

Реализация синтеза самосинхронных схем в базисе БМК

Плеханов Леонид Петрович¹⁾, Денисов Андрей Николаевич²⁾, Дьяченко Юрий Георгиевич¹⁾, Мамонов Дмитрий Игоревич²⁾, Морозов Николай Викторович¹⁾, Степченко Дмитрий Юрьевич¹⁾

¹⁾Институт проблем информатики Федерального Исследовательского Центра "Информатика и управление" Российской академии наук, 119333, г. Москва, ул. Вавилова, д. 44, корп. 2, <http://www.ipiran.ru>, E-mail: lplekhanov@inbox.ru, +79162774834

²⁾Научно-производственный комплекс "Технологический центр" (НПК "Технологический центр"), 244980, Москва, Зеленоград, проезд 4806, д.5, <http://www.asic.ru>, E-mail: den@tcen.ru, +74997208992

Аннотация

Доклад посвящен реализации автоматизированного синтеза самосинхронных (СС) схем на основе описания схемы на языке Verilog и библиотеки стандартных элементов 5503, расширенной специфичными для СС-схем элементами 5503СС. Программа синтеза имеет оконный интерфейс и интегрируется в САПР "Ковчег" СБИС на базовых матричных кристаллах. Приводится сравнение результатов автоматического и ручного синтеза.

Ключевые слова: самосинхронные схемы; автоматизированный синтез; синтез самосинхронных схем; маршрут проектирования.

1 Введение

В предыдущей работе [1] были приведены основные принципы и подходы в организации синтеза самосинхронных (СС) схем в рамках САПР цифровых устройств. В продолжение темы, данный доклад конкретизирует способы реализации такого синтеза.

Самосинхронные схемы обладают рядом уникальных свойств, недостижимых при синхронной и других асинхронных реализациях: отсутствием состояний при любых задержках элементов, предельно широкими областями работоспособности по температуре и напряжению питания, полной остановкой при возникновении константных неисправностей типа залипания и некоторыми другими. Эти свойства обоснованы теоретически и подтверждены экспериментально [2– 4]. Такие свойства СС-схем обеспечиваются их специальным построением и режимом функционирования.

Функционирование СС-схем осуществляется чередованием двух фаз: рабочей и вспомогательной (спейсерной) фазы. В рабочей фазе делаются необходимые логические преобразования сигналов, в спейсерной фазе выполняются вспомогательные действия. В обеих фазах подтверждается окончание всех переходных процессов и формируется единый подтверждающий сигнал. Каждая очередная фаза начинается только после подтверждения по обратной связи окончания переходных процессов в предыдущей фазе. Процесс формирования сигнала окончания переходных процессов далее будем называть *индикацией*.

Для организации двухфазного режима и различения процессов в фазах используется построение логики с помощью СС-кодов [2], в данном случае парафазного кода (ПФ-кода). В ПФ-коде каждый исходный информационный сигнал представляется двумя сопряженными сигналами. В рабочей фазе каждая такая пара сигналов имеет взаимно-обратные значения: 01 представляет информационную единицу, 10 – информационный ноль. В спейсерной фазе (спейсере) оба сигнала одинаковы и равны 00 или 11 в зависимости от места в схеме.

Обобщенно процедура предлагаемого синтеза состоит в преобразовании Verilog-описания схемы с верхних уровней в систему логических функций, переводе функций в ПФ-код, генерации регулярных конструкций (счетчиков, регистров и т.п.), подборе библиотечных элементов для их реализации и организации индикации всей схемы.

2 Постановка задачи

Синтез проводится на элементах заранее подготовленной библиотеки (*базовой библиотеки*).

Для синтеза предъявляются:

- исходное описание схемы на языке Verilog любого уровня – функций, регистровых передач, верхнего,
- параметры синтеза (задание на синтез).

Задача состоит в создании СС-схемы на элементах базовой библиотеки с критерием минимизации сложности [1] и учетом нагрузочной способности элементов.

Сложность элементов, нагрузки входов (входные емкости) и нагрузочные способности выходов задаются в базовой библиотеке. Нагрузка на выходы элементов синтезируемой СС-схемы вычисляется как сумма емкостей присоединенных входов и паразитной емкости трасс. На предварительном этапе, при отсутствии данных из топологии, трассы учитываются усредненно: подсчитывается число «веток» трассы и умножается на средний коэффициент, заданный в библиотеке. "Ветками" считаются выход элемента и присоединенные к нему входы.

