

Состязания сигналов в цифровых устройствах

На текущий момент цифровые устройства рассматривались только с точки зрения функций, которые они выполняют. Процессы *изменения* состояний цифрового устройства, которые происходят в процессе функционирования этих устройств, не учитывались. Считалось, что они мгновенные.

Физические причины состязаний сигналов

Тем не менее, в реальных цифровых устройствах присутствуют такие моменты, как конечная длительность фронта и среза импульсов, ненулевое время распространения сигнала между элементами микросхемы и ненулевая задержка сигнала в логических элементах. Другими словами, рассматривавшиеся ранее цифровые устройства представляли из себя идеальные модели, при анализе работы которых не учитывались временные параметры логических элементов и структура реальных сигналов внутри микросхемы.

Например, время распространения сигнала между таблицами истинности FPGA ненулевое и определяется технологическим процессом, а также расстоянием между конкретными таблицами в микрочипе.

На рис. 1. Схематично представлены примеры неидеальности фронтов/срезов реальных импульсов в цифровых устройствах, а также задержки срабатывания логических элементов.

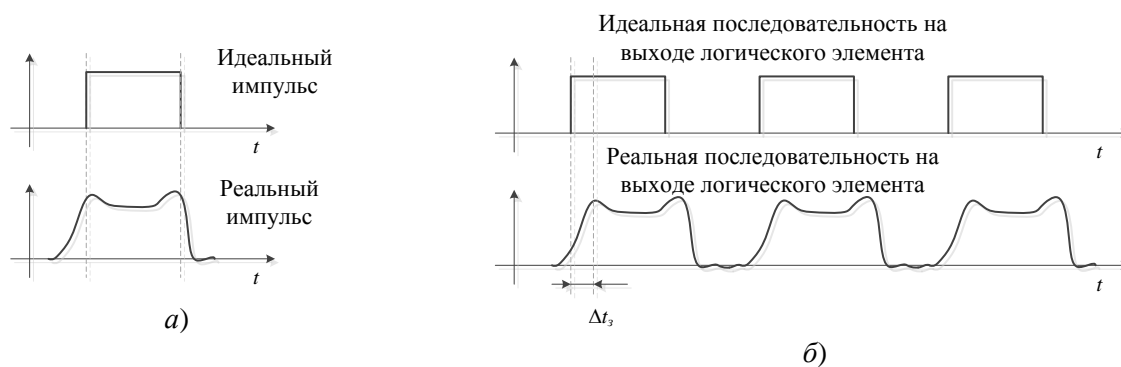


Рис. 1. Примеры неидеальности фронтов/срезов импульсов сигнала (а) и его задержки на выходе логического элемента (б)

Наличие задержек в компонентах микросхемы, разброс ее величины, неидеальность формы импульсов приводят к тому, что во время переходных процессов цифровое устройство функционирует не в соответствии с Булевой алгеброй, не в соответствии с тем алгоритмом, которое оно должно реализовывать. В этом случае говорят, что в устройстве наблюдаются состязания сигналов.

Состязания сигналов – это явление в цифровых устройствах, при котором вследствие переходных процессов устройство функционирует не в соответствии с алгеброй логики.

На рис. 2 показан пример состязаний, возникающий в логических элементах И и ИЛИ, обусловленный конечными длительностями фронта и среза импульсов. Видно, что в процессе смены полярностей входных сигналов на выходе устройства возникают импульсы, обусловленные конечными длительностями фронтов и срезов.

