



Math-Net.Ru

Общероссийский математический портал

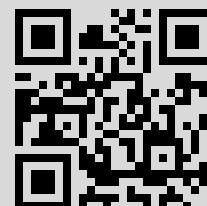
Л. П. Плеханов, Полнота анализа электронных схем на самосинхронность, *Системы и средства информ.*, 2010, том 20, выпуск 1, 48–58

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением  
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 176.15.50.216

7 сентября 2020 г., 13:26:10



УДК 004.312:004.21:004.032.34

## ПОЛНОТА АНАЛИЗА ЭЛЕКТРОННЫХ СХЕМ НА САМОСИНХРОННОСТЬ

*Л. П. Плеханов*

**Аннотация:** В статье обсуждаются вопросы полноты анализа самосинхронности схем, связанные с начальными состояниями и изменениями информационных входов. Показано, что неполный учет этих изменений приводит к ошибочным выводам. Получены достаточные для полноты условия изменения информационных входов. Предложены некоторые способы организации полного анализа для событийных методов и функционального метода.

**Ключевые слова:** самосинхронные схемы; анализ самосинхронности; полнота анализа самосинхронности

### 1. Введение

Обсуждаемые в статье схемы относятся к асинхронным схемам, интерес к которым в последние годы носит исключительный характер. Существующие в отношении рассматриваемых схем термины — русскоязычный *самосинхронные* и англоязычные *self-timed*, *speed independent* и *delay independent* — трактуются широко и неоднозначно, что в научных публикациях требует уточнения. Именно, в данном случае рассматриваются самосинхронные схемы по В.И. Варшавскому [1], имеющие свойство полумодулярности по Маллеру [2] после согласованного замыкания. В свою очередь полумодулярность означает правильное, безошибочное логическое преобразование информации независимо от задержек элементов схемы (вентилей). Далее для краткости такие схемы будут называться *СС-схемами*. На этапе логического проектирования, в соответствии с моделью Маллера, задержки межсоединений после разветвлений считаются пренебрежимо малыми по сравнению с задержками элементов, к которым они подключены.

Важность исследования СС-схем определяется тем, что они обладают рядом уникальных свойств, отсутствующих в синхронных [3, 4] и других асинхронных схемах. В частности, СС-

схемы характеризуются полным отсутствием состязаний и самопроверяемостью в классе выходных константных неисправностей типа залипаний.

Одной из главных задач, без которых создание СС-схем невозможно, является их анализ на самосинхронность. Такой анализ не может быть выполнен средствами моделирования или иными стандартными действиями, а выполняется с помощью специальных алгоритмов, основанных на теоретических разработках Д. Маллера и В. И. Варшавского с сотрудниками. Основная трудность анализа СС-схем состоит в том, что свойство самосинхронности зависит от всех элементов схемы, а проблема иерархического анализа (без раскрытия внутренних блоков) пока не решена.

Базовое свойство полумодулярности определено для замкнутой схемы и одного начального состояния. Традиционно все теоретические и практические методы анализа (за исключением функционального подхода [5], возникшего недавно) разработаны, исходя именно из этих предпосылок. По такому принципу построены и известные теоретические методы анализа: метод диаграмм переходов в полных состояниях Маллера [2], метод диаграмм изменений В. И. Варшавского [6], методы, основанные на сетях Петри [7], и др. Практические реализации некоторых из перечисленных методов — программы ТРАНАЛ, ТРАСПЕК (созданные группой В. И. Варшавского) [6], БТРАН, АСИАН (созданные в ИПИ РАН) [8], также следуют этому подходу. Упомянутые методы, основанные на описаниях в области событий — переключений элементов, будем называть далее *событийными*.

Однако реальные схемы имеют множество начальных состояний и разомкнуты. Для описания их в форме, пригодной для анализа, необходимо сделать согласованное замыкание, требования к которому приведены в [1]. Смысл замыкания состоит в самосинхронной имитации внешней для схемы среды. Задача построения замыкания остается за рамками традиционных методов анализа, т. е. возложена на пользователя.

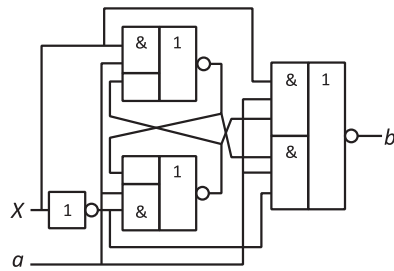
*Сеансом анализа* назовем однократный расчет самосинхронности (полумодулярности) заданной схемы.

