

# О проблемах RS-триггера (к решению одной параллельной задачи)

Вячеслав Любченко

Существует «темная сторона» работы самого простейшего из триггеров – RS-триггера. О ней мало кто знает. Связана она с проблемой запрещенных состояний триггера. Это те его состояние, в которых выходы триггера принимают одинаковое значение (две единицы или два нуля). Но кто или что запрещает триггеру в них находиться?

Но, во-первых, как говорится, если очень хочется и/или надо, то почему бы и не попробовать в них «загнать» триггер и посмотреть, что из этого получится (см., к примеру, схемы генератора на базе RS-триггера в [6])? А, во-вторых, многое случается помимо нашей воли, возникая подобно торнадо или «слепому дождю».

Так что же произойдет, если триггер в силу обстоятельств попадет в любое из своих запрещенных состояний? А, может, он просто не ведает про наши запреты и упорно в них стремиться? И есть ли гарантия, что триггер в них не попадет никогда?

На эти вопросы очень сложно получить ответы. Но далее мы на них ответим.

Также оказывается, что та «запретная зона», которую от нас скрывали и продолжают скрывать (в общем-то, надо признать, в силу объективных обстоятельств), выявляет достаточно серьезные проблемы, от которых мы не застрахованы. Но вот как их избежать, если, как выясняется, мы почти не знаем, как справляется с подобными проблемами простейший, но главнейший из всех триггеров (на нем основаны все остальные известные схемы триггеров) – RS-триггер?

## RS-триггер

Принципиальная схема RS-триггера известна хорошо. При этом чаще всего рассматривают схему на элементах И-НЕ (см., например, [1, 3]). Далее мы будем исходить из того, что эта же схема, по сути, представляет собой и структурную модель триггера, т.е. схему соединения неких абстрактных моделей реальных логических элементов. Последние (модели) преобразуют абстрактную же двоичную дискретную информацию (из «единичек» и «ноликов») в соответствии с реализуемой ими логической функцией.

Создавая модель реального триггера, нельзя не учитывать, что реальные элементы имеют задержки, которая не позволяют сигналу на выходе мгновенно изменяться, реагируя на изменения сигналов на входах. Задержку удобно измерять абстрактными дискретными интервалами, которым соответствует реальный временной интервал. Ту задержку, которая соответствует одному такому интервалу, называют *единичной*. И если, к примеру, единичная задержка равна двум наносекундам, то элементу, который имеет реальную задержку в десять наносекунд, будет соответствовать модель, имеющая модельную задержку, равную пяти единицам.

Заметим, что для триггера мы даже формально не имеем права считать задержку у его элементов равной нулю. Это следует из основ теории цифровых схем. Известно, что схемы, содержащие обратные связи, нельзя представлять в форме модели комбинационных схем (КС) (подробнее см. [2]). Поэтому модель логического элемента должна иметь как минимум единичную задержку (а возможно, что понятно, и больше).

## О формальной модели триггера в форме таблицы истинности

Чаще всего функционирование триггера описывают таблицей истинности (ТИ) (см. приведенную ТИ в [3] или определение RS-триггера в [3]). Заметим, что ТИ представляет в целом модель триггера (на уровне «черного ящика», имеющего два входных и два выходных канала) и соответствует нашему интуитивному представлению о его работе.

Хотя, как это часто бывает, и то (таблица) и другое (представление) не совсем верны. К примеру, если быть точным, то, исходя из схемы триггера, должно быть две ТИ. Таблица истинности логического элемента И-НЕ также известна (см., например, функцию  $f_{15}$  среди множества булевых функций от двух переменных в [2]). Но как объединить эти две таблицы в одну, и будет ли она соответствовать той таблице истинности триггера, которую обычно приводят? Из [4] следует, что сделать это можно. И при этом легко видно, что результирующая ТИ существенно отличается от «стандартной» [3].

