , поскольку она заведомо больше задержки между  $R0$  и  $Q$ . В результате, алгоритмы статического временного анализа схемы с измененными элементами будут оперировать реальными значениями емкостей выводов элементов, но внутренняя задержка этих элементов будет взята с запасом, что существенно ограничивает вариативность анализа.

Представление *Liberty*-моделей *RS*-защелок в качестве сдвоенного синхронного  $D$ -триггера, у которого к тактовому входу подключен сигнал сброса, позволяет представить схему самосинхронного конвейера (см. рис. 7) в качестве синхронной схемы, состоящей их элементов логики,  $D$ -триггеров, и у которой линия сброса может рассматриваться как псевдотактирующий сигнал. Если после указанной замены в схеме остались обратные связи, то достаточно выбрать удобный элемент для разрыва обратной связи, и изменить его модель на синхронный  $D$ -триггер. Для расчета скорости работы стадии самосинхронного конвейера с использованием СТА, необходимо выполнить расчет наихудшего пути сигнала, проходящего через фазовый вход и через выход окончания переходных процессов рассматриваемой стадии конвейера.

Таким образом, изменение *Liberty*-моделей элементов решает основную задачу использования САПР для проектирования самосинхронных схем, но расчет задержек возможен только для наихудшего случая, поскольку при изменении *Liberty*-модели асинхронного элемента часть задержек не учитывается.

## 5. Практические результаты

Маршрут проектирования самосинхронных схем использован при разработке самосинхронного блока умножения с накоплением (FMA) двойной точности с применением технологии TSMC 65 нм. Спроектирована библиотека асинхронных элементов, содержащая *RS*-защелку и *C*-элементы Маллера. В качестве прототипа взята синхронная RTL модель блока FMA, использующаяся в процессоре КОМДИВ-64 [5]. На первом этапе разработки осуществлен синтез синхронного прототипа FMA в базис стандартных библиотечных элементов. Далее

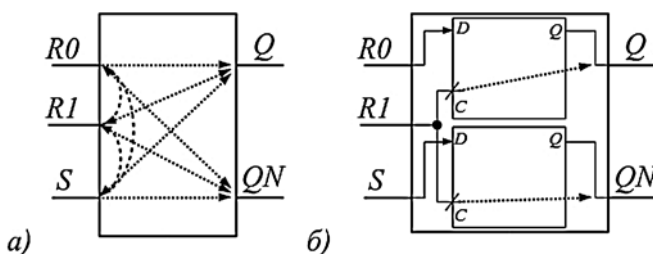


Рис. 8. Пути сигналов в *RS*-защелке после характеристики (а), и после изменения *Liberty*-модели (б)

выполнено перекрестное преобразование с помощью скрипта в прямой и инверсный каналы передачи данных с восстановлением монотонности полученных в результате преобразования функций. И в заключение, синхронные триггеры заменены на RS-защелки, построена схема индикации завершения переходных процессов в стадиях конвейера, добавлены элементы синхростратума для волновой обработки данных и выписан результирующий нетлист. Для дальнейшей работы потребовалось изменить библиотеки асинхронных элементов в соответствии с приведенной методикой. В результате получены два набора библиотек: библиотека с использованием оригинальных *Liberty*-моделей элементов, используемая для выписывания файла экстрагированных задержек элементов (SDF), и библиотека для использования алгоритмов STA. Проведено моделирование полученной самосинхронной схемы с экстрагированными задержками в САПР Incisive, а также сделан статический временной анализ наихудших путей распространения сигнала в САПР Design Compiler. С учетом внесенных изменений в модели элементов, ухудшающих точность расчетов, результаты моделирования и временного анализа совпали.

### Заключение

Использование алгоритмов STA является ключевым условием для работы современных САПР. При этом STA работает только с конечными путями распространения сигналов в схеме и на практике используется в основном с синхронными моделями элементов. Самосинхронные же схемы состоят из элементов логики и самосинхронных элементов

памяти, поэтому большая часть путей сигналов внутри схемы замкнута. Замкнутые пути сигналов, в свою очередь, препятствуют автоматизации проектирования с использованием САПР.

