

УДК 519.713

ББК 32.815

К ПЕРЕСМОТРУ КРИТЕРИЯ ЦИКЛИЧЕСКИХ ЦЕПЕЙ

Любченко В.С.¹

(ФГБУН Институт проблем управления им.

В.А.Трапезникова РАН, Москва)

В работе предложено пересмотреть критерий оценки корректности схем структурных автоматов, представленных циклическими цепями. В этих целях сформулирован закон функционирования автоматов, введена модель абстрактного дизъюнктивного автомата, определены сети из подобных автоматов, изложены графические методы реализации операций композиции и декомпозиции автоматов. Следствием полученных результатов явилась возможность исключения элементов памяти из структурно полного набора элементов.

Ключевые слова: закон функционирования автоматов, дизъюнктивный структурный автомат, автоматные сети, структурно полный набор элементов, циклические цепи, операции композиции и декомпозиции автоматов, параллельные системы.

1. Введение

По существу теория автоматов ориентирована на проектирование отдельных блоков цифровых схем – устройств, преобразующих дискретную информацию – *дискретных автоматов* [1]. В результате в ней выделилось два раздела – абстрактная и структурная теории автоматов. Сформировалось соответственно и две группы подходов к определению самого понятия конечного автомата – группы макроподхода и микроподхода [2].

¹ Любченко Вячеслав Селиверстович, ведущий инженер (slubch@mail.ru).

В рамках общей теории автоматов абстрактная теория решает задачи создания «внешнего» поведения модели объекта, а структурная – синтез модели в форме структурного автомата, созданной на первом этапе абстрактной модели. При этом структурный автомат представляет собой цифровую (переключательную, логическую и т.п. понятия) схему из элементарных [абстрактных] автоматов, представляющих множество взаимодействующих простых формальных моделей реальных дискретных элементов [3,4,5].

Современные требования к проектированию систем дополнились задачами реализации систем, представляющих собой множество далеко не элементарных моделей блоков/процессов. Но теория автоматов как не имела, так и не создала теории блочного синтеза (см. утверждение в [1]). Собственно даже название базовых моделей – комбинационные и последовательностные схемы отражает подобную ситуацию. На текущий момент, к сожалению, в этом смысле мало что изменилось.

Тем не менее, попытки развить теорию автоматов в направлении композиций автоматов известны. К ним относятся работы А.Н.Мелихова по алгебре автоматов [6]. Но использование в этом случае только абстрактных автоматов ограничило возможности и применимость полученных результатов. Казалось бы, структурная теория автоматов решает задачи многокомпонентных систем. Однако рассматриваемые в ней элементарные модели не позволяют обобщить ее методы на схемы из произвольных компонент.

Далее на примере схем, содержащих так называемые циклические цепи (другие названия – обратные связи, алгебраические цепи и т.п.), будет показано, в какой мере автоматные модели могут стать основой общей теории для систем из произвольных компонент. Речь идет даже о более общем случае – теории систем параллельного типа.

В этих целях будут представлен более адекватно отражающий реальные процессы закон функционирования автоматов. Будет введена формальная модель, позволяющая в компактной и удобной форме описывать дискретные модели на уровне и отдельных компонент, и многокомпонентных систем в целом. Определены основные алгебраические операции над компонен-

тами, позволяющие решать задачи анализа и синтеза сложных систем.

2. Проблема корректности схем

В теории автоматов сформулированы критерии построения корректных схем (структурных автоматов). К ним относятся правила построения схем синхронного и асинхронного типов, приведенные в монографии [3,4]. Хотя одной из точек отсчета создания подобных правил, а также определения структурно полного набора элементов можно считать описание «первичных элементов», разбитых на два класса – «производящих» и «задерживающих» элементов [5].

Более детальный анализ правил построения схем структурных автоматов приведен в монографии В.М.Глушкова [1]. *Второе условие корректности* говорит, что *неоднозначность элементарных сигналов в каком-либо узле схемы хотя бы в один момент времени является недопустимой*. При этом к источникам порождения подобной некорректности относятся так называемые *циклические цепи*.

Однако есть примеры схем, которые с точки зрения второго условия некорректны, но вопреки этому достаточно успешно используются на практике. Тем не менее, преодолевая подобное противоречие, можно определить формальную модель, которая теорию и практику приводит в соответствие друг другу. Более того, в ее рамках для построения корректных схем будет достаточно элементов «производящего» типа, имеющих более привычное современное название – логические элементы.

Полученные в работе результаты существенно влияют на прочно устоявшиеся представления о корректности схем. Важно, что их можно перенести и на общую теорию сложных систем. При этом элементы памяти, обязательно включаемые по существующим представлениям в структурно полный набор элементов, исключаются из него. Их функции в этом случае сводятся, по сути, к решению проблем гонок сигналов, порождаемых комбинационной частью схемы структурного автомата.

Далее будет показано, что лишь только изменением закона функционирования автомата можно *исключить из теории*

автоматов проблему корректности схем, представленных циклическими цепями. В этом случае они просто переводятся в разряд обычных [корректных] схем.

Заметим, что исключение в рамках тех или иных теорий рассмотрения циклических цепей не означает, что проблем для них не существует. Просто в определенных ситуациях она в существующей теории решается автоматически. Так, если взять теорию переключаемых схем Миллера, то в ней схемы представлены *основной моделью для последовательностных схем* [3,4], которая содержит элементы памяти и тем самым автоматически реализует условие корректности циклических схем.

3. Синхронные и асинхронные модели

Чаще всего *асинхронной моделью* считают конечный автомат, множество внутренних состояний которого состоит только из *устойчивых состояний*. Для последних причиной перехода в следующее состояние является *изменение* сигналов на входах автомата. Автоматы, которые включают еще и другие состояния относятся к типу *синхронных автоматов*. Ясно, что при таком определении первые от вторых отличаются видом функций переходов.

Но существует другой подход, определяющий синхронные автоматы, как модели, дискретный такт которых соответствует фиксированному отрезку реального времени. В этом случае вид функций переходов ничем не ограничен, а переход между состояниями определяется часто из единого источника – *генератора синхронизирующих импульсов*.

Соответственно у асинхронных автоматов в этом случае переходы между состояниями могут происходить в произвольные моменты времени и, вообще говоря, без всякой привязки к тому или иному источнику синхронизации, а определяться, например, длительностью исполнения переходов между состояниями.

В рамках второго подхода считается, что «теория асинхронных автоматов в ряде случаев может быть сведена к синхронному случаю». При этом функционирование автомата рассматривается в *абстрактном автоматном времени*, принимающем целые неотрицательные значения ($t = 0, 1, 2, \dots$). Это

позволяет строить теорию автоматов, как теорию синхронных автоматов, охватывающую в значительной мере и теорию асинхронных автоматов (см. подробнее в [1]).

Связывая «по определению» функционирование автоматов с автоматным временем, можно провести и соответствующую классификацию сложных многокомпонентных систем. Определяя их как сети из автоматов, где одни представляют системы, работающие в *едином абстрактном автоматном времени* или просто *едином времени*, а другие – системы, у которых каждая компонента обладает своим дискретным временем.

Далее *синхронными или асинхронными многокомпонентными параллельными системами* будем считать автоматные сети с *единым временем*, дискретный такт которых имеет соответственно *фиксированное* или *плавающее* значение. При этом «введение общего автоматного времени не означает, вообще говоря, отказ от рассмотрения асинхронных автоматов» [1].

Привязка к единому времени существенна с точки зрения создания теории подобных сетей. Только в едином времени можно рассматривать компонентное состояние сложной системы (сети, схемы и т.п.) как множество состояний компонент в терминах, например, алгебры автоматов [6]. Вне подобной привязки к единому времени состояние системы принимает «расплывчатый» характер из-за неопределенности ответа на вопрос о текущих состояниях компонент системы в рассматриваемый момент времени.

Существует и другая классификация сетей, определяющей синхронные сети, как сети с единым временем, а асинхронные – как сети с индивидуальными временами у компонент. Исходя из этого, строится и теория подобных асинхронных/синхронных систем. В результате формируется, например, теория схем, не зависящих от скорости, но, как оказывается, весьма зависимых от конфигурации схем и их взаимодействия с внешним миром (подробнее см. [4]).

Далее под автоматными сетями, если это только не оговорено отдельно, будут подразумеваться сети, компоненты которых функционируют в едином времени. Они будут подразделяться на синхронные и асинхронные сети, исходя из привязки к длительности такта абстрактного автоматного времени. В дру-

гой классификации они бы относились к классу синхронных сетей.

Но это не означает, что в рамках рассматриваемой модели сетей нельзя рассматривать сети, называемые асинхронными по другой классификации. Это, повторимся, только «позволяет строить общую теорию автоматов, как теорию синхронных автоматов, охватывающую в значительной мере и теорию асинхронных автоматов». Примером подобной возможности может служить «синхронная реализация» инерционной задержки, рассматриваемой обычно в рамках обсуждения проблем теории асинхронных схем [4].