Итак, уже при беглом анализе ясно, что ТИ триггера, которая обычно приводится и ассоциируется с его моделью, далеко не полностью отражает функционирование реального триггера. И, возможно, именно в силу этих обстоятельств пояснения о поведении триггера после его «насильственного» введения в

запрещенное состояние столь туманны. Обычно ссылаются на неясные обстоятельства, приводящие к случайному переходу в то или иное устойчивое состояние (см., к примеру, объяснение в [2]).

Но сразу возникает вопрос – почему триггер ведет себя так, а не иначе (не переходит, например, в другое свое запрещенное состояние (для схемы на ИНЕ это будет состояние выходов 00))? Из анализа ТИ это выяснить невозможно. И прежде всего потому, что ТИ ни как не учитывает временные процессы?

### **Конечно-автоматная модель триггера**

Но у нас есть модель, которая моделирует и реальное время. Это модель конечного автомата (КА). Модель в форме КА для логического элемента И-НЕ, имеющая единичную задержку, может быть представлена следующим автоматом S (ср. с описанием функции f15 в [2]):

$$\begin{aligned} S: \\ s1 &= \{s0(x1x2/-)\}, \\ s0 &= \{s1(^x1/-), s1(^x2/-)\} \end{aligned}$$

В данном случае мы использовали аналитический способ задания автоматов (подробнее о способах описания автоматов см. в [5]).

Имея автоматную модель элемента И-НЕ, можно теперь создать и модель триггера. Это будет автоматная сеть RS, состоящая из двух автоматов Q и W, т.е.

$$\begin{aligned} RS: \\ Q: \\ q1 &= \{q0(x1w1/-)\}, \\ q0 &= \{q1(^x1/-), q1(w0/-)\}. \\ W: \\ w1 &= \{w0(x2q1/-)\}, \\ w0 &= \{w1(^x2/-), w1(q0/-)\}. \end{aligned}$$

Здесь Q и W модели элементов триггера (автоматы S). Только у данных автоматов, чтобы отразить связи между ними, вместо соответствующих входных сигналов проставлены состояния автоматов, которые связаны с этими входами.

### **Эквивалентная модель триггера**

Модель триггера в форме ТИ представляет собой эквивалентную, т.е. без учета внутренней структуры триггера, модель. Как мы уже выяснили она не правильная (за исключением, может быть, ТИ созданной в [4]). А какова будет эквивалентная автоматная модель триггера?

Если применить к автоматам Q и W операцию умножения автоматов (о схемах соединения автоматов и операциях над ними см. в [5]), то можно получить автомат  $RS=Q \otimes W$ , который и будет подобно ТИ эквивалентной моделью триггера. В данном случае это следующий автомат:

$$\begin{aligned} RS: \\ q1w1 &= \{q0w0(x1x2/-), q0w1(x1^x2/-), q1w0(^x1x2/-)\}, \\ q0w1 &= \{q1w1(^x1/-)\}, \\ q1w0 &= \{q1w1(^x2/-)\}, \\ q0w0 &= \{q1w1(-/-)\}. \end{aligned}$$

После очевидных переименований в более компактной форме это будет автомат  $F = G \otimes W$ :

$$\begin{aligned} F: \\ f1 &= \{f4(x1x2/-), f2(x1^x2/-), f3(^x1x2/-)\}, \\ f2 &= \{f1(^x1/-)\}, \\ f3 &= \{f1(^x2/-)\}, \\ f4 &= \{f1(-/-)\}. \end{aligned}$$

### **Поведение триггера**

Но в отличие от таблицы истинности, эквивалентная автоматная модель намного точнее описывает поведение триггера. И, например, из анализа автомата F следует, что у него нет прямых переходов из одного устойчивого состояния в другое: любые переходы осуществляются через промежуточное

состояние  $f_1$ , которое и есть то самое «запрещенное состояние» (единичное состояние его выходов) триггера!

Таким образом, из анализа автомата  $F$  следует, что

1. Как бы мы тому не противились, но триггер хоть какое-то время, но обязательно будет находиться в своем запрещенном состоянии.
2. Триггер может быть генератором, т.к. есть цикл между состояниями  $f_1$  и  $f_4$ , т.е. переходы между двумя (!) его «запрещенными состояниями».