Если схема не имеет на входе информационных сигналов (относится к типу управляющих), замыкание для нее тривиально. Характерно, что во всех теоретических работах, посвященных событийным методам, в качестве примеров приводятся либо схе-

мы без информационных входов, либо схемы, где такие входы являются константами.

Для схем, имеющих информационные входы, возникает проблема, как поступать с ними в процессе анализа.

Самый простой случай — когда информационные входы в сеансе не меняются. Анализ можно организовать путем множества сеансов (по числу сочетаний значений информационных входов), в каждом из которых входы неизменны. Такой способ «простого перебора» до последнего времени считался достаточным. Однако в реальности схемы работают в другом режиме, с изменениями входов. Как впервые заметил Ю. А. Степченков, неучет изменений значений входов в сеансе приводит к не-



**Рис. 1** Схема триггера с нарушением самосинхронности:  $X$  — унарный информационный вход;  $a$  — управляющий вход;  $b$  — индикаторный выход

правильным выводам. Причина такой ситуации в том, что существуют схемы, в которых некоторые элементы полностью активизируются при неизменных значениях входов, т. е. их выходы не совершают обоих возможных переходов  $0 \rightarrow 1$  и  $1 \rightarrow 0$ . Если же в сеансе менять входы так, чтобы полностью активизировать подобные элементы, они могут вызвать нарушения самосинхронности.

Например, схема на рис. 1 приведена в [1] (рис. 4.11-в)

как самосинхронная. При неизменном сигнале  $X$  в сеансе инвертор в схеме не активизируется, и схема формально самосинхронна. В реальных условиях, при изменениях сигнала  $X$ , именно этот инвертор вызывает нарушение самосинхронности.

Другой подобный пример простой схемы приведен в [9].

В этих и других случаях при «простом переборе» в каждом сеансе схема не проходит всех состояний, которые могут быть в реальности, т. е. анализ в этих случаях неполон.

Под *полнотой анализа* будем понимать прохождение схемой в одном сеансе анализа всех своих рабочих состояний, т. е. состояний, значимых для реальной работы (уточнение дано ниже).

В имеющихся подходах остаются невыясненными вопросы выбора начальных состояний и подходящих значений вход-

ных сигналов в одном сеансе, прямо влияющие на полноту анализа.

В существующей литературе отсутствуют даже упоминания о подобных проблемах полноты, а приведенные примеры показывают ее актуальность.

Предметом статьи будет выяснение условий, обеспечивающих полноту анализа.

## 2. Полнота анализа

### 2.1. Начальные и рабочие состояния

Рассмотрим сначала вопрос о начальных состояниях. Они бывают *равновесными* и *неравновесными*. Равновесные состояния — это состояния, которые сохраняются неограниченно долго при неизменных входных сигналах. Неравновесные состояния без внешних воздействий со временем переходят в равновесные. Примером неравновесных состояний служат такие, в которых входы и выходы инверторов имеют одинаковые значения.

Неравновесные начальные состояния не представляют практического интереса, и их следует исключить из рассмотрения. Кроме того, анализ с неравновесными начальными состояниями практически всегда диагностирует нарушение самосинхронности.

Начальными состояниями, таким образом, должны быть равновесные состояния. Равновесные состояния зависят от значений входов схемы. При расчете начальных состояний в схемах с внутренними обратными связями выходы некоторых элементов могут принимать произвольные значения. Эти выходные переменные реализуют запоминание, поэтому будем называть их *переменными памяти* (ПП).

В начале работы схемы, при включении питания, ПП примут случайные значения, и равновесное начальное состояние установится в зависимости от этих ПП и значений на входах схемы.

Для практики значимыми являются состояния, которые схема проходит при всевозможных допустимых изменениях входных сигналов. Допустимость сигналов связана с тем, что для парафазных входов запрещены одинаковые, противоположные спейсерному значения (здесь и далее подразумеваются парафазные сигналы со спейсером).

Назовем *рабочими* состояния схемы, обладающие следующим свойством: из любого рабочего состояния можно перейти

в любое другое рабочее состояние с помощью допустимых изменений входных сигналов.

Начальные состояния могут входить или не входить во множество рабочих состояний (*рабочее множество*). В последнем случае перед анализом необходимо искусственно перевести схему в одно из рабочих состояний, например, путем предустановки или подачи нескольких специальных входных наборов, как это и делается в реальности для многих схем.