4. Циклические цепи

Как уже было отмечено, к одному из источников некорректности цифровых схем современная теория автоматов относит *циклические цепи* или *петли*. Считается, что «некорректность такого рода не будет иметь места, если хотя один из входящих в петлю автоматов является автоматом Мура, а не автоматом Мили» [1].

Циклической цепью называется последовательность из одного и более автоматов, начинающаяся любым входным узлом первого автомата цепи и оканчивающаяся любым выходным узлом последнего автомата, подключенным к входному узлу первого автомата.

Для любого автомата, включенного в циклическую цепь, существуют соединения, когда выходной узел некоторого предыдущего автомата соединен с одним из входных следующего за ним автомата. При этом цепь, состоящая из одного автомата, отождествляет один из его выходных узлов с любым из его входных узлов.

На рис.1 приведена абстрактная схема, на примере которой в [1] рассматривается проблема неоднозначности сигналов. Автоматы, показанные на схеме прямоугольниками, являются инверторами и составляют циклическую цепь. Утверждается, если в какой-то момент времени, например, в узле 1 сигнал равен 0, то, пройдя по цепи, он в этот же момент времени в этом же узле должен быть равен 1. В этом и заключено противоречие.

Но если схемы, содержащие циклические цепи типа рассмотренной на рис.1, в теории некорректны, то схема, аналогичная приведенной на рис.2, используется на практике в качестве генератора сигналов. Это свидетельствует о явно имеющемся **противоречии** теории с практикой.

Можно привести примеры и других схем, которые некорректны в теории, но востребованы на практике. Одна из таких схем – широко известный асинхронный RS-триггер.

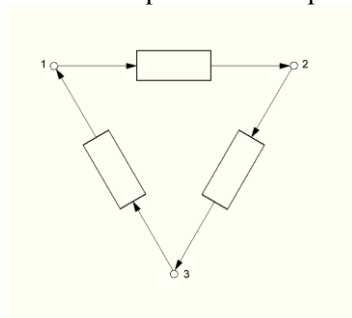


Рис. 1. Пример циклической цепи

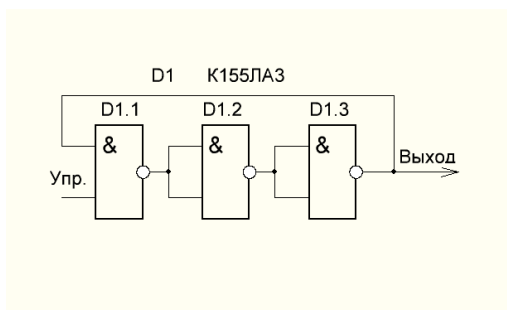


Рис. 2. Схема генератора на элемента И-НЕ

Если появляются реальные факты, противоречащие существующей теории, то это повод для ее критической оценки. А поскольку «природу отменить» невозможно, то, видимо, обнаруженное противоречие объясняется только противоречиями теории дискретных систем или, если точнее, теории дискретных автоматов.

Целью работы является устранение существующих проблем теории, выявляемых схемами, включающими циклические цепи. В этих целях будут определены: более адекватная реальным схемам формальная модель автоматного вида и закон ее функционирования. При этом доказательства корректности предложенных решений будут базироваться на введенных операциях алгебраического типа – операции композиции и декомпозиции автоматов.

Одна из них, операция композиции, позволяет находить результирующую однокомпонентную модель для произвольной многокомпонентной дискретной системы, представленной сетью автоматов в едином времени. Найденная модель служит доказательством свойств систем, представленных в данном случае схемами, содержащими циклические цепи.

5. О законе функционирования автоматов

Отличия работы абстрактной схемы на рис.1 от ее реального аналога на рис.2 связаны с трактовкой момента появления выходных сигналов по отношению к изменению входных сигналов. В существующей теории такая зависимость считается «мгновенной», хотя для реальных элементов – и это хорошо известно – определяется существующими задержками.

Напомним, что закон функционирования автомата определяется функциями переходов и выходов [1, 6]. В случае автоматов первого рода он задается уравнениями:

$$(1) \quad a(t) = \delta(a(t-1), x(t)), y(t) = \lambda(a(t-1), x(t)), (t = 1, 2, \dots),$$

а для автоматов второго рода:

$$(2) \quad a(t) = \delta(a(t-1), x(t)), y(t) = \lambda(a(t), x(t)), (t = 1, 2, \dots).$$

При этом автоматы, закон функционирования которых определяется уравнением (1) называются автоматами Мили, а автоматы, определяемые уравнением (2), если у них функция выходов $y(t)$ не зависит от $x(t)$, – автоматами Мура.

Для структурных автоматов выбирают «более естественный способ отсчета времени» и тогда законы функционирования автоматов принимают следующий вид (подробнее см. [1]):

$$(3) \quad a(t+1) = \delta(a(t), x(t)), y(t) = \lambda(a(t), x(t)), (t = 0, 1, 2, \dots),$$

$$(4) \quad a(t+1) = \delta(a(t), x(t)), y(t) = \lambda(a(t)), (t = 0, 1, 2, \dots).$$

Для учета того, что реальные объекты не обладают мгновенной реакцией, рассматриваемые уравнения можно переписать следующим образом:

$$(5) \quad a(t+1) = \delta(a(t), x(t)), y(t+1) = \lambda(a(t), x(t)), (t = 0, 1, 2, \dots),$$

$$(6) \quad a(t+1) = \delta(a(t), x(t)), y(t+1) = \lambda(a(t)), (t = 0, 1, 2, \dots).$$

Теперь функции выходов уравнений (5) и (6) отражают задержку, возникающую между моментом анализа входной ситуации и реакцией на нее. Величина ее в данном случае определяется значением реального времени, соответствующего *единичному* дискретному такту автомата. В общем же случае подобная задержка будет кратна некоторому их числу.

Поскольку в контексте рассмотрения циклических цепей речь идет о множестве взаимодействующих объектов, то актуален вопрос о законе функционирования системы в целом. В контексте рассматриваемых проблем его достаточно свести к дискретному времени, которое может быть индивидуальным у каждого объекта или единым для схемы в целом.

Если за основу взято время, реализуемое единым источником синхронизации – часами, то в терминологии [4] подобной схеме соответствует модель *синхронизированных последовательностных схем* или просто *синхронных схем*, в противном случае – это модель *асинхронных схем*.

Анализ работы схем даже из небольшого числа элементов непростая задача. Для асинхронных систем вне единого времени сложность возрастает из-за неопределенности взаимодействия во времени компонент асинхронной системы. При наличии единого времени подобный анализ будет много проще. Именно модель с единым временем позволяет создать теорию без существенных ограничений на общность решаемых задач (см. аргументы в пользу единого времени, приведенные в [1]).

Далее будут рассматриваться, если не оговорено обратное, модели, компоненты которых работают в едином дискретном времени. Среди них также будем выделять модели синхронного и асинхронного видов, которые различать будем лишь по значению длительности дискретного такта механизма синхронизации

компонент. В силу этого на уровне теории (речь идет о формальных операциях рассматриваемых далее) они будут фактически неотличимы.

Легко видеть, что для схемы, представленной циклической цепью, как и любой другой, в рамках введенного закона функционирования автоматов и с учетом единого времени у компонент никогда не возникнет проблемы противоречивости сигналов. Хотя бы потому, что анализ сигнала на входах и последующее за этим изменение сигналов на выходах гарантированно никогда не пересекутся во времени.

Добавим, что определенное противоречие между идеализированным представлением автомата и существующей реальностью подмечено, например, в [7]. Однако, утверждается, что оно «при решении многих практически полезных задач оказывается несущественным». Но пример схемы генератора убеждает, это весьма существенная информация, которая подлежит обязательному учету.

Если в многокомпонентной модели не учтено свойство инерционности реальных систем, то это будет порождать, как можно видеть, структурное различие между моделью и ее реальным аналогом. Примером этого может служить и блочная структура, подобная *основной модели последовательностных схем* из [4].

Итак. Реальный генератор не нуждается в элементах памяти – триггерах, задержках и т.п. Аналогично отпадает в них необходимость для моделей, компоненты которых работают в соответствии с описанным выше законом функционирования и в едином дискретном времени.

6. Дизъюнктивная форма конечных автоматов

Определение абстрактного автомата в форме, имеющей ограничения на число входов/выходов, резко сужает возможности модели для рассмотрения композиций автоматов. Схему на рис.1 еще можно описать, но для более сложных связей абстрактная модель уже фактически не пригодна.

Существуют, правда, определения абстрактного автомата для объектов, имеющих множество входных и выходных кана-

лов. К ним относятся функциональные элементы, используемые при описании логических сетей [8]. Но, не имея формально каких-либо ограничений, они используются для описания элементарных автоматов с одним состоянием. Кроме того в более сложных случаях из-за кодирования состояний подобных абстрактных автоматов приводит к усложнению их описания в форме вектор-функций переходов и выходов [2].

