**Основы статического временного анализа.**

**Часть 2.1: System Synchronous** **Input Delay Constraint.**

**Введение.**

Данная статья является второй в серии статей по временным ограничениям в FPGA. Основная цель – познакомить начинающих разработчиков с основами статического временного анализа. Будет рассмотрен временной анализ при передачи данных в FPGA из внешнего устройства и показано два способа создания ограничений для входных сигналов.

1. **Цель временных ограничений на входные сигналы.**

В цифровых синхронных устройствах данные передаются между двумя триггерами, которые разделены комбинационной логикой. В предыдущей статье был представлен временной анализ для случая, когда оба триггера располагаются внутри FPGA [1]. Предполагается, что читатель уже знаком с такими понятиями, как ограничения на максимальное и минимальное время распространения, время установки и удержания, *Slack* и т.д.

Зачастую обрабатываемые данные передаются между FPGA и другими микросхемами, расположенными на той же печатной плате. Эти пути также должны удовлетворять ограничениям по *Setup* и *Hold*, чтобы плата как единое целое могла работать корректно. Поэтому практически всегда для входных сигналов FPGA требуется вводить временные ограничения.

В качестве примера будем рассматривать устройство, схема которого показана рисунке 1. Помимо FPGA на плате присутствует микросхема 74HC595 [2], которая представляет из себя последовательный регистр сдвига и часто поставляется в составе обучающих наборов для Arduino. Также на плате располагается тактовый генератор (OSC), формирующий тактовый сигнала для FPGA и 74HC595.

Данная конфигурация, при которой тактовый сигнал для источника и приемника данных поступает от внешнего генератора, называется System Synchronous. Другой вариант, когда источник помимо данный также формирует и тактовый сигналы, называется Source Synchronous.

На рисунке 1 отмечены только те ножки микросхемы 74HC595, которые будут рассматриваться в последующем. Ножка *SER* соответствует входу регистра сдвига, ножка QH’ – его выходу. Регистр сдвига тактируется сигналом, который поступает на ножку SRCLK. В дальнейшем микросхему 74HC595 для краткости будем называть Device*.* Далее будет представлен временной анализ для передачи данных с выхода QH’ Device на вход IDATA FPGA. На рисунке 1 данный путь отмечен красным цветом.

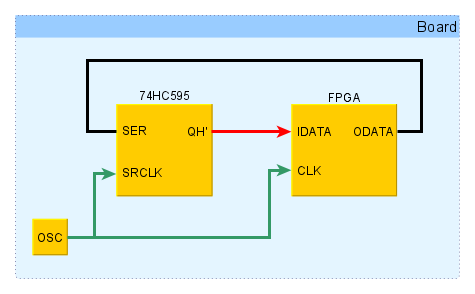


Рисунок 1. Схема соединения устройств на плате.

Также пусть FPGA загружен простой проект, состоящий из двух триггеров и двух LUT, которые реализуют логическое отрицание. Схема проекта показана на рисунке 2.

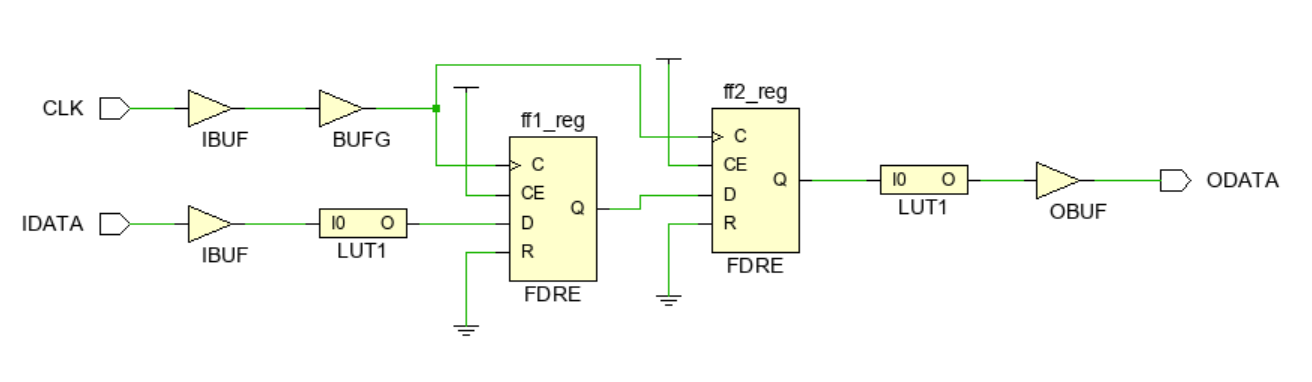


Рисунок 2. Схема FPGA проекта.