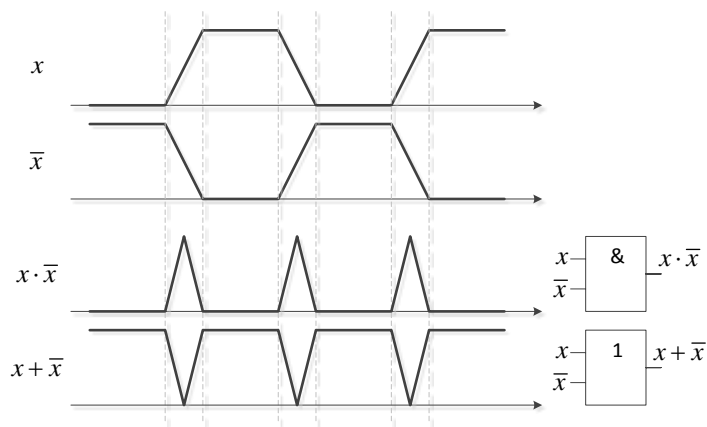


Рис. 2 Пример состязаний в элементах И и ИЛИ

Для иллюстрации состязаний, обусловленных конечным временем срабатывания логических элементов, каждый логический элемент удобно представлять в виде двух компонентов: первый выполняет заданную логическую функцию и является безынерционным, а второй включен за первым и является элементом задержки. Рассмотрим, например, логическую функцию вида $y = \overline{x_1 + x_2 + x_3}$ (рис. 3).

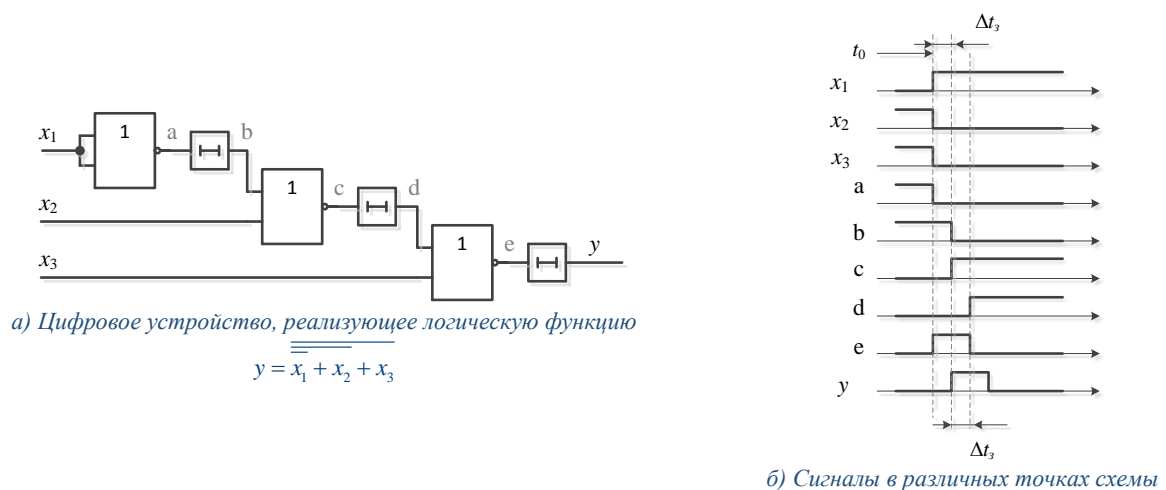


Рис. 3 Пример состязаний, обусловленных конечным временем срабатывания логических элементов

На рис. 3 начальное значение входных сигналов следующее: $x_1 = 0, x_2 = x_3 = 1$. Пусть далее в некоторый момент времени t_0 значения сигналов меняются на противоположные: $x_1 = 1, x_2 = x_3 = 0$. В случае идеальных логических элементов для обеих комбинаций входных сигналов $y = 0$. В случае же реальных логических элементов, обновление выходных сигналов после изменения входных будет происходить с некоторой задержкой Δt_3 . При этом сигнал x_3 поступает на вход последнего сумматора мгновенно, а результаты обработки сигналов x_1 и x_2 — с некоторой задержкой, в соответствии с временными диаграммами на рис. 3,б. В результате в схеме происходят состязания между сигналами, а на выходе — появляется импульс, которого в соответствии с алгоритмом работы быть не должно.

Виды состязаний

Состязания сигналов делятся на *опасные* и *неопасные*. И первые, и вторые приводят к возникновению на выходе устройства изменения уровня сигнала, не предусмотренного алгоритмом работы устройства. Состязания называются опасными, если обусловленные ими

неалгоритмические переходы изменяют работу последующих устройств. В противном случае, т.е. если работа последующих устройств не нарушается, состязания называются неопасными.

