

## Принципы построения и работы строго самосинхронных электронных схем\*

Л. П. Плеханов

Дискретные схемы подразделяются на два класса - синхронные и асинхронные. В синхронных схемах внутренние сигналы и выходы начинают меняться только после очередного изменения специального, внешнего по отношению к схеме, тактового (синхронизирующего) сигнала. В отличие от них в асинхронных схемах внутренние сигналы и выходы меняются после изменений их входов. Большинство используемых ныне схем - синхронные.

Как синхронные, так и асинхронные схемы подвержены сбоям - нарушениям правильной работы из-за изменений задержек элементов. Такие изменения естественны в процессе работы схем и проистекают от разбросов параметров, условий работы - температуры и напряжения питания, старения и, возможно, других факторов. "Коварство" сбоев состоит в том, что они появляются не всегда и практически не поддаются прогнозу.

При проектировании синхронных схем всегда делается предположение о том, что все переходные процессы заканчиваются за длительность такта. Поэтому практическая борьба со сбоями в синхронных схемах ведётся в основном на пути подгонки задержек и выбора подходящей длительности такта.

Потенциальная возможность сбоев определяется свойством схемы, известным как *состязания (гонки)* сигналов. Состязания - это свойство схемы, когда её поведение зависит от того, что какой-либо из фронтов сигналов на входе элемента придёт раньше или позже других фронтов.

Доказано (например, в [1]), что любая невырожденная логическая функция от нескольких переменных (а значит, и элемент, её реализующий) имеет состязания. Из этого следует, что для борьбы с состязаниями необходимы специальные меры.

Для асинхронных схем было предложено несколько подходов (моделей) борьбы с состязаниями (обзор есть в [1]). Модели, отличные от предмета этой статьи, не решают до конца проблемы состязаний. В некоторых из них существенны предположения о соотношении задержек элементов или входных сигналов - при их нарушении, значит, гонки возможны. Большинство моделей накладывают требования на окружение - схемы, формирующие входы для исследуемой схемы. Это необходимость парафазных сигналов и определённая дисциплина изменения входов. Однако способы организации окружения не обсуждаются, и таким образом, проблема состязаний переносится на окружающую среду.

В то же время ещё с 50-х годов существует теория асинхронных схем Дэвида Маллера [2]. В ней доказано существование схем, поведение которых не зависит от задержек элементов, и которые не имеют состязаний. Тем самым проблема сбоев решается радикально - в таких схемах их нет по определению. В

---

\* Работа выполнена при финансовой поддержке по Государственному контракту N 1.4/03 (регистрация РАН: N 10002-251/ОИТВС-04/103-098/260503-201)

теории разработан математический аппарат для подобных схем, исследовано их поведение и приведён критерий независимости от задержек.

По странности, большинство последующих исследователей прошли мимо этой теории и не воспользовались её результатами. Возможно, это произошло потому, что теория не даёт никаких рекомендаций по построению практических схем - как реализовать логические функции, счётчики, регистры и так далее.

Теорию Маллера эффективно развили В.И.Варшавский с сотрудниками [1, 3]. Ими разработаны методы целенаправленного построения практических схем, названных самосинхронными, и множество типовых решений. Однако эти работы развивались в основном как раздел теории автоматов (об этом говорят названия книг и других публикаций этих авторов) и оказались слишком трудны для подавляющего большинства разработчиков.

В результате сложилась ситуация, когда разработчики электронной аппаратуры либо совсем не знают, либо очень слабо представляют, как работают самосинхронные схемы и каковы их главные отличия от других схем. Материал по принципам работы таких схем разбросан по публикациям и не выделен явно.

Самосинхронные схемы имеют ряд уникальных свойств, недостижимых для синхронного способа реализации [4]. В последнее время всё чаще самосинхронными стали называть асинхронные схемы, не являющиеся полностью независимыми от задержек элементов. Поэтому во избежание неточностей схемы, полностью удовлетворяющие критерию Маллера, мы называем строго самосинхронными схемами (ССС-схемами).

Далее излагаются три основных принципа построения СССР-схем, отличающие их от других типов дискретных схем.

Рассматриваемый подход базируется только на булевой двоичной логике. Будем считать, что элементы схем имеют конечные задержки, и термины "произвольная" и "любая" задержка не будут означать её бесконечность.