Описание проекта на System Verilog представлено ниже:

module top (

input  logic CLK,

    input  logic IDATA,

    output logic ODATA

);

    logic ff1, ff2;

    always\_ff@(posedge CLK)

       ff1 <= ~IDATA;

    always\_ff@(posedge CLK)

       ff2 <= ff1;

   assign ODATA = ~ff2;

endmodule

Данный проект не имеет какой-либо практической ценности, однако на его примере можно продемонстрировать, как проводится временной анализ для входных сигналов.

1. **Задержки при временном анализе для входных сигналов.**

Анализ передачи данных между Device и FPGA мало отличается от случая, рассмотренного в [1] для двух триггеров внутри FPGA. Особенностью является то, что запускающий триггер располагается в одном устройстве, а защелкивающий в другом. На рисунке 3 показан анализируемый путь, на который нанесены задержки для данных и тактового сигнала.

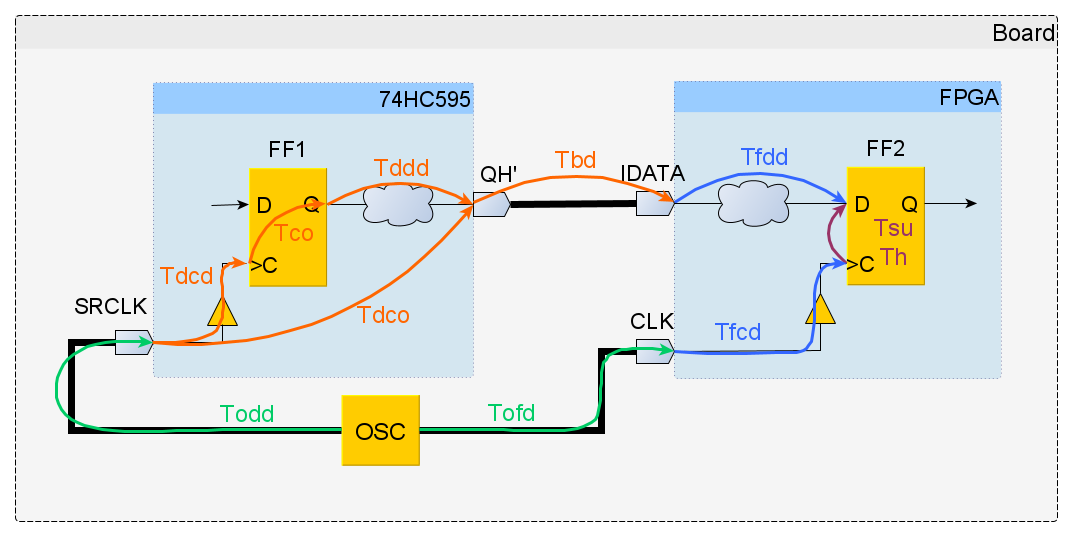


Рисунок 3. Путь с задержками для входных данных и тактового сигнала.

Ниже даны определения задержек, представленных на рисунке 3.

* (***O****sc to* ***D****evice* ***D****elay*) – задержка тактового сигнала от генератора до ножки SCLR микросхемы 74HC595;
* (***O****sc to* ***F****PGA* ***D****elay*) – задержка тактового сигнала от генератора до ножки CLK FPGA;
* (***D****evice* ***C****lock* ***D****elay*) – задержка тактового сигнала от ножки SCLR микросхемы 74HC595 до тактового входа запускающего триггера;
* (***C****lock to* ***O****utput*) – интервал времени между приходом фронта на тактовый вход триггера и появлением данных на его выходе Q;
* (***D****evice* ***D****ata* ***D****elay*) – задержка распространения данных от запускающего триггера до ножки QH’ микросхемы 74HC595;
* (***B****oard* ***D****elay*) – задержка распространения данных между микросхемой 74HC595 и FPGA по дорожкам платы;
* (***F****PGA* ***C****lock* ***D****elay*) – задержка тактового сигнала от ножки CLK FPGA до тактового входа защелкивающего триггера;
* (***F****PGA* ***D****ata* ***D****elay*) – задержка распространения данных от ножки IDATA FPGA до защелкивающим триггером;
* (***S****et****U****p time*) – время установки триггера.
* (***H****old time*) – время удержания триггера.