Опасные состязания в зависимости от места возникновения делятся на состязания между изменениями сигналов *на входах* устройства и состязания между сигналами *в логических элементах* схемы.

В зависимости от условий перехода одного комбинационного устройства из одного состояния в другое различают **статические** и **динамические** состязания. Статические состязания в свою очередь делятся на единичные (1-состязание) состязания и нулевые (0-состязание).

Когда при последовательном переходе между парой состязаний сигнал на выходе меняться не должен, а меняется, состязания называются статическими. Т.е. по алгоритму работы устройства, сигнал на выходе меняться не должен при определенном изменении входных сигналов, а из-за состязаний – меняется. Статические состязания называются единичными, если на выходе устройства должна сохраняться 1, а вследствие состязаний – на некоторый интервал времени появляется 0. При нулевых статических состязаниях на выходе появляется 1, тогда как должен сохраняться 0.

Примеры единичных и нулевых статических состязаний сигналов показаны на рис. 4 и 5 соответственно.

На рис. 4 показаны временные диаграммы, иллюстрирующие появление единичных состязаний в устройстве, которое реализует функцию $y = x_1 \cdot x_2 \cdot x_3 \cdot x_1$. В начальный момент времени в устройстве на рис. 4. начальное значение сигналов $x_1 = 1, x_2 = x_3 = 1$. В момент времени t_0 значение сигнала меняется – $x_1 = 0$ – при этом остальные сигналы не меняются. В случае идеальных логических элементов для обеих комбинаций входных сигналов $y = 1$. Но в реальной схеме изменение сигнала в точке **б**, обусловленное изменением x_1 , произойдет на дополнительное Δt_3 позже, чем аналогичное изменение в точке **а**. Как следствие, на выходе на период Δt_3 сигнал примет нулевое значение – произошли единичные статические состязания.

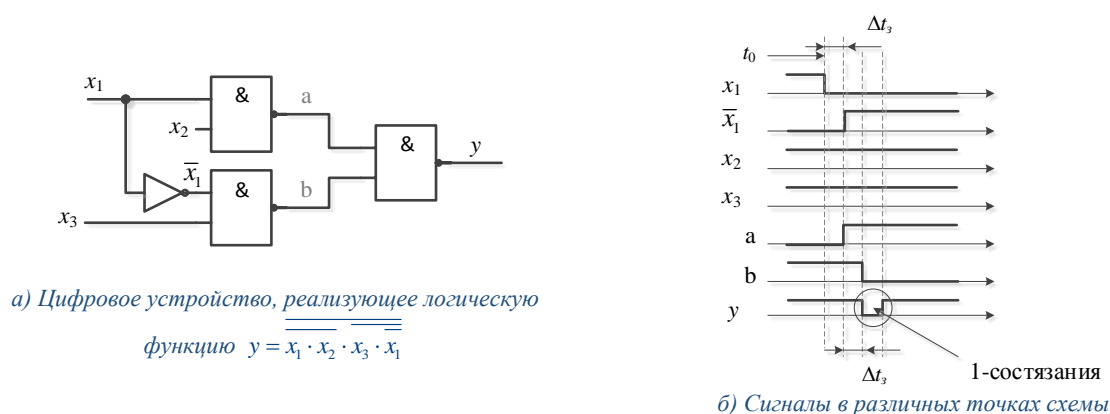


Рис. 4. Пример единичных статических состязаний

На рис. 5 показаны временные диаграммы, иллюстрирующие появление нулевых состязаний в устройстве, которое реализует функцию $y = x_1 \cdot x_2 + x_1 \cdot x_3$. Начальные значения сигналов $x_1 = 0, x_2 = x_3 = 1$. В момент времени t_0 значение сигнала x_1 меняется на противоположное. Вследствие того, что линия к верхнему входу сумматора имеет большую задержку (см. точки **а** и **б**: сигналы в них меняются неодновременно), чем линия к нижнему, возникает ситуация, когда на обоих входах сумматора нули, что приводит к появлению нулевых состязаний. В случае идеального цифрового устройства, сигнал на выходе в рассмотренной ситуации должен быть все время равен 0.