В НИИСИ разработана методика изменения асинхронных моделей элементов таким образом, чтобы самосинхронные конвейерные схемы становились неотличимы от синхронных схем с точки зрения STA и проектирования в САПР. Новый маршрут также включает в себя использование уже разработанных ранее синхронных блоков. Таким образом, новая методика позволяет не только сократить время проектирования самосинхронных конвейеров за счет использования САПР, но и экономит время на разработку логической модели и тестирование за счет применения имеющихся синхронных разработок.

*Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 14-29-09238 офи\_м.*

### Список литературы

1. **Мараховский В. Б., Мелехин В. Ф.** Проектирование средств синхронизации блоков глобально асинхронных систем с произвольной локальной синхронизацией // Информационно-управляющие системы. 2010. № 1. С. 29–38.
2. **Varshavsky V. I., Marakhovsky V. B., Chu T. A.** Asynchronous Timing of Arrays with Synchronous Prototype // Proc. of the Second Intern. Conf. on Massively Parallel Computing Systems (MPCS'96). 1996. P. 47–54.
3. **Сурков А. В.** Использование Synopsys Design Compiler для синтеза самосинхронных схем // Программные продукты и системы. 2014. № 4. С. 24–30.
4. **Moreira M., Oliveira B., Moraes F., Calazans N.** Impact of C-Elements in Asynchronous Circuits // 13<sup>th</sup> Int'l Symposium on Quality Electronic Design. 2012. P. 437–452.
5. **Бобков С. Г.** Архитектура микропроцессоров суперЭВМ эскафлопсного диапазона // Информационные технологии. 2012. № 12. С. 2–9.

**A. V. Surkov**, Senior Researcher, surkov@cs.niisi.ras.ru

Scientific Research Institute of System Analysis, Russian Academy of Sciences, Moscow, Russia

## STA Algorithms for Developing Self-Timed Schemes

*The article covers the problem of adopting the synchronous workflow for developing self-timed schemes. The modern EDA tools use the static timing analysis algorithms (STA) as interim step for the most of operations. The major requirement for the STA to work is the absence of feedback interconnections in the analyzed circuit. Suggested approach eliminates the feedbacks in the self-timed pipelines by the modifying the Liberty-model of used elements. The modification affects the functional description of modified elements, so it does not change the arc delays. This allows STA to work, so it becomes possible to use the modern EDA tools to automate design of the self-timed pipelines.*

**Keywords:** asynchronous, self-timed, STA, EDA tools

### References

1. **Marakhovsky V. B., Melekhin V. F.** Proektirovanie sredstv synchronizatsii blockov globalno asinchronnih sistem s proizvolnoy localnoy synchronizatsiyey, *Information and Control Systems*, 2010, no. 1, pp. 29–38 (in Russian).
2. **Varshavsky V. I., Marakhovsky V. B., Chu T. A.** Asynchronous Timing of Arrays with Synchronous Prototype, *Proc. of the Second Intern. Conf. on Massively Parallel Computing Systems (MPCS'96), Ischia, Italy*, 1996, pp. 47–54.
3. **Surkov A. V.** Synthesis of burst-mode asynchronous schemes using Synopsys Design Compiler, *Programmnye produkty i sistemy*, 2014, no. 4, pp. 24–30 (in Russian).
4. **Moreira M., Oliveira B., Moraes F., Calazans N.** Impact of C-Elements in Asynchronous Circuits, *13<sup>th</sup> Int'l Symposium on Quality Electronic Design*, 2012, pp. 437–452.
5. **Bobkov S. G.** Arhitektura mikroprocessorov super-EVM eskaflopsnogo diapazona, *Informacionnye Tekhnologii*, Moscow, 2012, no. 12, pp. 2–9 (in Russian).