Первый вывод легко подтверждается анализом работы реального триггера, а вот второй – генерацию! – выявить сложно. Но, может, при определенных условиях и это возможно? Или почему генерация не возникает у реального триггера? Что нужно сделать, чтобы реальный триггер, как и его модель, стал генератором?

### **Введение задержек в элементы триггера**

Мы уже согласились с тем, что элементы триггера имеют задержки. Кроме того, они могут быть разными у элементов. А еще они могут быть разными по фронту импульса (переход сигнала из 0 в 1) и спаду (из 1 в 0) импульса. Может такое разнообразие вариантов и влияет на поведение триггера, «срывая» его генерацию?

Модель элемента И-НЕ, содержащая произвольные задержки, может быть такой:

S:

$$\begin{aligned} s_1 &= \{s_0(x_1x_2/y_0)\}, \\ s_0 &= \{s_1(^x1/y_1), s_1(^x2/y_1)\} \end{aligned}$$

Здесь  $y_0$  и  $y_1$  – вложенные автоматы, которые задерживают на заданное число тактов переход из одного состояния, например,  $s_1$ , в другое –  $s_0$ .

Но можно создать и другую модель элемента И-НЕ с задержками. Без иерархии автоматов. Это, к примеру, соединение двух параллельных автоматов. Из них один реализует «идеальный» элемент И-НЕ (с единичной задержкой), а второй - задержку выходного сигнала (состояния) первого автомата. Если это соответственно автоматы  $S$  и  $Z$ , то модель элемента И-НЕ - автомат  $M$  - будет представлена их параллельным соединением, где  $M = S||Z$ . При этом автомат  $Z$  может иметь следующий вид:

Z:

$$\begin{aligned} z_1 &= \{z_0(s_0/y_0)\}, \\ z_0 &= \{z_1(s_1/y_1)\}. \end{aligned}$$

Здесь  $y_0$ ,  $y_1$  – уже упомянутые выше вложенные автоматы-задержки. Заметим также, что автомат  $S$  в этом случае принимает свою исходную форму, т.е. не содержит вложенных автоматов.

### **Генерация триггера с задержками**

Формальный анализ и программная реализация рассмотренных выше моделей убеждают в том, что при любом соотношении задержек, как для отдельного элемента И-НЕ (по фронтам его сигналов), так и для отдельных элементов триггера, генерация в триггере должна быть! И это даже несмотря и на то, что есть определенные отличия в поведении моделей триггеров, базирующихся на разных моделях элементов.

Последнее – разные модели элементов И-НЕ - само по себе тоже интересно. В этом случае двухкомпонентная модель элемента имеет в отличие от однокомпонентной модели минимум двойную задержку (одну вносит модель И-НЕ, вторую – модель задержки). Двухкомпонентная модель имеет и большее число состояний, что влияет и на вид эквивалентного автомата самого элемента (И-НЕ) и соответственно эквивалентного автомата системы (RS-триггера) в целом.

Программное моделирование показывает устойчивую генерацию, но реальный-то триггер не генерит! Можно ли его вообще как-то заставить его войти в режим генерации, если уж формальная модель убеждает в том, что это возможно?

### **Об аппаратной реализации генератора на базе RS-триггера**

Автор, задавшись целью подтвердить теоретические выводы, решил собрать генератор на базе триггера. Казалось бы, для этого достаточно в элементы триггера искусственно ввести достаточно большие задержки, на фоне которых другие факторы (их собственные задержки, задержки в проводах, входные емкости и др.) были бы пренебрежимо малы. И один из способов реализации этой идеи – задержка на основе RC-цепочки.

На сайте RADIOMAN (<http://www.radioman.ru>) приводятся разные схемы генераторов на логических элементах. В их числе и генератор на базе RS-триггера (см. [6]). Но это не те схемы, которые подтверждают наши теоретические выводы. Однако там есть схема задержки на RC-цепочках [7], которую можно попробовать использовать в эксперименте.