Но известен и более абстрагированный от внутренней структуры формальной модели подход. В первом приближении он может быть представлен моделью, названной *последовательностной функцией* [9], или эквивалентным, но более формализованным понятием *автомата с абстрактным состоянием* [7]. И в том и в другом случае модели учитывают структуру входных и выходных каналов модели, оперируя однокомпонентным внутренним состоянием. Возьмем подобную формализацию за основу для дальнейших построений.

Описать функционирование элемента И-НЕ (см. рис.2) на словах можно кратко и просто. Оно может быть таким: единичное значение входов переводит выход элемента из любого внутреннего состояния в состояние нуля, нулевой сигнал на любом из его входов – из любого состояния в состояние единицы.

Автоматная модель, соответствующая данному описанию, показана на рис.3а (модель, на рис.3б соответствует модели инвертора). На графах двойной кружок обозначает *инициальное* (начальное) состояние автомата, а прочерки на месте выходных сигналов отражают отсутствие выходных сигналов на соответствующем переходе.

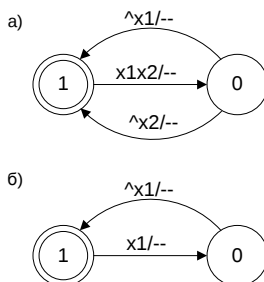


Рис. 3. Автоматные модели элемента И-НЕ (а) и инвертора (б)

Легко видеть, что внутренние состояния автоматов моделируют состояние выхода элементов. При этом для элемента И-НЕ единичное значение конъюнкции его входных сигналов определяет переход элемента в нулевое состояние, а отрицание любой из входных логических переменных – переход в единичное.

Назовем *дизъюнктивной нормальной формой конечных автоматов* (ДНФ КА) вполне (полностью) определенные автоматы с абстрактным состоянием, переходы которых помечены элементарными конъюнкциями логических переменных, соответствующих входным/выходным сигналам автомата.

Пусть модель ДНФ КА содержит только *результативные* переходы, т.е. переходы, помеченные выходными сигналами, или переходы с прочерком на месте выходных сигналов, но изменяющие текущее состояние автомата. Так мы сжимаем описание автомата до множества переходов, которые относятся к так называемому его *характеристическому множеству* [3].

Переходы, не включенные в описание автомата, представляют дополнение ДНФ КА до вполне определенного автомата. Это будет автомат, состоящий из изолированных состояний, содержащих переходы в виде петель с прочерком на месте выходных сигналов.

Далее компактность представления характеристического множества переходов ДНФ КА можно повысить за счет интервального свойства элементарных конъюнкций, помечающих условия срабатывания соответствующих переходов. Подобные конъюнкции покрывают *интервал* из 2^m переходов (см. подробнее [3]).

При этом *минимальный элемент* подобного интервала будет представлен переходом, помеченным конъюнкцией ранга m , в которую добавлены отсутствующие логические переменные входных каналов со знаком отрицания, а *максимальный элемент* – переходом, в котором эти же переменные представлены своим истинным значением. Заметим, что выходные переменные, определяющие выходные сигналы автомата рассматриваемого интервала переходов, имеют при этом одинаковое значение.

На смысловом уровне модель ДНФ КА отделяет явным образом «осмысленные» действия автомата от ситуаций, когда изменение информации на входах автомата не ведет к какому-

либо изменению, связанному со сменой внутреннего состояния автомата и/или изменению состояния его выходов. В остальном модель ДНФ КА – это обычный автомат, который соответствует понятию абстрактного автомата, но расширяющий его рассмотрением множества входных/выходных каналов.

В контексте рассматриваемых проблем модель ДНФ КА может быть вполне названа и абстрактным структурным автоматом. Или, когда это ясно из контекста, просто структурным автоматом, если это, конечно, не будет приводить к путанице с аналогичным термином из теории структурных автоматов.

Любую модель в форме ДНФ КА можно просто привести к эквивалентному классическому абстрактному автомату. Для этого необходимо восстановить множество переходов по его характеристическому множеству, а затем найти множества переходов (интервалы) для входных условий, представленных элементарными конъюнкциями ранга меньше чем m , где m – число входных каналов структурного автомата.

В результате получим вполне определенный автомат, условия переходов которого будут помечены конstituентами единиц (элементарные конъюнкции ранга m), которым останется поставить в соответствие символы входного алфавита соответствующего абстрактного автомата.

На рис.4 приведены модели эквивалентных абстрактных автоматов для структурных моделей на рис.3. Сравнивая модели на рис.3 и рис.4, можно оценить компактность представления [абстрактных] структурных автоматов в сравнении с классическими абстрактными автоматами.

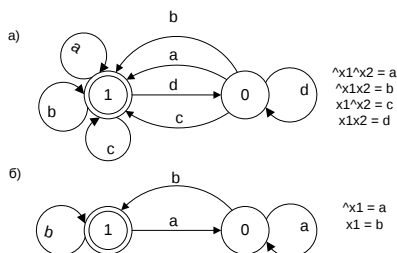


Рис. 4. Модели абстрактных автоматов элементов И-НЕ (а) и инвертора (б)

7. Сети из ДНФ КА

Для теории и практики большой интерес представляют разнообразные композиции из автоматов. Из них наиболее актуальна параллельная работа автоматов и, соответственно, подобная композиция автоматов.

Модель ДНФ КА в отличие от модели абстрактных автоматов более удобна и наглядна для отражения взаимодействия компонент для сетей из автоматов. Так, имея множество каналов, она может использовать одни из них для отражения связей между автоматами сети (схемы).

Кроме того, отождествляя состояния автомата с его выходными каналами, можно ввести новый вид синхронизации процессов на базе информации о текущем состоянии компонент. На рис.5 представлена сеть автоматов, моделирующая поведение схемы на рис.1, в которой использована синхронизация на базе информации о состояниях компонентных автоматов сети.

Рассматривая работу данной сети в соответствии с новым законом функционирования, представленным уравнениями (5) и (6) в едином времени, легко видеть, что автоматы, находясь первоначально в состоянии 0, на следующем такте перейдут синхронно в состояние 1 и т.д. Следовательно, из всего множества компонентных состояний сети «рабочими» будут два – компонентные состояния 000 и 111.

Таким образом, сеть на рис.5 ведет себя как генератор сигналов на рис.2. Но легко догадаться, что аналогично будет вести себя любая подобная сеть из нечетного числа автоматов-инверторов. Исходя из этого, схема на рис.2 может быть упрощена до схемы на одном элементе И-НЕ.

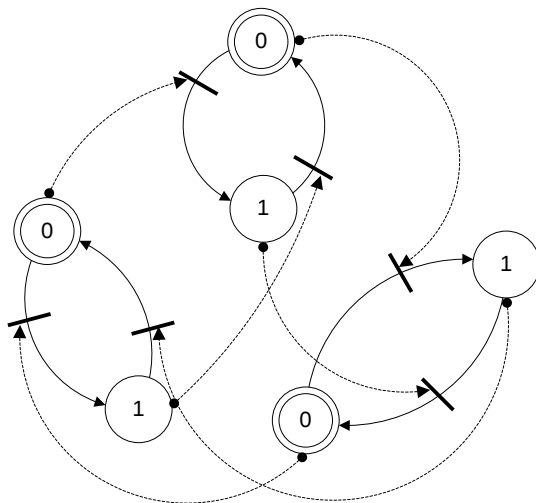


Рис. 5. Автоматная модель циклической цепи рис.1

8. Графический метод композиции автоматов

Теория структурных автоматов рассматривает построение схем для уже известной модели абстрактного автомата [2]. Но, если по какой-либо причине исходная модель не известна (см., например, рис.5 в [8]), то почти сразу возникает задача определения поведения схемы. Ее можно решить, найдя эквивалентный рассматриваемой сети (структурному автомату в обычном понимании) однокомпонентный автомат, который однозначно будет представлять алгоритм функционирования сети.

Для демонстрации нахождения результирующего автомата сети рассмотрим еще один пример сети из автоматов ДНФ КА. Сеть, которая является моделью RS-триггера на элементах И-НЕ, представлена на рис.6. Ее компонентными моделями являются модели элементов И-НЕ, показанные на рис.3. Из сравнения схемы триггера и структуры сети легко видеть, что каналы

$x1$ и $x2$ автоматов соответствуют установочным входам S и R триггера, а синхронизирующие связи, представленные пунктирными дугами, его перекрестным связям.