2.1 Исходное описание схемы

Исходное описание схемы на языке Verilog допускает иерархию, наличие переменных, векторов и регистров. Функционирование схемы описывается на любом уровне. Например, для описания суммирования достаточно использовать конструкцию:

$$\text{assign Sum} = A + B;$$

где A и B – векторы любой заданной длины.

Если в исходной схеме используются готовые СС-модули, для корректного встраивания в синтезируемую схему их интерфейсы должны быть описаны в специальном разделе библиотеки – макроэлементах.

2.2 Параметры синтеза

В отдельном файле – задании на синтез – приводится информация, определяющая СС-атрибуты входов и выходов результирующей схемы и другие необходимые для синтеза данные. Пример задания для синтеза умножителя 4×4 показан на рис. 1.

```
//=====
//                               Task for synthesis
//=====

Sour_file:  Mult_4x4.v; // source file
Module:    mult_buta;  // module to synthesize
IN_Space_0: * ;      // names or * (all)
IN_Space_1: ;        // --"--
IN_Indics: ;         // --"-- to be indicate on total OUT unar-phase
OUT_Indics: * ;      // --"-- --"--
SS_Amplify: * ;     // amplify ST-circuit's signals
OutLang:   Verilog;  // Verilog, VHDL, Form (FormatKey)
```

Рис. 1 Пример задания для синтеза умножителя 4×4

Обязательно задание спейсерных значений всех входов схемы (параметры **IN_Space_0** и **IN_Space_1** на рис. 1). Выходные спейсеры (параметры **OUT_Space_0** и **OUT_Space_1**) могут быть заданы, и они будут обеспечены при синтезе (в данном примере на рис. 1 они отсутствуют). Незаданные выходные спейсеры определяются при синтезе автоматически, исходя из критерия минимальной сложности.

Опционально задаются требования индикации внешних входов и выходов схемы на общем индикаторном СС-выходе схемы (параметры **IN_Indics** и **OUT_Indics** на рис. 1). (По условиям самосинхронности все внутренние сигналы СС-схемы будут индцированы на этом выходе автоматически).

Параметр усиления (**SS_Amplify**) дает гибкую возможность согласовать реальные нагрузки с нагрузочными способностями элементов СС-схемы:

- усилить все сигналы, формируемые элементами, нагрузочная способность которых меньше их нагрузки в схеме,

- не усиливать никакие сигналы,

- усилить указанные пользователем сигналы СС-схемы.

Параметр "**OutLang**" задает язык описания синтезированной схемы.

Регистры, счетчики и другие подобные устройства регулярной структуры в СС-схемах строятся специальным способом, поэтому в исходных данных необходимо задавать их параметры: разрядность, тип и другие. В процессе синтеза такие устройства будут сгенерированы автоматически.

3 Процедура синтеза

Синтез проводится в несколько этапов.

Этап 1. Предварительная обработка.

На первом этапе используется программа с открытым кодом Yosys [5]. Она переводит Verilog-описание схемы любого уровня в систему логических функций с числом аргументов не более трех. Такое преобразование весьма удобно для синтеза, так как позволяет избежать операций декомпозиции функций и эффективно использовать элементы базовой библиотеки.

Этап 2. Генерация заданных устройств и встраивание макроэлементов (внешних готовых СС-схем).

По описанию макроэлементов в исходной схеме и указанию в задании будут сгенерированы устройства необходимой разрядности и заданными СС-атрибутами из элементов базовой библиотеки. На текущий момент генерируются наиболее востребованные устройства: регистры хранения, регистры сдвига и счетчики. В дальнейшем список подобных устройств может быть расширен.

Внешние СС-схемы встраиваются в соответствии с их интерфейсами, находящимися в разделе макроэлементов базовой библиотеки. Интерфейсы должны содержать описания портов и их СС-атрибуты.

Этап 3. Выделение и балансировка комбинационных фрагментов.

На этом шаге выделяются комбинационные части схемы и определяются их СС-атрибуты по атрибутам входов-выходов схемы из задания и атрибутам генерируемых и встраиваемых устройств.

В каждом фрагменте путем подстановок исключаются константы и инверторы и удаляются промежуточные функции-константы. Далее проводится укрупнение функций путем подстановок таким образом, чтобы после перевода их в ПФ-код в базовой библиотеке нашлись элементы для реализации обоих компонентов ПФ-кода. Укрупнение позволяет снизить общую сложность схемы. Элементы выбираются с учетом критерия минимальной сложности.