Созданная схема и эксперименты над ней показали полное нежелание триггера входить в режим генерации даже при очень больших задержках! В чем дело? Оказывается, что все дело в том, что есть разные виды задержек. Точнее, в теории и практике моделирования цифровых схем различают задержки транспортного типа и инерционные задержки (подробнее см. в [8]).

Используемая в реальном эксперименте задержка – задержка инерционного типа. Для того, чтобы она сработала, т.е. передала входной сигнал, необходимо зафиксировать сигнал на ее входе на определенное время. В триггере же получается, что более быстрое его плечо срывает передачу задержкой входного сигнала (речь идет о передаче «нуля», т.е. спада импульса, формируемого «идеальным» элементом И-НЕ).

А так как все сходится к тому, что реальный элемент содержит в себе задержку скорее инерционного типа, то становится понятно, почему реальный триггер ни как не входит в режим генерации. Да, скорее всего, его в этот режим загнать и невозможно! Но с другой стороны, если иметь задержку транспортного типа, то триггер должен стать генератором. И такую задержку вполне можно создать (см. о цифровом способе реализации задержки [7]). И вот после этого, наверное, можно будет наслаждаться процессом генерации триггера.

### ***О моделях транспортной и инерционной задержек***

Эксперимент показал важность разделения задержек на два типа. Осталось только подтвердить эти выводы на их программных моделях. Такие автоматные модели транспортной и инерционной задержек были созданы и введены в двухкомпонентную модель элемента И-НЕ на место автомата Z (см. выше). Программное моделирование с разными типами задержек полностью подтвердило выводы, сделанные во время реального эксперимента. Модель триггера на основе моделей элементов, содержащих задержки инерционного типа, работает точно так же, как и реальный триггер. В то же время моделирование показало, что и с инерционной задержкой возможна генерация. Она может быть кратковременная (при небольшом отличии величин задержек), а при определенных значениях возможна даже устойчивая генерация. Например, при полном совпадении значений задержек, но также может быть и при разных значениях. В объяснение последнего можно лишь сказать, что для более точного программного эксперимента необходимо иметь и более точную модель инерционной задержки (в данном случае она достаточно проста). Вот тогда можно дать и более точный ответ по поводу данных эффектов (кратковременной или устойчивой генерации) и их присутствия в реальной схеме.

### ***Выводы***

Задача моделирования триггера уже достаточно давно была предложена автором в качестве теста на параллелизм программных средств[9]. Представленные выше результаты, доказывающие невозможность генерации у реального триггера, ни сколько не опровергают утверждение, что «в параллельной среде, имеющей трехмерные свойства, триггер при определенных условиях будет входить в режим генерации, в двумерной же - нет.»[9]. Любая параллельная модель триггера, независимо от наличия или отсутствия задержки, будет и должна входить в режим генерации. Только для случая с задержками инерционного типа для этого нужно создать и зафиксировать определенные значения задержек. И модель обязана генерить, т.к. в ней «тепловых шумов» и др. свойств (см. [1]), что присуще реальным элементам триггера (хотя ввести моделирование таких процессов в модель элемента ни что нам не мешает).

Программное моделирование с разными типами задержек полностью подтвердило выводы, сделанные во время реального эксперимента. Поэтому в определенном смысле по задаче триггера можно подвести итоги:

1. Наконец-то на вопрос может ли RS-триггер войти в режим генерации теперь можно дать достаточно аргументированный ответ. Он состоит в том, что реальный триггер – не может! Причина – инерционный тип задержек у элементов триггера.
2. Скорее всего, в силу п.1., реальный триггер так никогда и не попадет в свое «нулевое состояние».
3. Можно дать ответ и на вопрос в какое состояние переключится триггер из своего запрещенного состояния. Он перейдет в то, которое определяется элементом, имеющим меньшую задержку.
4. Но вероятность, что триггер может стать генератором, остается. Во-первых, это возможно при полной идентичности элементов. Во-вторых, если кто-то по какому-то умыслу вставит в него задержку транспортного типа (а почему бы и нет?).