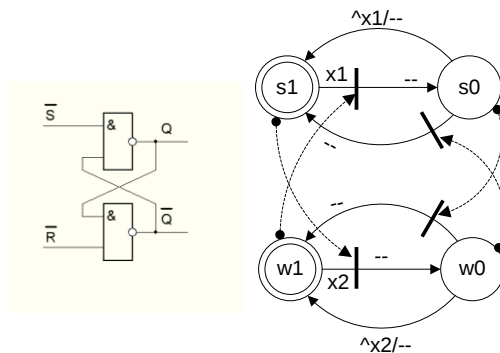


Рис. 6. Сетевая автоматная модель RS-триггера

Найдем по графу сети на рис.6 эквивалентный автомат и по нему определим соответствие работы исходной сети функционированию RS-триггера. Алгоритм его нахождения будет аналогичен графическому методу определения операции композиции автоматов по графу сети, представленному в [11].

Определим сначала множество компонентных состояний сети. Оно равно произведению состояний компонентных автоматов. Далее, рассматривая каждое компонентное состояние, будем анализировать переходы компонентных автоматов сети в ситуациях поочередной и одновременной работы автоматов.

У рассматриваемой сети множество состояний результирующего автомата представлено множеством компонентных состояний – $\{s0w0, s0w1, s1w0, s1w1\}$. Построим множество переходов, например, для состояния $s1w1$. Из анализа переходов компонентных автоматов следует, что всего будет три таких перехода: в состояние $s0w1$, когда «работает» автомат S , а W – «отдыхает»; в состояние $s1w0$, когда «работает» соответственно только автомат W ; в состояние $s0w0$ при условии, что изменят свои состояния оба автомата. В аналитической форме записи (подробнее о формах описания автоматов см. [6]) это можно представить так:

$$s1w1 = \{s0w1(x1 \wedge x2 / -), s1w0(\wedge x1 x2 / -), s1w1(x1 x2)\}.$$

Рассмотрим остальные компонентные состояния и выполним, если необходимо, минимизацию переходов автомата. Минимизация будет проводиться для логических функций, составленных из конъюнкций переходов, имеющих общее входное и выходное состояния и помеченных одинаковыми выходными сигналами. В результате получим автомат в форме ДНФ КА, который в аналитической форме записи будет выглядеть следующим образом:

(7) RS:

$$\begin{aligned} s0w0 &= \{s1w1(-)\}. \\ s0w1 &= \{s1w1(\wedge x1)\}. \\ s1w0 &= \{s1w1(\wedge x2)\}. \\ s1w1 &= \{s0w1(x1 \wedge x2 / -), s1w0(\wedge x1 x2 / -), \\ &\quad s1w1(x1 x2)\}. \end{aligned}$$

Граф результирующего автомата RS, представленного формулой (7), показан на рис.7.

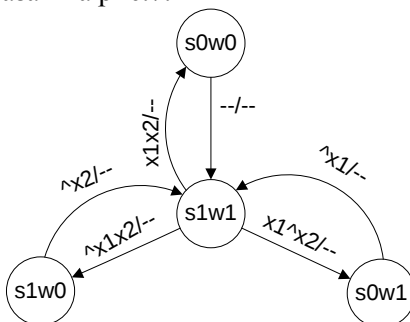


Рис. 7. Результирующий автомат сетевой модели RS-триггера

Таким образом, мы имеем: 1) некий автомат, представленный графом на рис.7 и 2) соответствующую ему «некорректную» схему RS-триггера на рис.6. Но поскольку найденный результирующий автомат эквивалентен графу работы RS-триггера, найденным другим путем, который описан в [10], то это может служить доказательство эквивалентности работы сетевой автоматной модели на рис.6 функционированию реального RS-триггера.

Теперь посмотрим на схему триггера с точки зрения существующей теории. Если реальный RS-триггер – «некорректная» по теории схема, то какими для автомата на рис.7, представляющего ее работу, будут выглядеть [корректные] схемы, построенные в соответствии с классическим методом канонического структурного синтеза автоматов?

Структурные автоматы, полученные в соответствии с аналогичным графическим методом синтеза автоматов (см. [11]), показаны на рис.8 и рис.9. На рис.8 представлен результат синтеза для структурного автомата на элементах задержки, а на рис.9 – на триггерах с отдельными входами.

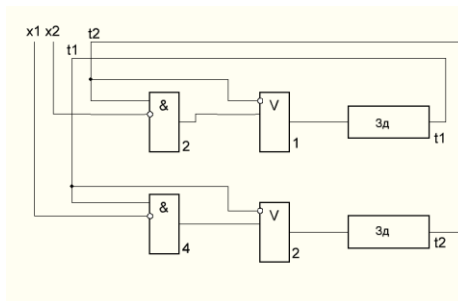


Рис.8. Структурная схема модели RS-триггера на задержках

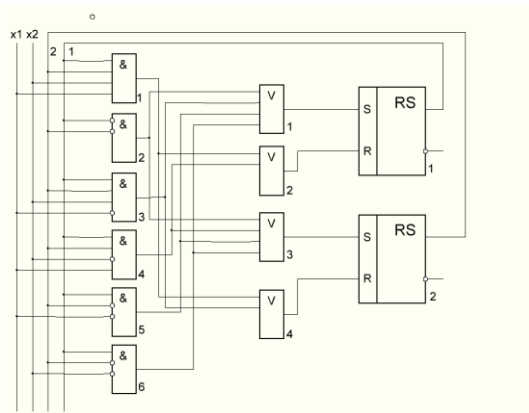


Рис.9. Структурная схема модели RS-триггера на триггерах с отдельными входами

Сравнение исходной схемы RS-триггера с его «корректными» реализациями выявляет, что схемы, полученные в результате формального синтеза, явно сложнее. При этом должно быть понятно, что по функционированию они эквивалентны, т.к. реализуют одну и ту же модель – автомат на рис.7.

Строго говоря, эквивалентные автоматы для схем на рис.8 и рис.9, если они будут найдены, реализуют не только соответствующий классический структурный автомат, но и автоматную модель, представленную сетью на рис.6. Однако, найти подобные автоматы для сетевых моделей данных схем путем парной композиции автоматов будет многократно сложнее, чем для сети на рис.6, являющейся моделью исходного RS-триггера.

Так, например, даже для относительно простой схемы на задержках (рис.8) число состояний системы, для которых будет строиться эквивалентный автомат, равно 64. Для схемы на триггерах (рис.9), даже если модели элементов будут содержать не более чем по два состояния, их число составит (по числу компонентных автоматов) уже 2^{12} (2 в степени 12).

Сравнив схемы, которые соответствуют теории, с эквивалентной схемой, используемой на практике, мы выявили их явное отличие. Но нельзя ли путем эквивалентных преобразований свести синтезированные схемы к реальной схеме?

Оказывается, что именно в рамках представленной концепции автомата, моделирующего инерционность реальных систем, можно, как минимум, схему на задержках свести к исходной схеме триггера. Во-первых, потому, что теперь модель не нуждается в элементах памяти, и, следовательно, из схемы на рис.8 их можно исключить. А, во-вторых, логические функции реализуемые парными элементами И и ИЛИ, путем эквивалентного преобразования легко сводятся к элементарной функции И-НЕ. В результате схема на рис.8 преобразуется в классическую схему триггера (рис.6).

Таким образом, мы получили доказательство большей адекватности введенной автоматной модели «некорректным» реальным объектам, включающих множество взаимодействующих компонент. В данном случае – на примере реальной логической схемы RS-триггера и, соответственно, ее сетевой автоматной модели, которые представлены на рис.6.

Последовательность шагов, положенных в основу обоснования приведенных выше аргументов, следующая. Во-первых, сначала для реальной схемы была предложена модель в форме автоматной сети. Далее для нее был найден эквивалентный автомат, который в точности совпал с моделью из независимого источника. Если принять, что последняя модель правильна, то, соответственно, корректна операция композиции и правильны рассуждения, приведшие к получению эквивалентной модели, показанной на рис.7.

Уже, казалось бы, приведенных аргументов достаточно для доказательства эквивалентности предложенной модели реальному объекту. Но, отталкиваясь уже от методов существующей теории, мы синтезировали для эквивалентной модели триггера структурные автоматы. Упростив один из этих структурных автоматов, но уже в соответствии с концепциями новой модели, мы пришли к схеме, совпавшей с точностью до элемента со схемой, которая некорректна по теории, но используется на практике.

Так мы получили не только доказательство корректности выполненных дополнительно преобразований на базе введенной операции композиции, но и, соответственно, дополнительные аргументы в пользу большей адекватности новой автоматной модели в сравнении с формальной моделью, не учитывающей инерционность реальных систем.

9. Графический метод декомпозиции автоматов

С точки зрения полноты операций операцию композиции автоматов необходимо дополнить парной операцией декомпозиции. Дадим определение также графического метода декомпозиции произвольного автомата на множество автоматов по графу исходного автомата. Сделаем это на примере автомата на рис.7.

Сразу отметим, что если операция композиции однозначна, то операции декомпозиции, напротив, не однозначна. Ее результат будет определяться выбором множества автоматов, на которые будет разлагаться исходный автомат, и кодированием его

состояний компонентными состояниями результирующей схемы (сети автоматов).