## 1. Индикация

Для построения любой дискретной схемы принципиально важным является знание моментов окончания переходных процессов в различных частях схемы или, что равнозначно, задержек её элементов. В синхронных схемах это знание учитывается априорно и определяет период повторения тактовых сигналов.

Допуская произвольные задержки и отказываясь от синхронного принципа, мы должны прямо измерять моменты завершения переходных процессов в различных точках схемы. Как это можно сделать? Определить такие моменты на выходах схемы означает сформировать некоторый двоичный сигнал, значения которого показывают факт окончания переходного процесса в ответ на изменение её входов. Такой сигнал будем называть индикаторным сигналом (И-сигналом), а схему его формирования – индикатором.

Здесь возможны два варианта. В первом выходные сигналы не зависят друг от друга, что свойственно комбинационным схемам. Во втором варианте – ячейках памяти - выходы зависят друг от друга.

Индикация для этих двух вариантов осуществляется по-разному.

### *А. Кодовая индикация*

Возьмём для примера комбинационную схему с двумя выходами (рис. 1а).

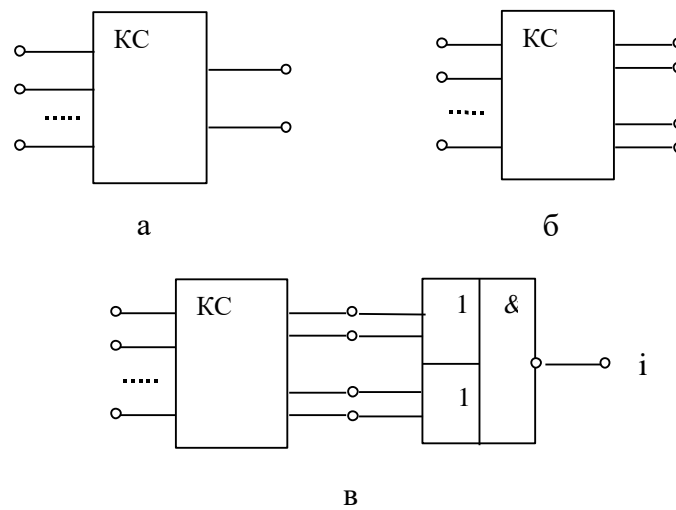


Рис. 1. Комбинационная схема (КС): исходная схема (а), схема в кодах (б), схема в кодах с индикатором (в) (i - индикаторный сигнал).

Два выхода схемы на рис. 1а независимы друг от друга и могут принимать такие сочетания (назовём их *рабочими* наборами): 00, 01, 10, 11. Предположим, что при некотором изменении входов выходные сигналы должны перейти от набора 01 к набору 10.

Такой переход может произойти следующими путями: 01 → 10, 01 → 00 → 10 или 01 → 11 → 10. Каков бы ни был путь перехода, следующий же набор после исходного также относится к рабочему, и мы лишены возможности узнать, закончился переходный процесс или ещё будет продолжаться.

Таким образом, в «обычной» схеме, с «обычными» выходами не удаётся найти способ определения окончания переходных процессов.

Единственным решением этой задачи в рамках двоичной логики является расширение кодового пространства выходных сигналов. Для этого, кроме рабочих наборов необходимо ввести ещё множество *переходных* наборов. Эти два множества не должны пересекаться, чтобы быть легко различимыми с помощью простых электронных схем. Другими словами, надо применить подходящее кодирование сигналов схемы.

Возьмём наиболее простой способ кодирования - парафазный: исходное значение 1 кодируется как 01, а значение 0 – как 10. В этом случае от схемы на рис. 1а мы переходим к эквивалентной схеме на рис. 1б. Для неё рабочими будут четыре набора: 0101, 0110, 1001 и 1010, а остальные двенадцать – переходными.

Далее будем считать, что все переходы между рабочими наборами происходят только через переходные наборы (реализация этого условия изложена в следующем разделе).

В принятых условиях переход 01 → 10 исходной схемы может реализовываться по-разному в эквивалентной схеме, например, так: 0110 → 0010 → 0011 → 0001 → 1001. Видно, что в этой последовательности первый и последний наборы – рабочие, остальные – переходные. Поскольку оба множества чётко разделены, мы можем построить схему индикации переходов (рис. 1в).