Будем считать далее, что все изменения входов относятся к допустимым, и анализ начинается с рабочего состояния.

## 2.2. Процедура полного анализа

В управляющих схемах (без информационных входов) в процессе анализа значения служебных входов за счет согласованного замыкания меняются при смене фаз, и все рабочие состояния проходятся автоматически.

Для схем с информационными входами изменений только служебных входов недостаточно. Полноту прохождения рабочих состояний необходимо обеспечивать подходящим изменением информационных входов.

Назовем *инфовходом* виртуальную независимую переменную, представляющую один бит информации на входе схемы. Каждая пара парафазных и бистабильных входных сигналов порождает один инфовход — парафазный и бистабильный соответственно. Один вход из пары принимает значение инфовхода (для парафазного — только в рабочей фазе), а другой — инверсное ему значение. По аналогичному принципу, но несколько сложнее, порождаются инфовходы для мультифазных и мультистабильных входов, но они для простоты здесь рассматриваться не будут.

Служебные входы (управляющие и другие) инфовходов не порождают.

Набор значений всех инфовходов схемы будем называть *инфотестом*.

Обобщенно процедура полного анализа на самосинхронность состоит в следующем.

Выбираются некоторые начальные значения служебных входов и инфовходов, по ним рассчитывается соответствующее равновесное состояние схемы. Переменным памяти можно придавать любые значения, 0 или 1.

Вопрос о принадлежности полученного начального состояния рабочему множеству выходит за рамки статьи. Обычно разработчик схемы обладает такой информацией.

При необходимости схема переводится в одно из рабочих состояний.

Далее проводится один сеанс анализа.

Для управляющих схем анализ на этом заканчивается, и дальнейшее изложение касается только схем с информационными входами.

Для этих схем необходимо подходящим образом менять как парафазные, так и бистабильные инфовходы. Здесь необходимо сделать уточнения.

В соответствии с двухфазным принципом работы [1] схема в процессе анализа (как и в реальности) циклически переходит из *рабочей фазы* в *спейсер* и обратно. Для изменений инфовходов существует дисциплина, определяемая необходимостью имитации самосинхронной внешней среды. Все парафазные инфовходы должны меняться в спейсерной фазе. Бистабильные должны меняться каждый в своей *транзитной* фазе, т. е. в той, в которой эти изменения не активизируют внутренние элементы схемы.

Таким образом, в зависимости от схемы, при каждой смене фазы могут меняться те или иные инфовходы, а в полном цикле «рабочая фаза – спейсер» могут измениться все инфовходы (инфотест целиком). Здесь для общности считается, что если инфовходы в цикле не изменились, то на схему повторно подан предыдущий инфотест.

Обеспечить прохождение всех рабочих состояний схемы в сеансе могут подходящие инфотесты, подаваемые на схему в каждом цикле, поэтому вопрос полноты анализа в описанной процедуре сводится к подбору последовательности инфотестов, обеспечивающей эту полноту.

### 3. Инфотесты для сеанса

Минимально необходимые инфотесты, естественно, будут определяться как структурой схемы, так и предысторией их изменений в сеансе. Из-за разнообразия схем и сложности вопроса либо невозможно сформулировать необходимые условия изменения инфотестов, либо эти условия чрезмерно сложны для реализации. В такой ситуации целесообразно рассмотреть достаточные условия. Достаточные условия будут полезны как для

общего исследования проблемы анализа, так и для построения универсальных алгоритмов. Кроме того, достаточные условия в ряде практических случаев могут совпадать с необходимыми.

Рассмотрим переход схемы из некоторого рабочего состояния  $A$  в рабочее состояние  $B$ . Обозначим  $D_{AB}$  — минимально необходимое число инфотестов (и, соответственно, циклов работы) для такого перехода. Для комбинационных схем это число равно 1, для схем с памятью потребуется несколько инфотестов, чтобы установить нужные значения ПП. Максимальное из чисел  $d_{AB}$  для любых исходных состояний  $A$  обозначим как  $d_B$ .

*Установочной длиной (УД)* назовем число  $d$  — максимальное из чисел  $d_B$  по всем конечным состояниям  $B$ . Смысл установочной длины в том, что после подачи на схему такого или меньшего количества инфотестов ее состояние (рабочее) будет определяться только этими инфотестами, а не исходным состоянием.