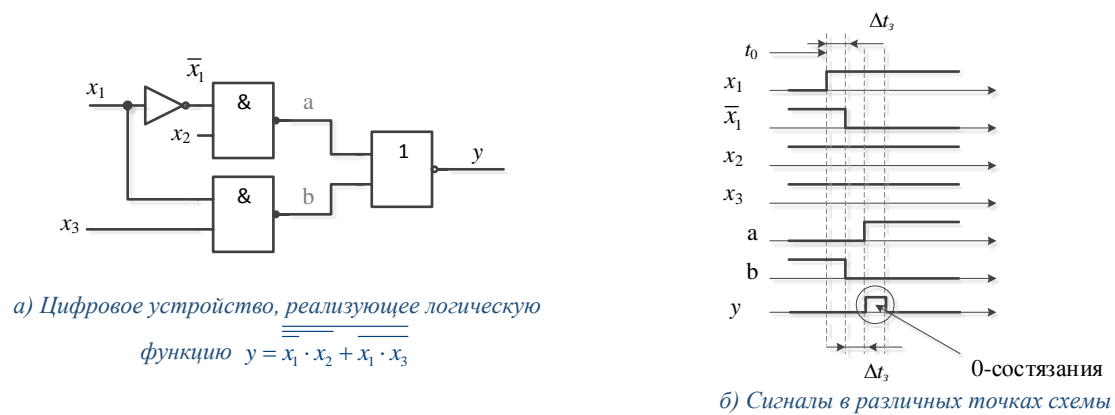


Рис. 5. Пример нулевых статических состязаний

Статические состязания обобщаются на случай, когда они обусловлены изменением не одного, а нескольких входных сигналов. Такие состязания делятся на функциональные и логические. Функциональные состязания являются свойством реализуемой функции и не могут быть устранены путем изменения формы ее представления. Логические же состязания можно устранять путем применения структурных методов проектирования цифровых устройств, свободных от состязаний.

В случае, когда при последовательном переходе между парой состояний сигнал на выходе должен поменяться определенным образом, но происходят дополнительные его изменения состязания называются динамическими. При динамических состязаниях при изменении сигналов на входе на выходе помимо правильных изменений сигнала произойдут дополнительные переходы, не соответствующие алгоритму работы устройства.

Иллюстрация динамических состязаний на примере функции $y = x_1 \cdot x_2 \cdot x_1 \cdot x_3 \cdot x_1 \cdot x_4$ рассмотрен на рис. 6. В точке **с** возникают статические состязания и, если при этом время распространения сигнала от точки **d** к точке **d'** велико (умножители находятся далеко друг от друга в ПЛИС), то на выходе устройства могут возникать динамические состязания, как показано на временных диаграммах на рис. 6,б.