Кодовые системы, пригодные для такой индикации, получили название самосинхронизирующихся кодов [3].

## Б. Индикация ячеек памяти

Рассмотрим простейшую ячейку памяти - бистабильную ячейку (БЯ) (рис. 2а). В ней значения обоих выходов в стабильном состоянии противоположны друг другу, а в переходном – совпадают.

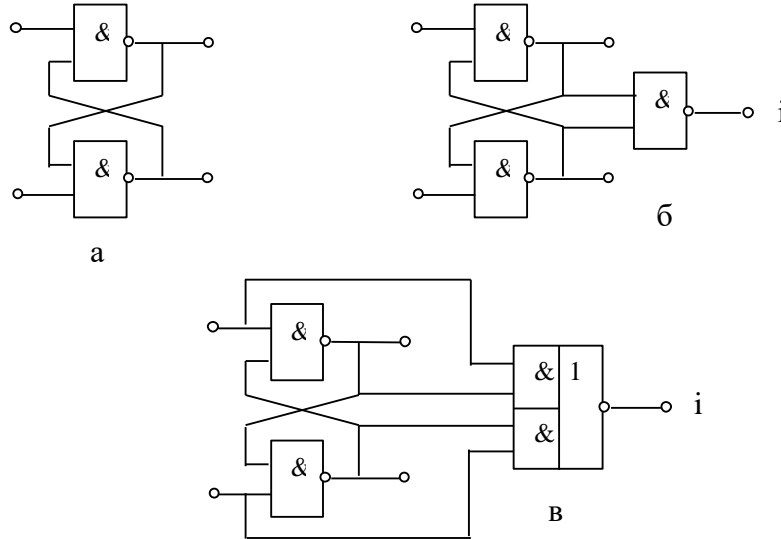


Рис. 2. Бистабильная ячейка ( $i$  – сигнал индикатора): исходная (а), с индикацией по выходам (б), с индикацией по входам и выходам (в).

Есть две идеи индикации БЯ [3]. Первая - индикация с использованием только выходов ячейки (рис. 2б). Предположим, в исходном состоянии входы ячейки имеют значения 11, выходы – 01 (по порядку сверху вниз), сигнал индикации – 1. Подадим на входы значения 01. Тогда выходы будут изменяться так: 01  $\rightarrow$  11  $\rightarrow$  10, а И-сигнал соответственно 1  $\rightarrow$  0  $\rightarrow$  1. В результате значение 1 И-сигнала показывает стабильное состояние БЯ, а 0 - переходное. Такой способ индикации удаётся использовать вместе с режимом гашения, когда входы зануляются либо непосредственно, либо с помощью дополнительного сигнала, подаваемого на оба элемента БЯ. Однако в этом режиме теряется информация, ранее запомненная в ячейке.

Если необходимо сохранять информацию в БЯ, используется другая идея, состоящая в установлении соответствия между входами и выходами БЯ (рис. 2в). Если выходы соответствуют входам, то переходные процессы закончились, иначе они не завершены. В тех же условиях И-сигнал для этого способа изменяется как 0  $\rightarrow$  1  $\rightarrow$  0, где 0 - стабильное состояние, 1 - переходное.

Принципиально важным моментом в изложенных способах индикации является то, что по построению выход индикаторного элемента индицирует окончание переходных процессов не только в основной схеме, но также и в самом индикаторе. Другими словами, нет необходимости строить специальную схему индикации для индикатора, что не обязательно справедливо для других подходов, например, физического.

В схеме может быть много И-сигналов. Все они комбинируются по несложным правилам в один, общий для всей схемы, сигнал. Таким образом, выходной И-сигнал ССС-схемы индицирует не только выходные сигналы схемы, но и её внутренние элементы и сигналы.

Индикация переходных процессов является важнейшим фундаментальным принципом построения ССС-схем. Реализация этого принципа организует работу таких схем *по факту* завершения переходных процессов, а не *по предположению*, как это имеет место при построении схем других типов. Она же делает работу схем максимально быстрой в текущих условиях.