Период тактового сигнала будем обозначать как *Tclk*. Красным и зеленым цветом на рисунке 3 представлены задержки для участков пути, которые располагаются вне FPGA. Эти задержки необходимо самостоятельно указать временному анализатору Vivado.

1. **Максимальное время распространения.**

Для начала рассмотрим, каким образом выполняется анализ для проверки ограничения на максимальное время распространения (*Setup*). Напомним, что временной анализ по *Setup* всегда проводится для самого пессимистичного случая, которому соответствует максимально задержанный запускающий фронт, максимально медленное распространение данных и максимально быстро распространяющийся защелкивающий фронт.

Сначала найдем фактическое время прибытия данных ко входу расположенного внутри FPGA защелкивающего триггера. Уравнения для расчета представлены ниже (см. рисунок 3):

* Время прибытия фронта к запускающему триггеру в внутри Device (***S****ource* ***С****lock* ***A****rrival time*):
* Задержка распространения данных (***D****ata* ***D****elay*):
* Время прибытия данных на вход защелкивающего триггера внутри FPGA (***D****ata* ***A****rrival time*):

Подставив (1) и (2) в уравнение для , получим:

Введем обозначение:

Задержкасоответствует интервалу времени между приходом фронта на тактовый вход микросхемы 75HC595 и появлением данных на ее выходе QH’.Тогда уравнение для ***D****ata* ***A****rrival time* можно записать в виде:

Теперь вычислим время требуемое время прибытия данных. Защелкивающий фронт появляется через один такт после запускающего фронта, поэтому к задержке распространения добавлен один период тактового сигнала.

* Время прибытия фронта к защелкивающему триггеру внутри FPGA (***D****estination* ***C****lock* ***A****rrival time*):
* Требуемое время прибытия данных (***D****ata* ***R****equired time*):

В предыдущем уравнении учитывается, что данные на входе защелкивающего триггера должны быть стабильны в течении времени *Tsu* до прихода фронта тактового сигнала.

Для анализа по *Setup* величина запаса для задержки данных (*Slack*) вычисляется по формуле:

Если *Slack* принимаетотрицательное значение, то это указывает, что данные приходят на вход защелкивающего триггера позже, чем требуется. То есть,ограничение по *Setup* нарушено. Используя ранее полученные уравнения, можно получить полное выражение для расчета *Slack*

1. **Минимальное время распространения.**

Теперь рассмотрим, как выполняется анализ для проверки ограничения на минимальное время распространения (*Hold*). При анализе по *Hold* считается, что задержки для запускающего фронта и данных имеют минимальное значение, а для защелкивающего фронта – максимальное.

Расчет фактического времени прибытия данных представлен ниже:

* Время прибытия фронта к триггеру внутри Device (***S****ource* ***С****lock* ***A****rrival time*):
* Задержка распространения данных (***D****ata* ***D****elay*):
* Время прибытия данных к триггеру внутри FPGA (***D****ata* ***A****rrival time*):

где, как и ранее используется обозначение

Далее представлены уравнения для вычисления требуемого времени прибытия данных:

* Время прибытия фронта к защелкивающему триггеру внутри FPGA (***D****estination* ***C****lock* ***A****rrival time*):
* Требуемое время прибытия данных (***D****ata* ***R****equired time*):

Напомним, что защелкивающий фронт для предыдущих данных появляется в тот же момент времени, что и запускающий фронт для следующих данных. По этой причине в задержке распространения отсутствует слагаемое, равное периоду тактового сигнала. Также отметим, что слагаемое в уравнении для учитывает, что данные не должны изменяться в течении времени удержания после защелкивающего фронта.