5. Но более важный вывод состоит в том, что на примере решения задачи моделирования триггера выявляется качественное отличие в поведении программных (абстрактных) систем и реальных систем. По умолчанию программные системы имеют задержки транспортного типа, а реальные – инерционные. А раз так, то понятно, почему, с одной стороны, сложно заставить реальный триггер войти в режим генерации, а программную модель, с другой стороны, – «сорвать» генерацию.
6. Но еще важнее то, что эксперименты и с реальным триггером и с его программной моделью служат убедительным доказательством того, что при параллельном программировании обязательно необходимо учитывать временной фактор. Любая параллельная программа – это сложная система, состоящая подобно триггеру из множества взаимодействующих компонент. И лучше, если «время» будет заложено уже в алгоритмическую модель (язык программирования). Подобно тому, как это сделано на формальном уровне у модели конечного автомата.

То, о чем было сказано выше, можно «пощупать» на созданной автором автоматной модели триггера. В ней можно «на лету» изменять задержки у любого из элементов триггера, изменять их значение, как и по фронту и по спаду сигнала, выбирать разные автоматные модели элементов И-НЕ (однокомпонентную, двухкомпонентную, с инерционными или транспортными задержками). При этом в процессе работы модели выводятся диаграммы сигналов (на входе и выходе триггера, а также выходной сигнал дополнительного элемента И-НЕ, подключенного к выходам триггера). В процессе работы модели протоколируются текущие состояния автоматных моделей элементов. Пример окна диалога и графического вывода программы (осциллограммы сигналов) показан на рис. 1.

### **Литература:**

1. Д.Гивоне, Р.Росер. Микропроцессоры и микрокомпьютеры. Вводный курс. М.: Мир, 1983. – 463 с.
2. Проектирование цифровых вычислительных машин. Под ред. С.А. Майорова. Учебное пособие для студентов вузов. - М.: «Высшая школа», 1972. – 344 с.
3. Толковый словарь по вычислительным системам/Под ред. В. Иллиноута и др.: Пер. с англ. А.К. Белоцкого и др.; Под ред. Масловского. – М.: Машиностроение, 1991. – 560 с.
4. Шалыто А.А. SWITCH-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998. 628 с.
5. Мелихов А.Н. Ориентированные графы и конечные автоматы. – М.: Наука, 1971. – 416 с.
6. Генераторы импульсов. Автогенераторы. Мультивибраторы.  
[http://www.radioman.ru/shem/other/impuls\\_generators.shtml](http://www.radioman.ru/shem/other/impuls_generators.shtml)
7. Задержка импульсов. [http://www.radioman.ru/teoria/mop\\_zaderjka.shtml](http://www.radioman.ru/teoria/mop_zaderjka.shtml)
8. Армстронг Дж.Р. Моделирование цифровых систем на языке VHDL: Пер с англ./М.: Мир, 1992. – 175 с.
9. Любченко В.С. Фантазия или программирование? “Мир ПК”, №10/97, с.116-119.  
<http://www.osp.ru/pcworld/1997/10/116.htm>