Этот способ гарантирует покрытие получаемых функций элементами базовой библиотеки.

Этап 4. Синтез комбинационных фрагментов схемы.

На данном шаге исходные функции переводятся в ПФ-код и подбираются покрывающие их библиотечные элементы.

Выполняется согласование спейсеров: на входы каждой пары ПФ-функций должны подаваться сигналы с одинаковым спейсером. При необходимости добавляются согласующие инверторы.

Этап 5. Синтез индикаторной подсхемы.

Индикация связана с необходимостью использования в СС-схемах особых сигналов, показывающих текущую фазу работы схемы – унарных фазовых (УФ) сигналов. Такие сигналы в процедуре синтеза появляются при генерации СС-устройств и встраивании внешних СС-схем. Кроме того, УФ-сигналы появляются и при построении самой индикации.

Индикаторная подсхема решает две задачи: получает от каждой пары ПФ-функций (элементов) один УФ-сигнал и объединяет все частичные УФ-сигналы в один общий индикаторный сигнал схемы.

УФ-сигнал ПФ-пары получается подключением на выходы пары 2-входового элемента: ИЛИ (или ИЛИ-НЕ) для пары со спейсером 00 и И (или И-НЕ) для пары со спейсером 11. Для объединения УФ-сигналов в один общий индикаторный выход применяются гистерезисные триггеры (Г-триггеры) [2]. Существует несколько разновидностей Г-триггеров:

- только с ПФ- входами (2 пары входов),
- только с УФ- входами (от 2 до 4 входов),
- с ПФ и УФ входами с общим числом входов до четырех.

На выходе Г-триггера формируется один УФ-сигнал.

Для улучшения быстродействия и уменьшения сложности индицировать окончание переходных процессов целесообразно по каскадам. Каскад – это множество элементов и макроэлементов, все входы которых принадлежат предыдущим каскадам. Начальным каскадом служат входы схемы. В каждом каскаде предпочтительно обеспечивать максимальную степень сжатия за счет использования Г-триггеров с максимально допустимым числом входов. Оставшиеся незадействованными УФ-сигналы учитываются в следующих каскадах. После последнего каскада формируется общий для всей схемы индикаторный УФ-сигнал.

Одновременно при построении индикации вводятся согласующие инверторы для обеспечения требуемых значений спейсеров на выходах комбинационных фрагментов, Г-триггеров и всей схемы.

Этап 6. Усиление.

После построения всей схемы проверяется соответствие нагрузок элементов схемы их нагрузочным способностям. Вся эта информация предоставляется пользователю в журнальном файле. Далее пользователь может специфицировать в файле задания на синтез сигналы, требующие усиления с помощью параметра "**SS_Amplify**", и повторить синтез.

4 Результат синтеза

Результатом синтеза является описание СС-схемы на языке, определенном заданием на синтез. Рис. 2 демонстрирует в качестве примера начало файла описания синтезированной умножителя 4×4. Входы и выходы схемы представляются соответствующими парами сигналов в ПФ-коде. Дополнительным выходом служит общий индикаторный сигнал (**_UI_79**).

После описания портов приводятся их СС-атрибуты в формате Verilog, содержащие:

- тип сигналов: парафазные ("**BinPhas**"), унарные ("**UnPhas**") или иные,
- спейсерные значения входов и выходов (цифра 0 или 1 в конце списка сигналов в описании атрибута),
- указания, какие входы и выходы схемы индицируются общим индикаторным сигналом ("**PhasOut**").

Схема, полученная в результате логического синтеза, является самосинхронной и не требует дальнейшей проверки на это свойство. Однако реальные емкости трасс, получившиеся при проектировании топологии, могут потребовать дополнительной корректировки схемы по усилению элементов, что планируется в дальнейшем развитии программы синтеза.

5 Требования к библиотеке

Библиотека элементов для проектирования СС-схем отличается от стандартных библиотек своим составом. Рассмотрим вопрос о типе и минимально необходимом наборе элементов для предложенного синтеза.

В соответствии с теорией [2], для реализации комбинационной логики могут использоваться только элементы с монотонной функцией: изотонной или антитонной. Функция является изотонной, если существует ее дизъюнктивная нормальная форма (ДНФ), в которой все аргументы используются без отрицаний. Функция является антитонной, если существует ее ДНФ, в которой все аргументы используются с отрицаниями.