Уравнение для расчета *Slack* при анализе по *Hold* имеет вид:

Используя полученные выше результаты, выражение для *Slack* можно записать в виде:

1. **Первый способ создания временных ограничения в Vivado.**

Перейдем от теории к практике и рассмотрим первый способ создания временных ограничений для входного сигнала. Для начала разберемся с ограничениями для анализа по *Setup*.

В уравнении (1) слагаемые, выделенные зеленым и оранжевым цветом, неизвестными анализатору Vivado, так как они описывают задержки вне FPGA. Также неизвестным является период тактового сигнала . Создание ограничений на период тактового сигнала с помощью команды create\_clock было рассмотрено в [1].

Для определенности будем считать, что источник питания для микросхемы 74HC595 имеет напряжение 4.5 В. Из таблицы 7.6 datasheet для 74HC595 [2] находим, что максимальная рабочая частота равна 31 МГц. Также пускай требуется, чтобы плата на рисунке 1 могла работать, когда частота тактового генератора (OSC) равна 10 МГц. Тогда ограничение на период тактового сигнала можно записать в виде:

# ограничение на период тактового сигнала

create\_clock -period 100 -name clk\_10MHz [get\_ports CLK]

Данная команда создает тактовый сигнал clk\_10MHz с периодом 100 нс, который поступает в FPGA на ножку CLK.

Объединим все оставшиеся в уравнении (1) неизвестные слагаемые в одну переменную . Тогда выражение для *Slack* можно переписать в виде:

где

В слагаемые , и задают задержки обусловленные распространением сигнала по дорожкам печатной платы. Их значения зависят от многих факторов: материал подложки; длины дорожек; типы дорожек, например, полосковые или микрополосковые и т.д.

Для каждого типа дорожки существуют приближенные выражения для оценки скорости распространения по ней сигнала. Зная скорость и длину дорожки, можно оценить задержку. Пример приближенных расчетов для разных типов дорожек можно найти в [3].

Будем считать, что мы смогли оценить минимальные и максимальные задержки распространения сигнала по дорожкам печатной платы. В качестве примера примем следующие значения в наносекундах: , , , , и . Значения задержек можно указать в файле с временными ограничениями (xdc-файл) в следующем виде:

# минимальное и максимальное время распространения данных по

# дорожкам платы

set Tbd\_max 0.6

set Tbd\_min 0.5

# минимальное и максимальное время распространения тактового сигнала

# от генератора до микросхемы 74HC595

set Todd\_max 0.4

set Todd\_min 0.2

# минимальное и максимальное время распространения тактового сигнала

# от генератора FPGA

set Tofd\_max 0.3

set Tofd\_min 0.2

Рассмотрим задержку и уравнения (3), которая соответствует интервалу времени между приходом фронта на тактовый вход SCLK микросхемы 75HC595 и появлением данных на ее выходе QH’. Ее значение можно получить из таблицы 7.6 datasheet для 74HC595 [2]. Данная задержка обозначена как и при напряжении питания 4.5 В может изменяться от 17 до 32 нс. Эти значения также вносятся в xdc-файл:

# задержка распространения между приходом тактового сигнала и

# появлением данных на выходе QH' микросхемы 74HC595

set Tdco\_min 17

set Tdco\_max 32

С учетом выше сказанного для создания временных ограничений на входной сигнал IDATA для анализа по Setup в xdc-файл необходимо внести следующие команды:

# временные ограничение для входного сигнала IDATA

set idelay\_max [expr $Todd\_max - $Tofd\_min + $Tdco\_max + $Tbd\_max]

set\_input\_delay -clock clk\_10MHz -max $idelay\_max [get\_ports IDATA];

В первой строке объявляется переменная idelay\_max, значение которой инициализируется в соответствии с уравнением (10). Во второй строке с помощью команды set\_input\_delay содеются ограничения для входного сигнала. Опция -max $idelay\_max указывает задержку из уравнения (10) для анализа по Setup. Конструкция [get\_ports IDATA] указывает, что ограничения задаются для входного сигнала, который поступает в FPGA на ножку IDATA.

Стоит заметить, что так как запускающий триггера располагается не в FPGA, а внутри микросхеме 74HC595 Vivado необходимо указать, каким тактовым сигналом тактируется данный триггер. Это делается с помощью опции -clock clk\_10MHz, которая указывает на ранее созданный с помощью команды create\_clock тактовый сигнал. Так как защелкивающий триггер располагается внутри FPGA, Vivado известен его тактовый сигнал.