Предположим, для декомпозиции автомата на рис.7 решено взять два автомата по два состояния. Назовем эти автоматы S и W. Пусть кодировка состояний исходного автомата состояниями компонентных автоматов соответствует рис.7.

Декомпозиция, т.е. нахождение отображений структурных автоматов схемы, выполняется в соответствии со следующим алгоритмом: рассматриваются состояния компонентных автоматов, кодирующих текущее состояние исходного автомата, и для каждого компонентного автомата определяются переходы, исходя из анализа графа исходного автомата. Для тех переходов исходного автомата, которые изменяют состояния компонентных автоматов, составляются переходы компонентных автоматов, нагруженные входными и выходными конъюнкциями переходов исходного автомата, дополненных логическими переменными, совпадающими с именами состояний всех остальных автоматов декомпозиции.

Например, для компонентного состояния $s1w1$ у состояния $s1$ автомата S будут следующие переходы:

S:

$$s1 = \{s0(x1x2w1/-), s0(x1\wedge x2w1/-)\}.$$

для состояния $w1$ автомата W:

W:

$$w1 = \{w0(\wedge x1x2s1/-), w0(x1x2s1/-)\}.$$

После минимизации :

$$s1 = \{s0(x1w1/-)\}, \text{ в силу того, что:}$$

$$x1x2w1 \vee x1\wedge x2w1 = x1w1(x2 \vee \wedge x2) = x1w1.$$

Найдем переходы для остальных компонентных состояний. Для состояния $s0$:

$$s0 = \{s1(w0/-), s1(\wedge x1w1/-)\} = \{s1(w0/-), s1(\wedge x1/-)\},$$

т.к. после замены состояний $w0$ и $w1$ на эквивалентную логическую переменную, например, с именем $x2$, где $\wedge x2 = w0$, $x2 = w1$, придем к минимизации логического выражения:

$$\wedge x2 \vee \wedge x1x2 = \underline{x1\wedge x2 \vee \wedge x1\wedge x2 \vee \wedge x1x2 \vee \wedge x1\wedge x2} = \wedge x2 \vee \wedge x1 = w0 \vee \wedge x1.$$

В результате отображение автомата S и автомата W будут представлять сеть, которую в аналитической форме описания можно представить следующим образом:

(8) S :

$$s1 = \{s0(x1w1/-)\}.$$

$$s0 = \{s1(w0/-), s1(^x1/-)\}.$$

W :

$$w1 = \{w0(x2s1/-)\}.$$

$$w0 = \{w1(s0/-), w1(^x2/-)\}.$$

Легко видеть, что аналитическое описание автоматной сети, полученной в результате декомпозиции, совпадает с сетью, представленной графом на рис.6.

Итак, ранее был найден эквивалентный автомат, представленный графом на рис.7, который был получен с помощью операции композиции, примененной к некоторой исходной автоматной сети. Применив к данному автомату операцию декомпозиции, получили сеть, описываемую выражением (8), которая полностью соответствует сети, к которой была применена операция композиции. Следовательно, можно утверждать, что введенные операции композиции/декомпозиции, дополняя друг друга, входят в некую алгебру автоматов.

Совпадение сети, которая была источником для операции композиции, с сетью, полученной в результате декомпозиции, это, безусловно, дело «рукотворного» случая. Но, как должно быть понятно, в любом случае эти автоматы будут эквивалентны друг другу. В целом же, повторимся, операция декомпозиции неоднозначна, т.к. зависит во многом от параметров для декомпозиции – количества автоматов результирующей сети, множества их состояний и кодировки состояний исходного автомата состояниями компонентных автоматов.

10. Заключение

В работе предложен закон функционирования автоматов, моделирующий инерционность реальных систем. Он снимает ограничения, характерные для текущего состояния теории автоматов, представленных циклическими цепями. Предложена соответствующая этому закону организация сетей автоматов. В

ее рамках компонентные автоматы, работая в едином дискретном времени, сначала одновременно анализируют ситуацию на входах модели (или принимают символы) и лишь затем изменяют значение выходных сигналов (или выдают символы).

Введенная модель автомата с однокомпонентным абстрактным состоянием в дизъюнктивной форме – ДНФ КА и модель сетей на их базе позволяют значительно упростить создание дискретных моделей сложных систем. Логические схемы, соответствующие введенной сетевой модели, проще, чем аналоги, порождаемые каноническим методом синтеза цифровых схем, т.к. могут использовать более компактный (не включающий элементы памяти) структурно полный набор элементов.

Отметим, что исключение элементов памяти справедливо для случаев, когда не существенны гонки сигналов, связанные, например, со сменой состояний компонентными автоматами и/или когда нет необходимости применять структурные решения, характерные для классического канонического синтеза цифровых схем. Это относится, например, к использованию основной модели последовательностных схем.

Приведенные доказательства корректности «чистых» циклических цепей, т.е. не откорректированных в соответствии с существующей теорией, по существу определяет процедуру тестирования программных систем/сред логического и/или визуального проектирования. Здесь в качестве тестов выступают циклические цепи (см. приложение 1). При этом оценка результатов тестирования может базироваться на анализе результирующего автомата, найденного для сетевой модели циклической схемы применением операции композиции автоматов.

Одно из самых интересных и важных направлений развития предложенной модели – использование ее в качестве базовой модели управления для параллельных систем вообще и теоретической и практической базы параллельного программирования в частности. Именно рассмотренные качества модели были положены в основу технологии визуального параллельного программирования и ее реализации – среды ВКПа [12].

Литература

1. ГЛУШКОВ В.М. *Введение в кибернетику*. Изд-во АН Украинской ССР. К.: 1964. - 324с.
2. КУДРЯВЦЕВ В.Б., АЛЕШИН С.В., ПОДКОЛЗИН А.С. *Введение в теорию автоматов* – М.: Наука. Гл. ред. физ.-мат. лит., 1985. – 320 с.
3. МИЛЛЕР Р. *Теория переключательных схем, т.1 – Комбинаторные схемы*: Пер. с англ. – М.: Наука, 1970. – 416с.
4. МИЛЛЕР Р. *Теория переключательных схем, т.2 – Последовательностные схемы и машины* : Пер. с англ. – М.: Наука, 1971. – 304с.
5. БЕРКС А., РАЙТ ДЖ. *Теория логических сетей*. В кн.: Кибернетический сборник, №4., М., 1962.
6. МЕЛИХОВ А.Н. *Ориентированные графы и конечные автоматы*. – М.: Наука, 1971. – 416 с.
7. ЗАКРЕВСКИЙ А.Д. *Логический синтез каскадных схем*. – М.: Наука. Гл. ред. Физ.-мат. лит., 1981. – 416 с.
8. *Математическая энциклопедия*. – М.: Сов. энциклопедия, Т.1. Статья: «Автомат конечный», (с.52 – 67).
9. АНГЕР С. *Асинхронные последовательностные схемы*. – М.: Наука. Гл. ред. физ.-мат. лит., 1977. - 400 с.
10. ФРИКЕ К. *Вводный курс цифровой электроники. 2-е исправленное издание*. – Москва: Техносфера, 2004. – 432с.
11. БАРАНОВ С.И. *Синтез микропрограммных автоматов*. – Л.: Энергия, 1979. – 232с.
12. ЛЮБЧЕНКО В.С. *Автоматная парадигма параллельного программирования* //Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции (17-22 сентября 2012 г., г. Новороссийск). - М.: Изд-во МГУ, 2012. - 752 с. ISBN 978-5-211-06394-5.
13. АРМСТРОНГ ДЖ.Р. *Моделирование цифровых систем на языке VHDL*: Пер с англ./М.: Мир, 1992. – 175 с.
14. *Апериодические автоматы*. Под ред. В.И. Варшавского. – М.: Наука. Гл. ред. Физ.-мат. лит., 1976. - 424 с.
15. ДЬЯКОНОВ В.П. *VisSim+Mathcad+MATLAB. Визуальное математическое моделирование*. – М.: СОЛОН-Пресс, 2004. – 384 с.

16. ЛЮБЧЕНКО В.С. *Фантазия или программирование?*, Мир ПК, №10/97, с.116-119. URL: <http://www.osp.ru/pcworld/1997/10/116.htm> (дата обращения: 28.06.16).

11. Приложение 1

Циклические цепи, называемые часто обратными связями, алгебраическими петлями и т.п. оказывают существенное влияние на результаты программного моделирования логических схем. Можно даже утверждать, что сложно отыскать программный пакет, который бы корректно реализовал подобные схемы.

Таким образом, схемы с циклическими цепями служат хорошим тестом программных сред, ориентированных на проектирование логических схем. Но они применимы и в более широком смысле – для тестирования параллельных систем вообще. Хотя бы потому, что любая подобная схема представляет собой множество параллельно функционирующих и взаимодействующих между собой объектов/процессов.