## 2. Двухфазная дисциплина

Вернёмся к примеру комбинационной схемы на рис. 1. Пусть исходные выходы по-прежнему делают переход  $01 \rightarrow 10$ . Так как задержки в схеме по предположению произвольны, при некотором их соотношении может возникнуть коллизия в схеме на рис 1б. Например, выходные кодированные сигналы будут осуществлять переход  $0110 \rightarrow 0010 \rightarrow 1010 \rightarrow 1011 \rightarrow 1001$ . В этом случае первая пара уже закончила свой переход в новое состояние, а вторая пара, в силу больших задержек, ещё не успела измениться. В результате в середине перехода возникло состояние  $1010$ , относящееся к рабочему множеству, и индикатор должен показать завершение переходного процесса, хотя фактически тот не окончен.

Исключение подобных коллизий состоит в наложении требования недопустимости появления других рабочих наборов при переходе, кроме начального и конечного, то есть в определённой дисциплине переходов.

Опыт показал, что наиболее эффективной и простой дисциплиной является следующая. Среди переходных наборов выделяется один или несколько наборов, называемых промежуточными, или *спейсерами*. Затем налагается требование, чтобы любой переход обязательно совершался через какой-либо один, и только один спейсер. В этом случае каждый переход осуществляется в два этапа, или фазы: от начального рабочего набора через другие промежуточные к спейсеру и от спейсера – к конечному рабочему набору.

Такая дисциплина получила название *двухфазной*. Фазу перехода от рабочего набора к спейсеру принято называть *спейсерной*, а от спейсера к рабочему набору – *рабочей* фазой. В рабочей фазе реализуются те функциональные преобразования сигналов, для которых схема предназначена (логические операции, суммирование и тому подобное).

В каждом месте схемы может быть свой постоянный спейсер. Обычно в качестве спейсеров берут наборы  $0000$  или  $1111$ .

При двухфазной дисциплине задачу индикации в ряде случаев можно упростить по сравнению с общим случаем: поскольку прохождение через спейсер обязательно, индикатору нет необходимости отличать рабочие наборы от переходных. Ему достаточно распознать только один спейсер, который является «полномочным представителем» всего множества переходных наборов в данном месте схемы.

Важно здесь отметить, что задача индикации фаз эквивалентна задаче индикации рабочих и переходных наборов, и её реализация также позволяет обеспечить работу схемы при любых задержках.

Значение двухфазной дисциплины в ССС-схемах состоит в строгом упорядочении всех переходов сигналов, упрощении индикации и организации работы схемы на основе чередования двух фаз.

## 3. Обратная связь

В общем случае на входы ССС-схемы наряду с информационными сигналами (инфосигналами) подаётся и сигнал, показывающий окончание перехода всех инфосигналов в очередную фазу. По отношению к ССС-схеме этот сигнал

будет управляющим. Управляющий сигнал обеспечивается окружающими схемами (внешней средой). На выходе ССС-схемы должен присутствовать её И-сигнал.

По определению, внутри ССС-схемы состязаний нет. Однако они могут возникнуть на входах схемы, и для их исключения необходима определённая дисциплина взаимодействия с внешней средой. Такая дисциплина в данном случае диктуется двумя правилами:

1. Управляющий сигнал переходит в очередную фазу только после перехода в ту же фазу входных инфосигналов. Это правило индикации для внешней среды.
2. Входные инфосигналы схемы могут меняться только в её фазе спейсера. В рабочей фазе, когда инфосигналы обрабатываются, они должны оставаться неизменными.

Взаимодействие иллюстрируется на рис. 3.

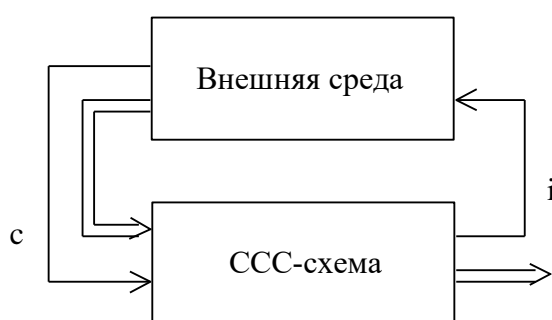


Рис 3. Взаимодействие ССС-схемы с внешней средой.