В схемах, изготавливаемых по технологии "комплементарный металл-диэлектрик-полупроводник" (КМДП) антитонными являются все однокаскадные элементы с одним выходом. В таких элементах все входы подключены к затворам транзисторов, стоящих непосредственно в цепях переключения нагрузки. Если к выходу однокаскадного элемента подключить инвертор, то получится изотонный элемент, также пригодный для синтеза.

```
//=====//  
// 25.07.2021 20:46                SS-Synthesis v.3.1           //  
//=====//
```

```

//Prefixes for SS-names:
//"_M" - PFS main
//"_C" - PFS conjugate
//".._I_" - inverse signal for PFS
//"_WW_" - inverse signal to agree
//"_UI_" - unary indicator
//"_LAB_" - label
//Library: MSS_5503
//+++++
module mult_buta_SS ( _M_data_in1, _C_data_in1, _M_data_in2, _C_data_in2,
  _M_data_out, _C_data_out, _UI_79 );
input [3:0] _M_data_in1, _C_data_in1, _M_data_in2, _C_data_in2;
output [7:0] _M_data_out, _C_data_out;
output _UI_79;
// - - - - -
(*UnPhas="_UI_79 1"*)
(*BinPhas={"_M_data_in1[3] _C_data_in1[3] _M_data_in1[2] _C_data_in1[2]",
  "_M_data_in1[1] _C_data_in1[1] _M_data_in1[0] _C_data_in1[0]",
  "_M_data_in2[3] _C_data_in2[3] _M_data_in2[2] _C_data_in2[2]",
  "_M_data_in2[1] _C_data_in2[1] _M_data_in2[0] _C_data_in2[0] 0",
  "_M_data_out[7] _C_data_out[7] _M_data_out[6] _C_data_out[6]",
  "_M_data_out[5] _C_data_out[5] _M_data_out[4] _C_data_out[4]",
  "_M_data_out[3] _C_data_out[3] _M_data_out[2] _C_data_out[2]",
  "_M_data_out[1] _C_data_out[1] _M_data_out[0] _C_data_out[0] 1"}*)
(*PhasOut={"_UI_79 <*> _M_data_out[7] _C_data_out[7] _M_data_out[6]",
  "_C_data_out[6] _M_data_out[5] _C_data_out[5] _M_data_out[4]",
  "_C_data_out[4] _M_data_out[3] _C_data_out[3] _M_data_out[2]",
  "_C_data_out[2] _M_data_out[1] _C_data_out[1] _M_data_out[0]",
  "_C_data_out[0]"}*)
// - - - - -
wire [7:1] _M_005_, _C_005_;

```

Рис. 2 Пример описания результата синтеза умножителя 4×4

Таким образом, в КМДП-библиотеке достаточно иметь только антитонные элементы, включая инвертор, изотонные функции из них получить легко.

Как было сказано выше, предварительный этап синтеза дает описания функций с числом аргументов до трех. Из этого следует, что для комбинационной логики в библиотеке необходимо иметь все антитонные элементы с таким числом входов.

Из процедуры индикации следует, что все УФ-сигналы можно получить с помощью двухвходовых элементов, входящих в необходимый набор. Также из теории известно [2, 3], что любое количество УФ-сигналов можно объединить в один с помощью Г-триггеров с двумя УФ-входами.

В итоге минимальный набор библиотеки элементов для синтеза включает:

- инвертор,
- 2 элемента с двумя входами: И-НЕ, ИЛИ-НЕ,
- 5 антитонных элементов с тремя входами,
- СС-триггер хранения,
- счетный СС-триггер,
- СС-ячейку сдвига,
- Г-триггер с двумя УФ-входами.

Однако для улучшения быстродействия и снижения сложности синтезируемых схем крайне желательно включать в библиотеку не только антитонные, но и изотонные элементы, а также элементы с числом входов более трех. Кроме того, желательно иметь в библиотеке разные триггерные ячейки и Г-триггеры со входами всех типов.

Для проектирования СС-схем на базовом матричном кристалле (БМК) была создана библиотека [6], насчитывающая более 250 элементов с числом входов до девяти.

Для примера, результат синтеза умножителя 4×4 (в составе СС-ядра микроконтроллера [7]) на библиотеке [6] был получен с использованием комбинационных элементов до восьми входов включительно и всех типов Г-триггеров, перечисленных в *Этапе 5* раздела 3.