Установочная длина характеризует запоминающие свойства схемы, и во многих случаях она определяется просто.

Например, для комбинационной схемы УД равна 1. Для схемы, содержащей один двухтактный триггер и комбинационную часть, зависящую от выходов триггера и входов схемы, УД равна 2, так как конечное состояние в этом случае зависит и от поступившего инфотеста, и от запомненного в ПП результата воздействия предыдущего. В этих же примерах необходимые условия совпадают с достаточными.

Назовем *кластером* упорядоченный набор из  $d$  инфотестов, последовательно поданных на схему.

Для определенности будем считать, что крайний правый инфотест в кластере был подан на схему первым, а крайний левый — последним. При подаче очередного инфотеста на схему существующий кластер заменится на новый по типу очереди или сдвига: крайний правый инфотест удаляется, а очередной инфотест добавляется слева, формируя новый кластер.

Из приведенных выше условий и определений следует, что если в сеансе анализа на схему поступают инфотесты таким образом, что кластеры пробегут все возможные значения, то этого достаточно, чтобы схема побывала во всех своих рабочих состояниях. Действительно, в любое рабочее состояние из любого другого рабочего состояния схема может попасть при подаче на нее некоторой последовательности инфовходов длиной



**Таблица 1** Достаточное для полноты количество тестов (кластеров) в сеансе

$n$	$d$				
	1	2	3	4	5
1	2	4	8	16	32
2	4	16	64	256	1 024
3	8	64	512	4 096	32 768
4	16	256	4 096	65 536	1 048 576

не более  $d$ . Эта последовательность является либо кластером, либо частью одного из кластеров.

Обозначим  $n$  число инфовходов в инфотесте,  $N = 2^n$  — число всевозможных инфотестов для  $n$  инфовходов.

Кластер можно представить в виде записи:  $k_0k_1 \dots k_{d-1}$ , где  $k_i$  — двоичное значение  $i$ -го инфотеста в кластере,  $i = 0, \dots, \dots, d - 1, k_i = 0, \dots, N - 1$ .

Можно заметить, что запись (1) представляет собой  $d$ -значное число с основанием  $N$ , и количество всех возможных кластеров равно количеству представляемых этой записью разных чисел (с нуля):

$$K = N^d = 2^{nd}. \tag{1}$$

Поскольку кластеры сменяют один другого только при подаче очередного инфотеста в схему, то и общее количество инфотестов, обеспечивающее в сеансе все кластеры, должно быть не менее  $K$ .

В табл. 1 показаны значения  $K$  для некоторых значений  $n$  и  $d$ , рассчитанные по формуле (1).

Можно видеть, что запоминающие свойства схемы столь же сильно (экспоненциально) влияют на объем вычислений для полного анализа, что и количество информационных входов.

В табл. 1 «простому перебору» входных значений соответствует столбец при  $d = 1$ , и таблица показывает масштаб увеличения вычислительных затрат, требуемых для полноты анализа.

#### 4. Организация анализа

Изменения входов организуются по-разному в различных методах анализа.

#### 4.1. Событийные методы

В событийных методах изменения входов должны быть реализованы в замыкании. Описание замыкания, наряду с описанием самой схемы, входит в общую систему уравнений для анализа. Поэтому замыкание увеличивает, иногда существенно, общую размерность задачи.

Анализ начинается с произвольно выбранного равновесного начального состояния. Из-за специфики событийных методов изменения инфотестов должны совершаться циклически: после прохождения всех кластеров они вновь должны подаваться в схему в том же порядке.

Требуемые кластеры можно было бы подавать на схему поочередно встык. Однако такой способ чрезмерно избыточен, поскольку не учитывает факт, что кластеры сменяются путем сдвига. Наиболее целесообразно организовать циклическую последовательность инфотестов с тем условием, что никакой набор из  $d$  подряд идущих инфотестов, составляющих один кластер, не повторяется.

Например, в схеме с одним триггером ( $d = 2$ ) и соответствующим ему инфовходом ( $n = 1$ ) циклическая последовательность инфотестов проста (после последнего инфотеста возврат к началу):

$$0-0-1-1-\dots$$

Действительно, никакие из четырех возможных пар (кластеров) не повторяются.