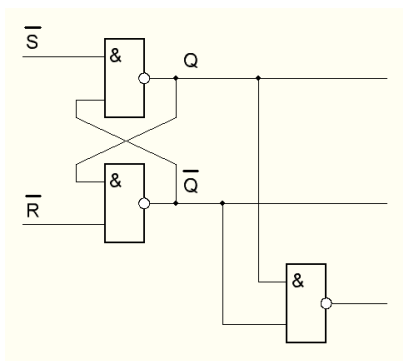


Рис.14. Схема индикации запрещенных состояний RS-триггера.

Рассмотрим схему, результаты работы которой не подлежат сомнению, в качестве теста для конкретных программных сред проектирования. Данная схема представлена на рис.14. Это RS-триггер на элемента И-НЕ, к выходам которого подключен

логический элемент И, который индицирует запрещенное состояние выходов RS-триггера в случае, если они, конечно, возникнут. А они обязательно возникнут, поскольку хорошо известно, что они неотъемлемое условие, например, переходов триггера между его устойчивыми состояниями.

Заметим, что данная схема не является искусственно надуманной. Она используется в схемотехнике асинхронных схем для индикации завершения процесса перехода триггера из одного состояния в другое – схема формирования *фазового сигнала* [14].

Тест 1. Моделирование RS-триггера в среде OWEN Logic.

Реализация схемы индикации запрещенных состояний RS-триггера в среде OWEN Logic (см. сайт www.owen.ru) представлена на рис.15. На рисунке отображен первый шаг пошагового моделирования работы схемы при переключении триггера из нулевого состояния в единичное. Легко видеть, что в данном случае его выходы уже находятся в запрещенном состоянии, что и отражает своим состоянием элемент-индикатор. На втором шаге, представленном на рис.16, триггер переходит в устойчивое состояние и соответственно выход индикатора принимает нулевое значение. Можно сказать, что результаты описанных шагов моделирования схемы правильно отражают работу RS-триггера.

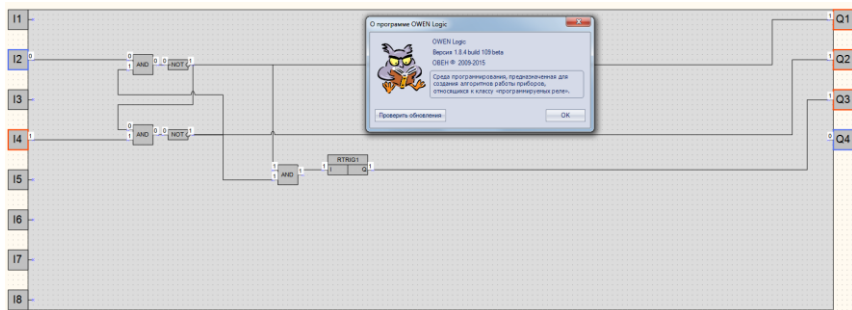


Рис.15. Шаг 1. Моделирование в среде OWEN Logic

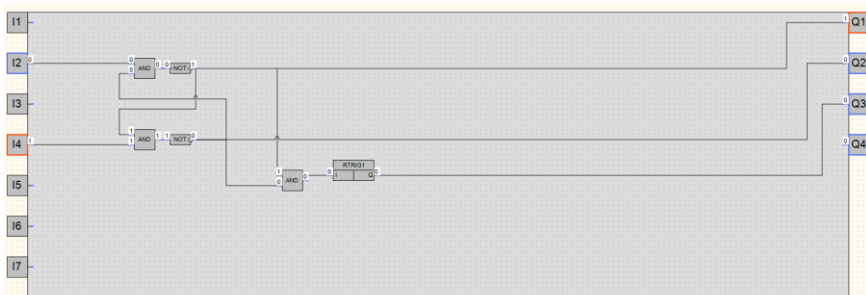


Рис.16. Шаг 2. Моделирование в среде OWEN Logic

Однако при другом эксперименте – моделировании переключения триггера из единичного состояния в нулевое выход индикатора будет все время в нулевом состоянии. А это уже ошибка в моделировании работы схемы!

Тест 2. Моделирование RS-триггера в среде KLogic.

Создадим аналогичную схему в среде KLogic (также представлена на сайте www.owen.ru). Ее вид показан на рис. 17. Результаты моделирования схемы при переключении триггера приведены на рис.18. Диаграмма сигналов на рис.18а отражает переключения триггера из нулевого состояния в единичное, а рис.18б – переход из единичного в нулевое. Видно, что импульс на выходе индикатора появляется при переходе в нулевое состояние. Это прямо противоположно результатам моделирования триггера в среде OWEN Logic, но в целом проблема та же – проблемы с моделированием переключения триггера.

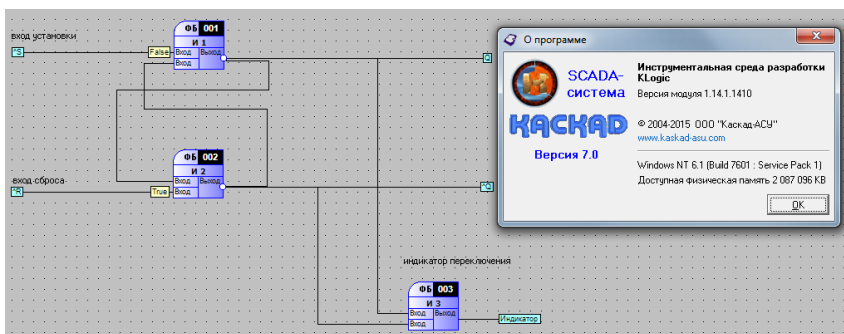


Рис.17. Модель RS-триггера триггера в среде KLogic

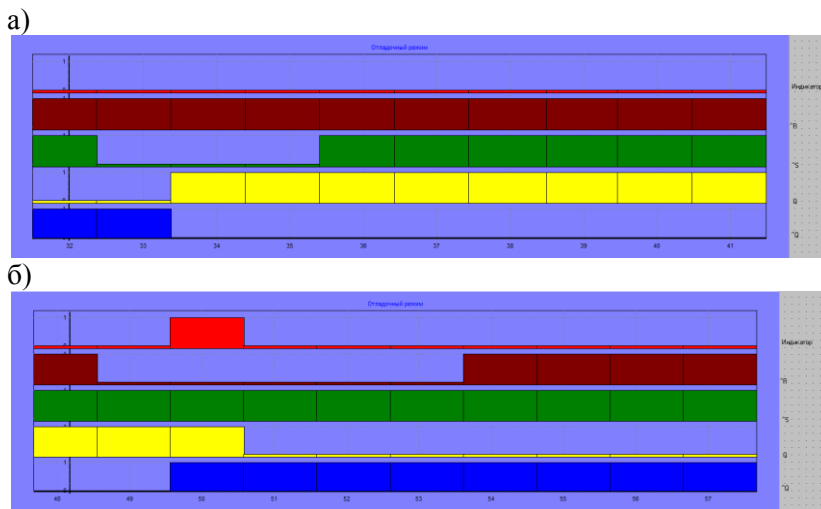


Рис.18. Результаты моделирования RS-триггера триггера в среде KLogic

Из тестирования модели RS-триггера в разных программных пакетах выясняется, что результаты противоречат друг другу. Так в какой же среде триггер моделируется правильно? Может быть, правильный ответ на заданный вопрос – ни в одной?!

Рассмотрим, возможно, более авторитетную среду проектирования CoDeSys, множество языков проектирования которой соответствуют международному стандарту МЭК-61131-3. Ориентированные на разный уровень подготовки и предпочтения пользователей они должны формировать проекты, которые, отличаясь языком проектирования, по идее должны – а как же иначе! – давать одинаковые результаты.

Тест 3. Моделирование RS-триггера на языке CFC в среде CoDeSys.

Протестируем CoDeSys для разных вариантов языков проектирования на примере рассматриваемой схемы. Выберем для начала язык CFC – язык непрерывных функциональных схем. Рис.18 демонстрирует проект на подобном языке.

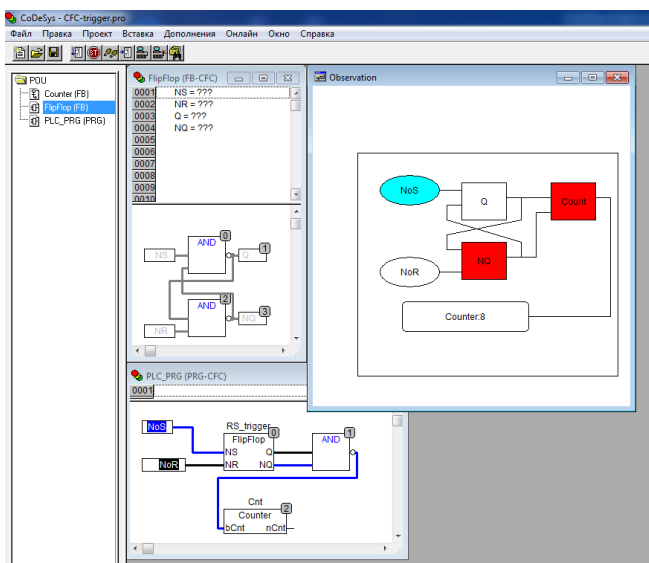


Рис.18. Моделирование в среде CoDeSys 2.3.9 на языке CFC

Тестирование демонстрирует все ту же проблему – зависимость состояния выходов триггера от режима переключения триггера. В данном случае запрещенное состояние возникает

только при переключении из единичного состояния триггера в нулевое.