Аналогичным образом задаются ограничения для анализа по Hold. Объединим все неизвестные слагаемые в уравнении (10) в одну переменную, которая будет иметь:

Тогда уравнение для расчета можно переписать в виде:

Для указания ограничений при анализе по Hold в xdc-файл необходимо внести следующие команды

# временные ограничение для входного сигнала IDATA

set idelay\_min [expr $Todd\_min - $Tofd\_max + $Tdco\_min + $Tbd\_min]

set\_input\_delay -clock clk\_10MHz -min $idelay\_min [get\_ports IDATA];

В первой строке создается переменная, значение которой равно из уравнения (1). Во второй строке временные ограничение создается командой set\_input\_delay. Опция -min $idelay\_min указывает, что задержки указаны для проведения анализа по Hold.

Полное содержимое xdc-файла представлено ниже:

# задержка распространения между приходом тактового сигнала и появлением

# данных на выходе QH' микросхемы 74HC595

set Tdco\_min 17

set Tdco\_max 32

# минимальное и максимальное время распространения данных по дорожкам платы

set Tbd\_max 0.6

set Tbd\_min 0.5

# минимальное и максимальное время распространения тактового сигнала от

# генератора до микросхемы 74HC595

set Todd\_max 0.4

set Todd\_min 0.2

# минимальное и максимальное время распространения тактового сигнала от

# генератора FPGA

set Tofd\_max 0.3

set Tofd\_min 0.2

# ограничение на период тактового сигнала

create\_clock -period 100 -name clk\_10MHz [get\_ports CLK]

# временные ограничение для входного сигнала IDATA

set idelay\_max [expr $Todd\_max - $Tofd\_min + $Tdco\_max + $Tbd\_max]

set\_input\_delay -clock clk\_10MHz -max $idelay\_max [get\_ports IDATA];

set idelay\_min [expr $Todd\_min - $Tofd\_max + $Tdco\_min + $Tbd\_min]

set\_input\_delay -clock clk\_10MHz -min $idelay\_min [get\_ports IDATA];

Теперь рассмотрим отчеты о проведении временного анализа, полученные после размещения и трассировки проекта. На рисунке 4 представлен раздел Summary при проведении анализа по Setup. В данном разделе источник сигнала (ножка IDATA), защелкивающий триггер (ff1), задержка данных внутри FPGA (Data Path Delay) и количество уровней логики (Logic Levels). Также можно увидеть рассчитанный Slack, расфазировку (Skew) и неопределенность (Uncertainty) для тактового сигнала.