Состав библиотеки для синтеза напрямую влияет на характеристики синтезируемых схем. Известная методология проектирования СС-схем, называемая NULL Convention Logic (NCL) [8], использует для синтеза весьма ограниченную библиотеку, включающую 29 элементов с нулевым спейсером. С одной стороны, это облегчает формализацию синтеза. С другой стороны, получающиеся NCL-схемы обладают большей сложностью (в 4,49 раза при реализации двоичного счетчика, в 1,13 раза при реализации умножителя 4×4, до 2 раз при реализации более

простых логических схем), худшим быстродействием и повышенным энергопотреблением в сравнении с СС-схемами, синтезированными на базе предлагаемого подхода и библиотеки.

6 Эффективность автоматизированного синтеза

Известно, что автоматический синтез часто уступает ручному по ряду показателей. В данном случае это может происходить в комбинационной схеме или ее части.

Например, сложность автоматически созданного 4-разрядного комбинационного сумматора с последовательным переносом получилась на 30% больше сложности варианта, спроектированного вручную на элементах той же библиотеки. Анализ показал, что вручную разработчик использовал сложный библиотечный элемент, реализующий функцию суммы трех слагаемых в ПФ-коде. Доля такого элемента в сумматоре оказалась значительной. Программа синтеза пока такие элементы не привлекает.

В других случаях, когда доля суммирующей логики меньше, превышение аппаратных затрат получается меньше. Комбинационный умножитель 4×4 по алгоритму Бута, синтезированный программой, уже лишь на 20% сложнее аналога, спроектированного вручную, из-за наличия не только сумматоров, но и другой комбинационной логики. Чем больше доля простой логики в комбинационной СС-схеме, тем ближе по аппаратной сложности оказываются варианты, синтезированные автоматически и вручную.

Данный опыт показал, что для уменьшения сложности необходимо совершенствовать алгоритм синтеза в направлениях:

- анализировать пары логических функций на предмет их связности по входам,
- использовать библиотечные комбинационные элементы с несколькими выходами.

Это позволит применить компактные схемотехнические решения, формирующие ПФ-сигналы одной принципиальной схемой с двумя выходами, которые сейчас не выявляются и не используются.

Например, функция сложения трех переменных по модулю два:

$$Y = A \oplus B \oplus C$$

при синтезе СС-схемы преобразуется в пару функций с парафазными переменными $\{A, AB\}$, $\{B, BB\}$, $\{C, CB\}$:

$$\begin{cases} Y = A \cdot B \cdot C + A \cdot BB \cdot CB + AB \cdot BB \cdot C + AB \cdot B \cdot CB, \\ YB = AB \cdot B \cdot C + A \cdot BB \cdot C + A \cdot B \cdot CB + AB \cdot BB \cdot CB. \end{cases}$$

Эти функции реализуются принципиальной схемой на КМДП-транзисторах, показанной на рис. 3, содержащей 24 КМДП-транзистора и не нарушающей ограничения на число последовательно соединенных транзисторов (не более трех) в цепочках между шинами питания и "земли" и выходом схемы.

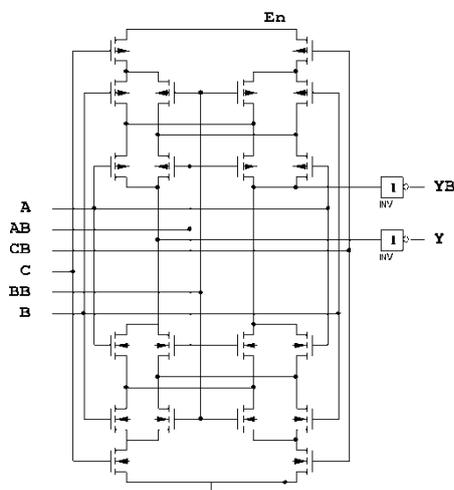


Рис. 3 Принципиальная схема элемента сложения по модулю два с тремя парафазными входами и парафазным выходом

Аналогичная ситуация получается при выполнении СС-схемой функций сложения по модулю два от двух и четырех переменных и некоторых других сильно связанных функций, допускающих компактную реализацию в виде двухвыходовой принципиальной схемы.

В отличие от комбинационных СС-схем, последовательностные схемы при автоматическом синтезе получаются практически идентичными ручным вариантам за счет формализованного подхода, воплощающего эвристические принципы построения регистровых устройств и счетчиков из библиотечных триггерных элементов.