Анализ результатов экспериментов убеждает, что источник проблем – в последовательном исполнении функциональных блоков, реализующих элементы схемы. Изменяя только последовательность их исполнения, можно получить разные результаты моделирования. Видимо, только такой особенностью сред и объясняется наличие у них режима, определяющего порядок исполнения блоков.

Тест 4. Моделирование RS-триггера на языке SFC в среде CoDeSys.

Но среди языков CoDeSys есть язык SFC – язык последовательных функциональных схем. С его помощью можно определять параллельное исполнение элементов схем. При этом предполагается, что параллелизм среды адекватно отражает параллелизм работы реальных объектов.

На рис.19 представлен проект на языке SFC. Но и для этого проекта тестирование приводит к аналогичным результатам. При этом для так называемых шагов параллельной ветви (на рис.19 шагами являются элементы *ine1*, *ine2*, *ine3*) нельзя определить порядок их исполнения. И это по идее следует считать правильным, если рассматривать именно параллелизм и, прежде всего, параллелизм реальный.

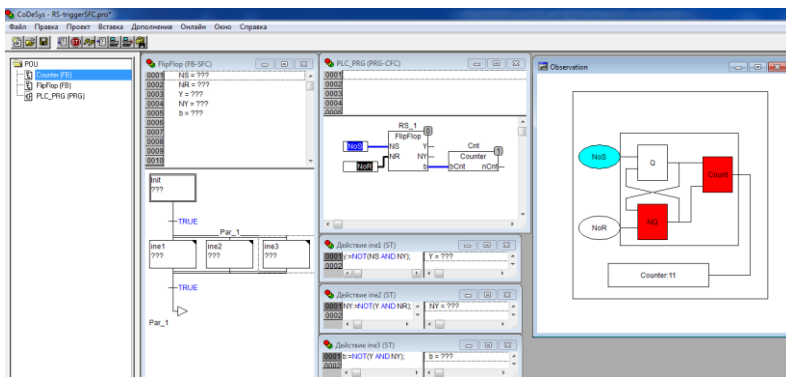


Рис.19. Моделирование в среде CoDeSys 2.3.9 на языке SFC

Однако, изменив имена шагов, мы получаем уже другой результат работы схемы. Т.е. ситуация аналогична процедуре изменения последовательности исполнения блоков, но только не путем их нумерации, а определяя лексикографический порядок имен шагов. Таким образом, можно утверждать, что параллелизм среды CoDeSys является, скорее, декларированным, чем реальным, при котором результаты работы блоков не зависят от последовательности их исполнения (одно из свойств параллелизма).

Тест 5. Моделирование RS-триггера в средах VisSym и MBTU.

Для предыдущих тестов не имело смысла проверять модель на способность входить в режим генерации. Для чего необходимо было бы сначала подать на входы триггера нулевые значения сигналов, переводящие выходы триггера в единичное состояние, а затем одновременно (!) установить на входах единичные значения. В соответствии с моделью (рис.7) триггер после этого должен войти в режим генерации, переключаясь между единичными и нулевыми состояниями своих выходов.

В средах MBTU (см. <http://mytu.power.bmstu.ru/>) и VisSym (см. [15], <http://www.vissim.com/>), которые имеют блоки, реализующие задержки, можно создать модели RS-триггера, отвечающие критериям правильных схем. Такие модели и результаты моделирования приведены соответственно на рис.20 и рис.21.

При этом среда MBTU предупреждает о наличии алгебраических петель, а среда VisSym на этот счет хранит молчание. Тем не менее, в обеих средах в случае равенства значений задержек триггер исправно входит в режим генерации, что в целом соответствует модели триггера на рис.7.

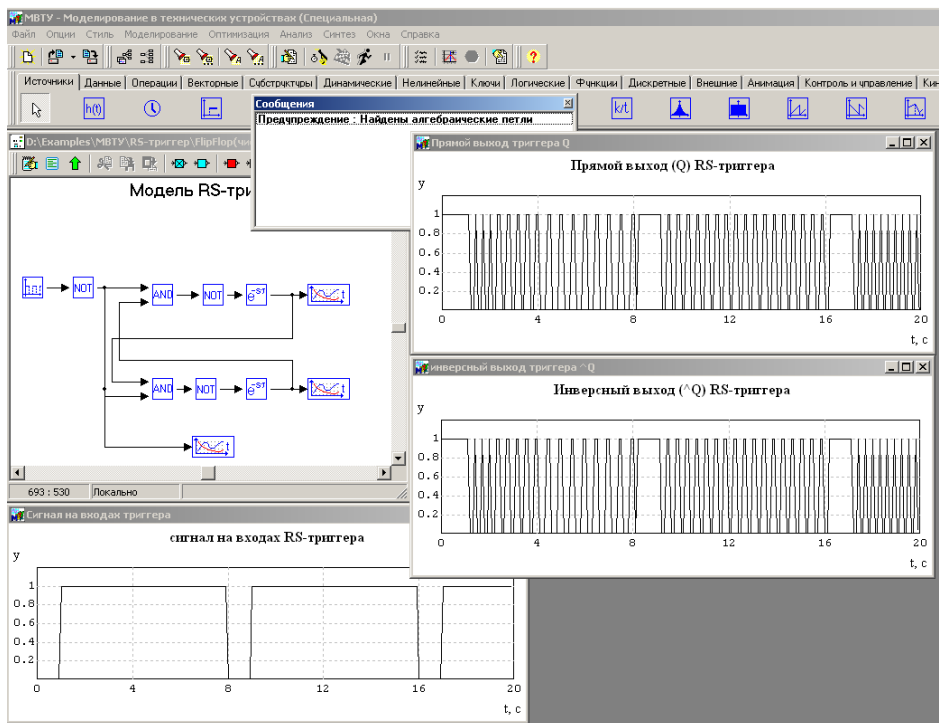


Рис.20. Модель RS-триггера в среде MBTU

Определим для моделей разные величины задержек. Это больше отражает реальную ситуацию, в которой невозможны элементы с абсолютно одинаковыми задержками. И если модель триггера в VisSym по-прежнему устойчиво «генерит» – рис.22, то модель триггера в MBTU будет «срывать» режим генерации – рис.23.

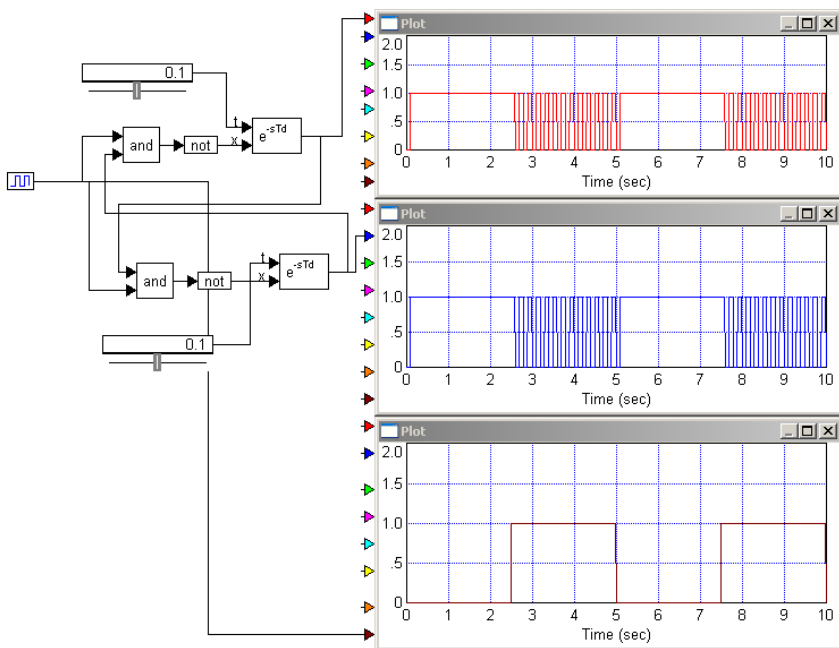


Рис.21. Модель RS-триггера в среде VisSym.

Результаты тестирования подталкивают к выводу, что реализация программного блока задержки в среде VisSym соответствует модели транспортного типа, в среде MBTU – инерционного (о типах задержек см. [13]).