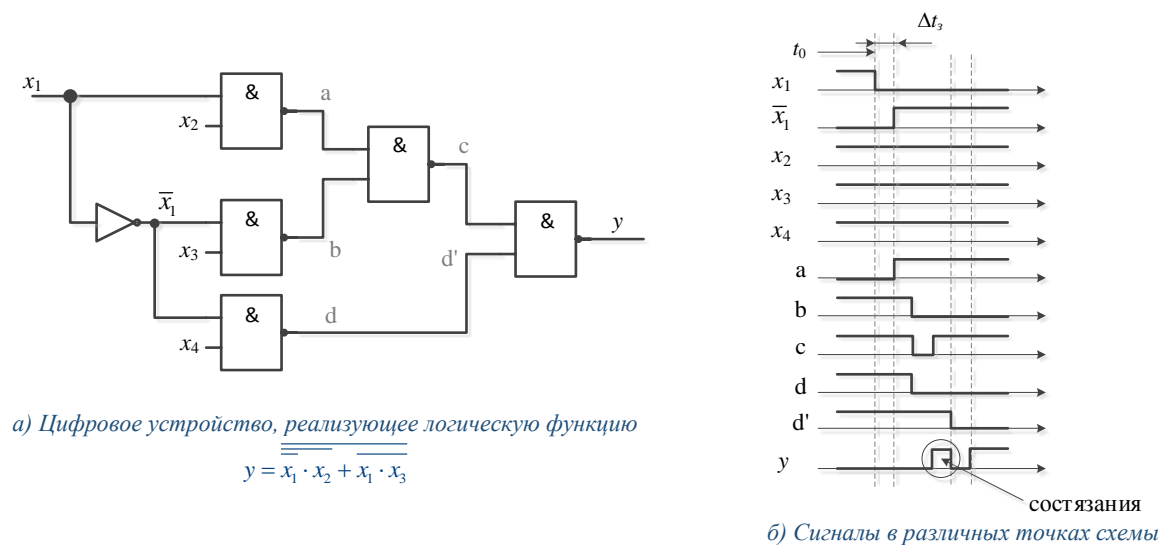


Рис. 6. Пример динамических состязаний

Борьба с состязаниями

Для борьбы с состязаниями могут использоваться разные подходы. К ним относятся структурные методы на основе карт Карно или метода Мак-Класки. Возможны подходы с использованием коррекции опасных состязаний, когда структура цифрового устройства подбирается таким образом, чтобы задержки в логических элементах подавлялись.

На сегодняшний день вследствие высокой сложности разрабатываемых цифровых устройств основным методом борьбы с состязаниями является введение синхронизации. В этом случае работой устройства управляет сигнал тактирования (тактовый сигнал), который представляет из себя меандр. Запись и считывание сигналов на входах/выходах цифрового устройства разрешается только в строго определенные моменты времени, а именно в моменты фронта (и/или среза) сигнала тактирования. При этом переключение комбинационных устройств должно происходить за интервал времени не превышающий период (или полпериода, в случае синхронизации по фронту и срезу тактового сигнала) следования тактовых импульсов, чтобы все переходные процессы, связанные с переключением логических элементов, завершились, а сигналы приняли свое стационарное значение.

Рассмотрим борьбу с состязаниями на примере статических состязаний на рис. 4,б. выходной сигнал считывается только в моменты фронтов сигнала тактирования, при этом его период выбран большим, чем длительность возможных состязаний. Из рис. 7 видно, что ложные изменения сигнала не пропускаются на выход, а игнорируются.



Рис. 7. Борьба с состязаниями путем применения сигнала тактирования

Более подробно сигнал тактирования и синхронные цифровые устройства будут рассмотрены в следующих лекциях.

Особенности FPGA

Процесс отладки проектов под FPGA обычно дольше, чем под DSP, при прочих равных условиях. Это обусловлено тем, что необходимо учитывать большее количество сигналов, в том числе низкоуровневых (сигналы тактирования, сброса).

Для ускорения разработки часто используемые блоки (от простейших комплексных умножителей до сложных микропроцессоров или блоков обработки сигналов) организуют в так называемые IP-ядра, IP – Intellectual Property. Ядра представляют из себя отдельно разработанные и оптимизированные под конкретную архитектуру FPGA решения (например, ядро блока быстрого преобразования Фурье, ядро декодера турбокода системы LTE). IP-ядра подвергаются тщательному тестированию на ошибки, производительность, количество используемых ресурсов. Стоимость некоторых сложных ядер может достигать десятка тысяч долларов.

В составе сред разработки (например, Vivado) присутствует репозиторий IP-ядер, который позволяет генерировать различные часто используемые ядра (часть ядер не требует дополнительной оплаты, часть – требует). Сгенерированное ядро можно далее использовать в

своей программе. Подключается ядро также, как и любой другой модуль VHDL. Более того, пользователи могут оформлять свои модули в виде IP-ядер и подключать их в общий репозиторий.