Таким образом, эффективность программы автоматизированного синтеза СС-схем соответствует стандартным САПР для большей части типов СС-устройств. Ввод в программу синтеза эвристического алгоритма выявления функций, сильно связанных по переменным, обеспечит уменьшение сложности синтезируемых автоматически СС-схем до уровня, соответствующего сложности результатов ручного проектирования.

7 Новизна результатов

Авторы считают, что в данной работе новой является методика синтеза СС-схем на основе теории самосинхронности и специализированной библиотеки элементов для проектирования СС-схем на БМК, интегрированной в САПР "Ковчег" (НПК «Технологический центр»). Постановка задачи синтеза СС-схем и приведенный алгоритм реализации являются уникальными и не встречаются в публикациях.

8 Заключение

Особенности СС-схем: двухфазный режим работы, специальное кодирование сигналов и их индцирование диктуют специальные методы проектирования таких схем. К этим методам относятся: дуализация (перевод логических функций в один из СС-кодов, в данном случае парафазный) и построение индикаторной подсхемы.

Расширение библиотеки стандартных элементов САПР "Ковчег" СБИС на БМК элементами, необходимыми для проектирования СС-схем, обеспечило успешную интеграцию в нее специфичных этапов проектирования СС-схем и получение единого маршрута автоматизированного проектирования синхронных и СС-схем. Существенными отличиями маршрута синтеза СС-схемы от маршрута проектирования синхронных схем являются этап повторного синтеза логики и этап анализа на самосинхронность после топологического синтеза. Последний анализ проверяет отсутствие нарушений самосинхронности, привнесенных топологической реализацией схемы.

Предложенный способ синтеза обеспечивает получение схем, обладающих меньшей сложностью и более высоким быстродействием в сравнении со схемами, синтезированными в соответствии с существующей NCL-методологией.

Сравнение с результатами ручного проектирования показывает, что в суммирующих и некоторых других схемах для уменьшения сложности и улучшения быстродействия целесообразно находить и использовать компактные СС-элементы с парафазными выходами.

Дальнейшее развитие методов синтеза будет идти в направлениях выявления и реализации сильно связанных функций, увеличения списка автоматически генерируемых структур, учета данных реальной топологии.

9 Поддержка

Исследование выполнено в рамках государственного задания № 0063-2019-0010.

Список литературы

- [1] Л.П. Плеханов, А.Н. Денисов, Ю.Г. Дьяченко, Ю.А. Степченков, Д.И. Мамонов, Д.Ю. Степченков. Синтез самосинхронных схем в базисе БМК / *Научная конференция «Микроэлектроника – ЭКБ и электронные модули»*. Алушта (Республика Крым), 30.09-05.10.2019 - М: Техносфера, 2019. С. 450-454.
- [2] Автоматное управление асинхронными процессами в ЭВМ и дискретных системах (под ред. В.И.Варшавского) – М.: НАУКА, 1986. – 308 с.
- [3] Л. П. Плеханов. Основы самосинхронных электронных схем – М.: БИНОМ. Лаборатория знаний, 2013. — 208 с.
- [4] И.А. Соколов, Ю.А. Степченков, С.Г. Бобков, В.Н. Захаров, Ю.Г. Дьяченко, Ю.В. Рождественский, А.В. Сурков, Базис реализации супер-ЭВМ эксафлопсного класса // *Информатика и ее применения*. 8(1), 2014. С. 45-70.
- [5] Yosys Open SYnthesis Suite. URL:<http://www.clifford.at/yosys/download.html> (дата последнего обращения 09.08.2021).
- [6] Ю.А. Степченков, А.Н. Денисов, Ю.Г. Дьяченко, Ф.И. Гринфельд, О.П. Филимоненко, Н.В. Морозов, Д.Ю. Степченков, Л.П. Плеханов. Библиотека функциональных ячеек для проектирования самосинхронных полузаказных БМК микросхем серий 5503/5507. М.: Техносфера. 2017. 367 с. — ISBN 978-5-94836-332-5. URL: <http://www.technosphera.ru/lib/book/497> (дата последнего обращения 09.08.2021).
- [7] Степченков Ю.А., Дьяченко Ю.Г., Петрухин В.С. Опыт разработки самосинхронного ядра микроконтроллера на базовом матричном кристалле // *Нано- и микросистемная техника*, №5, 2006., с.29-36.
- [8] Fant K.M., Brandt S.A. NULL convention logic: a complete and consistent logic for asynchronous digital circuit synthesis // *Proceedings of the International Conference on Application Specific Systems, Architectures, and Processors*, 1996. P. 261–273.