с - управляющий, i - индикаторный сигналы

Предположим, что значение 0 управляющего и индикаторного сигналов соответствует спейсеру, а значение 1 - рабочей фазе. Работа схемы происходит циклически, по четыре шага в цикле. Примем исходную фазу системы - спейсер.

- 1) Переход  $c = 0 \rightarrow 1$ . Установка на входе схемы новых инфосигналов, запрос внешней среды на обработку.
- 2) Переход  $i = 0 \rightarrow 1$ . Рабочая фаза схемы: обработка входных сигналов и получение выходных.
- 3) Переход  $c = 1 \rightarrow 0$ . Ответ внешней среды на окончание обработки.
- 4) Переход  $i = 1 \rightarrow 0$ . Спейсерная фаза схемы, запрос на новые входные инфосигналы.

Легко заметить, что приведённый процесс полностью аналогичен известному механизму запрос-ответного взаимодействия, широко используемому при асинхронном обмене информацией. Для ССС-схем такое взаимодействие является естественным и единственно возможным механизмом, обеспечивающим независимость от задержек элементов.

Таким образом, ССС-схема работает с окружающей средой по принципу отрицательной обратной связи по фазе работы: получив от схемы сигнал окончания очередной фазы, внешняя среда формирует на входе схемы сигналы противоположной фазы. Такой режим является режимом автогенерации и характерен для ССС-схем.

Этот же режим, в силу обязательности индикации, является и режимом ожидания всех необходимых переключений элементов схемы и тем самым

обеспечивает безопасность её работы при неисправностях. Действительно, при возникновении константных неисправностей на выходе элементов схема, ожидая переключения, останавливается, автогенерация прекращается, и ошибочная информация не проходит на выход. Остановку работы схемы можно использовать как признак наличия неисправностей для организации восстановления.

Из порядка запрос-ответного взаимодействия видно, что внешняя среда взаимодействует с ССС-схемой симметрично по такому же механизму. Поэтому внешней средой ССС-схемы может быть другая ССС-схема, что позволяет легко строить составные схемы любой сложности, например, конвейеры. Но окружением для ССС-схемы могут быть и устройства другой природы, но с обязательной обратной связью. Например, если ССС-схема сформировала сигнал для исполнительного механизма - шагового двигателя, то она должна получить в ответ сигнал, что требуемое перемещение совершено. Тем самым в контур обратной связи включается и исполнительный механизм – внешняя среда, и теперь работа системы ССС-схема – шаговый двигатель не зависит от задержек не только элементов ССС-схемы, но и шагового двигателя.

Особый случай применения ССС-схем - в устройствах, где время является главным фактором, например, в таймерах. В них всегда имеется задающий генератор. Такое использование накладывает естественное ограничение на величины задержек любой схемы, в том числе и ССС. При выполнении этих ограничений задающий генератор для ССС-схемы является элементом внешней среды, обеспечивающим замыкание обратной связи и не влияющим на внутреннюю синхронизацию в ССС-схеме.

Значение принципа обратной связи в ССС-схемах состоит в организации их естественной циклической работы. Кроме того, в течение каждого цикла, попутно с обработкой информации, происходит тестирование всех элементов схемы (переключился или нет в очередной фазе), что обеспечивает безопасность работы схемы.

## **Заключение**

Строго самосинхронные схемы, построенные на трёх основных принципах - индикации, двухфазной дисциплине и обратной связи - класс схем, обладающих уникальными свойствами отсутствия состязаний и достоверности работы. Такие свойства, отсутствующие у обычных синхронных схем, весьма желательны для всех электронных схем. Поэтому ССС-схемы перспективны как для применений общего назначения, так и особенно в промышленной, бортовой и автономной аппаратуре.

## **Литература**

1. Аперидические автоматы. / Под ред. В.И.Варшавского. М.: Наука, 1976. – 423с.
2. Muller D.E., Bartky W.C. A theory of asynchronous circuits. - Proc. Int. Symp. on the Theory of Switching, Part 1. Harvard University Press, 1959, p. 204-243.
3. Автоматное управление асинхронными процессами в ЭВМ и дискретных системах. / Под ред. В.И.Варшавского. М.: Наука, 1986. – 398с.
4. А.В.Филин, Ю.А.Степченко. Компьютеры без синхронизации // Системы и средства информатики. Вып. 9. М.: Наука, 1999. С. 247-261.