Другой пример: для схемы с двумя параллельными двухтактными триггерами ( $d = 2$ ), соединенными непосредственно со входами схемы ( $n = 2$ ) инфотесты в цикле можно расположить так:

$$0-0-1-0-2-0-3-1-2-1-1-3-3-2-2-3-\dots$$

Здесь также никакие пары (кластеры) не повторяют друг друга, и суммарный набор кластеров (16) обеспечивает полноту анализа.

Нахождение неизбыточных циклических последовательностей инфотестов, удовлетворяющих условиям полноты, в общем случае должно выполняться специальным алгоритмом. Разработка такого алгоритма, однако, выходит за рамки статьи.

## 4.2. Функциональный метод

В функциональном методе [5] каждый инфовход и каждая ПП считаются независимыми переменными, и расчет самосинхронности проводится в функциях от этих переменных.

Анализ производится отдельно для рабочей фазы и спейсера. Исходным состоянием для вычисления функций элементов в одной фазе является заранее вычисленные функции другой фазы.

Изменения входов реализуются введением  $n$  новых независимых переменных для инфовходов. Исходные состояния рассчитываются при «старых» значениях инфовходов, и на входы, в соответствии с их дисциплиной изменения, подаются «новые» значения. Наличие двух независимых переменных для каждого инфовхода моделирует все четыре возможных перехода между «старыми» и «новыми» значениями:  $0 \rightarrow 0$ ,  $0 \rightarrow 1$ ,  $1 \rightarrow 0$ ,  $1 \rightarrow 1$ .

В результате, если независимым переменным придавать всевозможные значения, учитываются все состояния: и в которых схема начинает работу, и в которых она может оказаться как при неизменных, так и при изменяющихся входах.

Таким образом, данный подход обеспечивает полноту анализа ценой удвоения независимых переменных для инфовходов и соответствующего увеличения вычислительной сложности.

## 5. Заключение

Анализ самосинхронности схем, имеющих информационные входы, бывает ошибочным, если он не полон, т.е. в процессе анализа схема проходит не все состояния, возможные в реальной работе. Проблема полноты связана с начальными состояниями и изменениями информационных входов.

В существующей литературе обсуждение вопросов полноты отсутствует, приводимые примеры либо не имеют информационных входов, либо такие входы не меняются в процессе анализа.

В статье рассматриваются начальные и рабочие состояния схем, приводятся достаточные условия для полноты анализа.

Для событийных методов указывается достаточный способ смены входных значений. Для функционального метода установлено, что учет полноты приводит к увеличению размера обрабатываемых логических функций.

Показано, что пути решения проблемы полноты связаны со значительным увеличением объема вычислений по сравнению с существующими (неполными) способами.

Автор выражает благодарность Ю. Г. Дьяченко за конструктивное обсуждение статьи.

### Литература

1. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах / Под ред. В. И. Варшавского. — М.: Наука, 1986. 400 с.
2. *Muller D. E., Bartky W. C.* A theory of asynchronous circuits // Symposium (International) on the Theory of Switching Proceedings. Part 1. — Harvard University Press, 1959. P. 204–243.
3. *Филин А. В., Степченков Ю. А.* Компьютеры без синхронизации // Системы и средства информатики. Вып. 9. — М.: Наука, 1999. — С. 247–261.
4. *Плеханов Л. П., Степченков Ю. А.* Экспериментальная проверка некоторых свойств строго самосинхронных схем // Системы и средства информатики. Вып. 16. — М.: Наука, 2006. — С. 476–485.
5. *Плеханов Л. П.* Анализ самосинхронности электронных схем функциональным методом // Системы и средства информатики. Вып. 18. — М.: Наука, 2008. — С. 225–233.
6. *Kishinevsky M., Kondratyev A., Taubin A., Varshavsky V.* Concurrent hardware: The theory and practice of self-timed design. — London: John Wiley & Sons, 1994. 388 p.
7. *Kondratyev A., Kishinevsky M., Taubin A., Cortadella J., Lavagno L.* The use of Petri nets for the design and verification of asynchronous circuits and systems // J. Circuits Systems Computers, 1998. Vol. 8. No. 1. P. 67–118.
8. *Рождественский Ю. В., Морозов Н. В., Степченков Ю. А., Рождественскене А. В.* Универсальная подсистема анализа самосинхронных схем // Системы и средства информатики. Вып. 16. — М.: Наука, 2006. — С. 463–475.
9. *Плеханов Л. П.* Реализация функционального метода анализа самосинхронности электронных схем // Системы и средства информатики. Вып. 19. — М.: Наука, 2009. — С. 142–148.