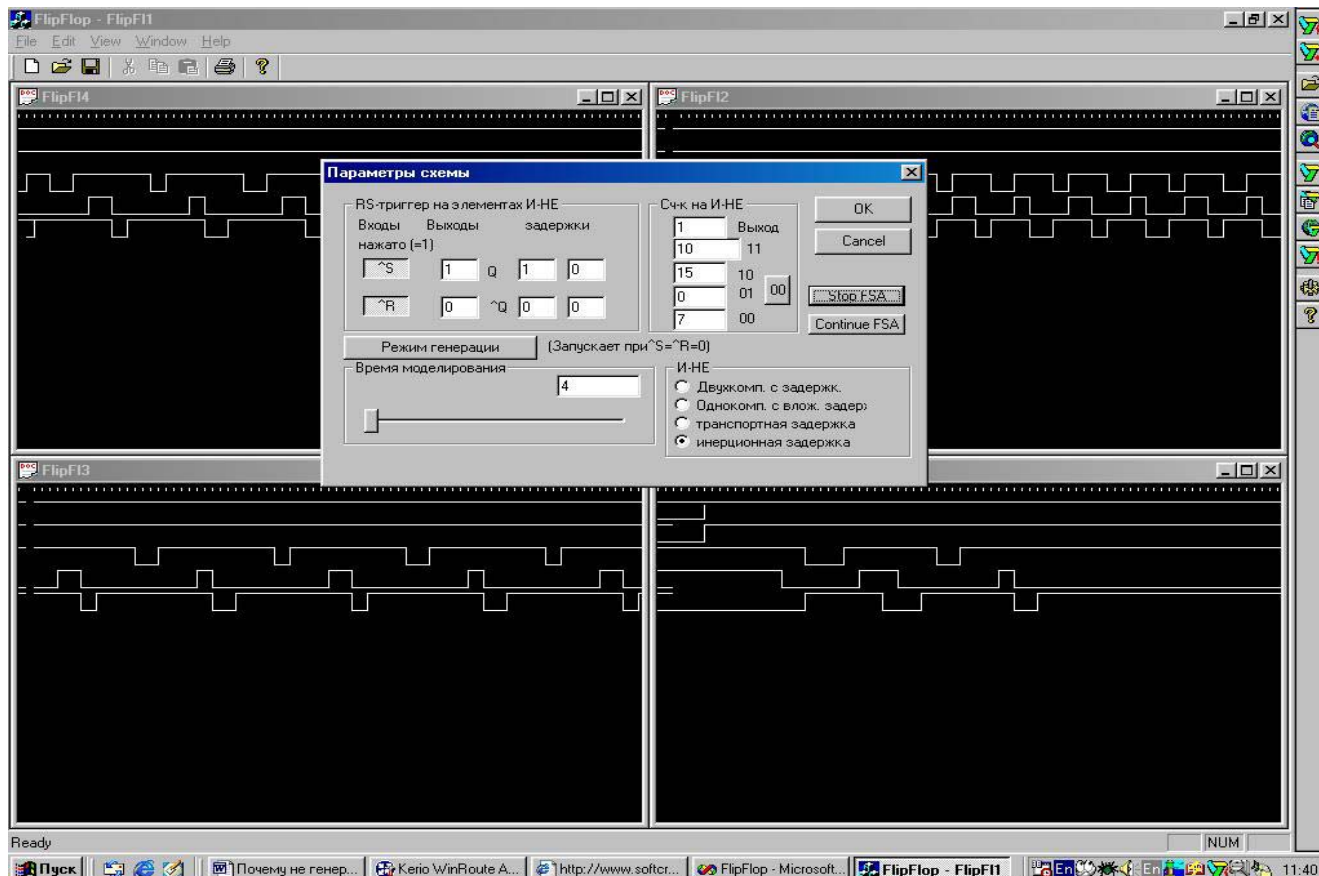


Рис. 1. Отображение результатов моделирования различных моделей RS-триггера.

**Комментарий к диаграмме.**

Левые окна отражают странность - «вложенность» сигналов в поведении триггера на базе двухкомпонентной модели и модели с транспортной задержкой. Верхнее правое окно представляет диаграмму триггера с однокомпонентной моделью элемента И-НЕ, имеющую у одного из элементов задержку по спада сигнала (у элемента ^S). На нижнем справа окне «заморожен» процесс срыва генерации у триггера с инерционной задержкой (виден небольшой колебательный процесс!).

**Примечание.** Диалог настройки модели вызывается нажатием правой кнопки мыши в пределах соответствующего окна отображения. Параметры сразу же вступают в действие при перемещении фокуса ввода из поля задания параметра. С помощью «ползунка» можно управлять временем моделирования, задавая длительность одного такта модельного времени.