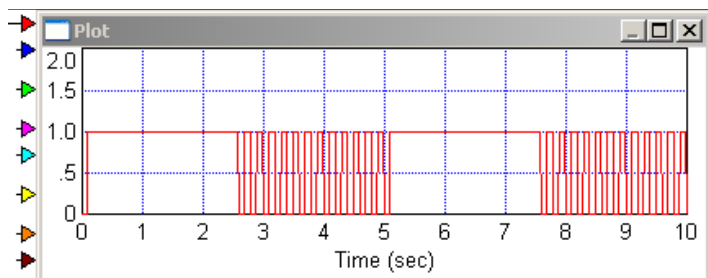


Рис.22. Режим генерации в среде VisSym.

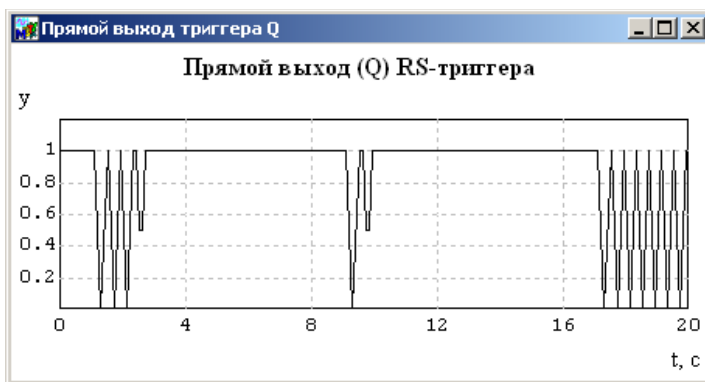


Рис.23. Режим генерации в среде MBTU.

Подводя итоги тестированию одной и той же схемы в рамках различных программных сред логического моделирования, можно говорить *о серьезных, а скорее – критических, проблемах реализации циклических схем современными программными средствами*. И это вполне обоснованно вызывает серьезную тревогу за работоспособность проектов, создаваемых в этих средах.

Тест 6. Моделирование RS-триггера в среде ВКПа.

Однако, если провести аналогичные эксперименты в среде ВКПа, то будут получены **достоверные** результаты работы не только для схем из идеализированных элементов (имеющих единичные задержки), но и для схем, включающих разные задержки – транспортного и инерционного типов. Более того, в ВКПа идеальный триггер или триггер с элементами, имеющими одинаковые задержки, можно вполне «вогнать» в режим генерации.

На рис.24 показана работа модели RS-триггера на элементах И-НЕ с транспортными типами задержек, имеющими разное значение у разных элементов. На рис.25 – на таких же элементах, но с задержками инерционного типа. Первая модель при подаче на входы нулей, а затем, спустя какое-то время, единиц

сразу входит в режим генерации. Вторая в подобный режим войдет лишь при определенных условиях – полной идентичности элементов триггера.

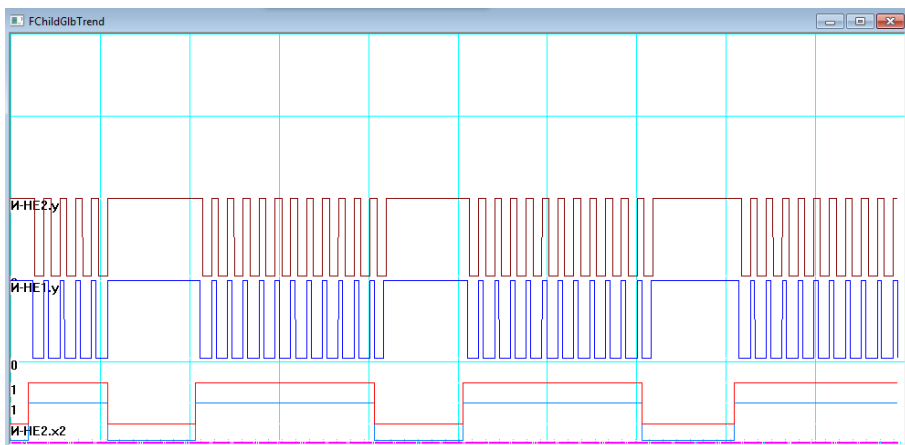


Рис.24. Тестирование RS-триггера с транспортным типом задержек в среде ВКПа.

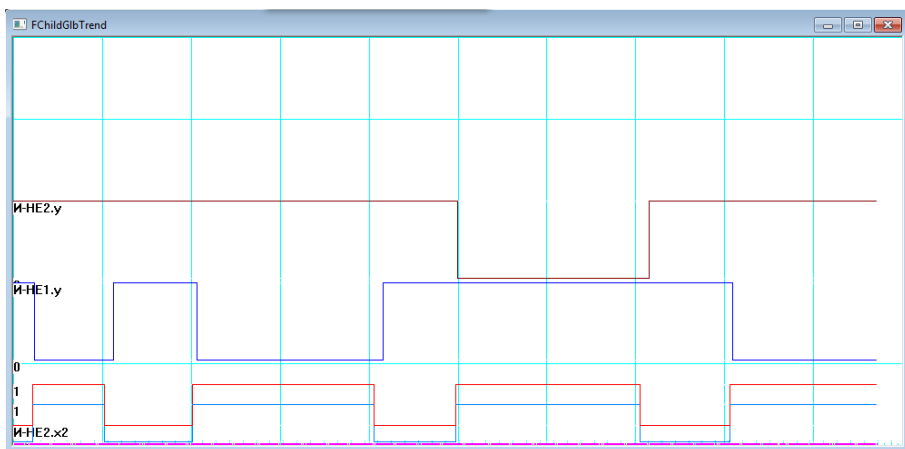


Рис.25. Тестирование RS-триггера с инерционным типом задержек в среде ВКПа.

Легко также видеть, что устойчивое состояние второй модели триггера определяется «плечом» с более быстрым логическим элементом. В данном случае, даже не зная величины задержек, можно точно сказать, что таким является элемент И-НЕ1 (см. график И-НЕ1.у рис.25). Но, как уже было сказано выше, если элементы триггера будут идентичны, то в режим генерации войдет даже триггер с инерционными задержками.

Таким образом, модель RS-триггера, созданная в среде ВКПа, правильно отражает работу реального триггера во всевозможных режимах его работы. В том числе и при переключении из запрещенного режима, куда его легко ввести, подав на его входы нули – для триггера на элементах И-НЕ, или единицы – в случае триггера на элементах ИЛИ-НЕ.

Подытоживая тему модели RS-триггера, можно сказать, что когда-то, а теперь уж – очень давно, модель триггера была предложена в качестве *одного из тестов*, проверяющих свойства параллелизма той или иной формальной модели, программной среды, языка программирования или программного продукта, претендующих на звание параллельных моделей и/или средств [16]. Но, к сожалению, за прошедшие годы ***проблема триггера*** не была не то, чтобы не решена (ее решение приведено выше), а так и не была осознана, ***как проблема, которая актуальна и поныне.***

Нельзя сказать, что за прошедшие годы что-то не изменилось. Теперь можно указать хотя бы несколько источников, в которых RS-триггеру уделяется достойное внимание пусть даже и «на пальцах». Один из них упомянут – это монография К. Фрике [10]. Основная же часть авторов уходит от детального анализа работы триггера. Ответом же оппонентам, которые не признают серьезность проблемы «элементарного триггера», являются приведенные выше результаты тестирования реальной схемы в реальных программных пакетах.

Формальные модели, программные пакеты и т.п. становятся все совершеннее, а реализовать «элементарный триггер» в них – все еще проблема! И проблема эта как была, так и осталась. Но это не «проблема триггера», как такового, а проблема наличия адекватной параллельной модели дискретных систем. И данную

проблему не устранила ни современная «новая» многоядерность, ни давно известная многопоточность.

Решение проблемы циклических цепей дает ответы на вопросы, какой же должна быть или может быть адекватная реальным системам их алгоритмическая параллельная дискретная модель и ее реализация. В контексте данной работы – это алгоритмическая модель с моделью управления в форме ДНФ КА автоматных сетей в едином времени и ее реализация в форме среды визуально-компонентного автоматного программирования – ВКПа [12].

ARTICLE TITLE

Viacheslav Lyubchenko, Institute of Control Sciences of RAS, Moscow, principal engineer (sllubch@mail.ru).

Abstract: In work it is offered to reconsider criterion of an estimation of a correctness of schemes of the structural automatic machines presented by cyclic chains. In these purposes the law of functioning of automatic machines is formulated, the model of the abstract disjunctive automatic machine is entered, networks from similar automatic machines are defined, graphic methods of realization of operations of a composition and decomposition of automatic machines are stated. A consequence of the received results was possibility of an exception of elements of memory from structurally full set of elements.

Keywords: cyclic chains, the law of functioning of automatic machines, the disjunctive structural automatic machine, automatic networks, operations of a composition and decomposition of automatic machines.

Статья представлена к публикации
членом редакционной коллегии ...заполняется редактором...

Поступила в редакцию ...заполняется редактором...
Опубликована ...заполняется редактором...