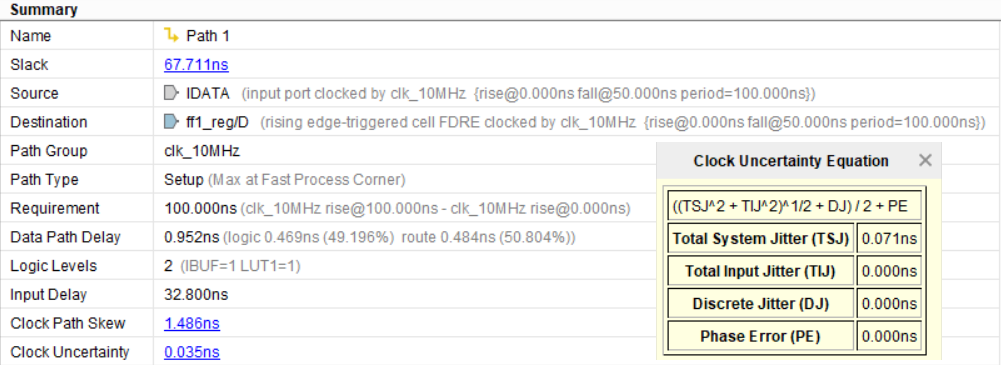
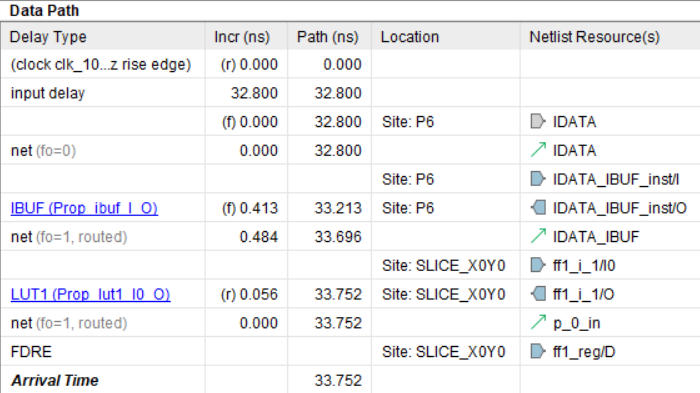
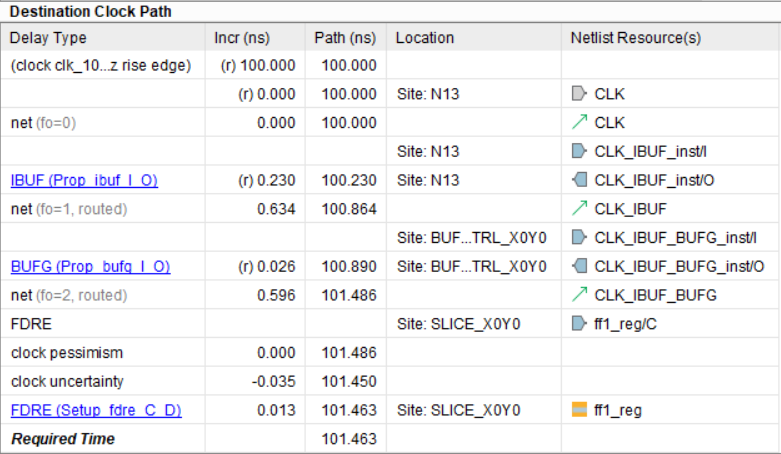
**

Рисунок 4. Общие сведения о входном пути

На рисунках 5 и 6 показаны задержки для данных и защелкивающего фронта соответственно.

**

Рисунках 5. Расчет фактического времени прибытия данных.

**

Рисунках 6. Расчет требуемого времени прибытия данных.

Результаты, представленные на данных рисунках, следует интерпретировать следующим образом. Из рисунка 6 следует, что защелкивающий фронт поступает в FPGA на ножку CLK в момент времени 100 нс. Далее этот фронт проходит через входной и тактовый буферы и попадает на защелкивающий триггер в момент времени 101.486 нс. С учетом времени удержания и неопределенности тактового сигнала данные должны достигнуть защелкивающего триггера в момент времени 101.463 нс.

Если защелкивающий фронт поступает на ножку FPGA в момент времени 100 нс., значит, учитывая задержку распространения он формируется на выходе генератора в момент времени нс. Запускающий фронт формируется на один период раньше, то есть когда время равно -0.2 нс. Через этот фронт поступает на микросхему 74HC595, то есть в момент времени -0.2 + 0.4 = 0.2 нс.

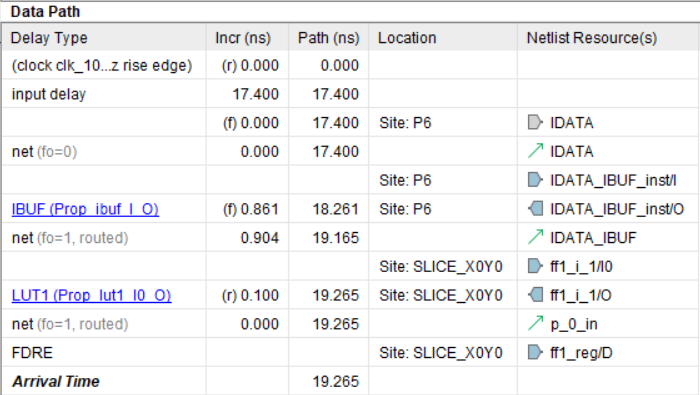
Поступление этого фронта запустит данные триггера внутри 74HC595. Спустя задержку нс., то есть в момент времени 32.8 нс. эти данные попадут на ножку FPGA. На рисунке 5 этой задержке соответствует параметр Input Delay во второй строке. Внутри FPGA данные проходят через входной буфер и LUT и попадают на вход защелкивающего триггера в момент времени 33.752 нс.

Отсюда Slack равен нс.

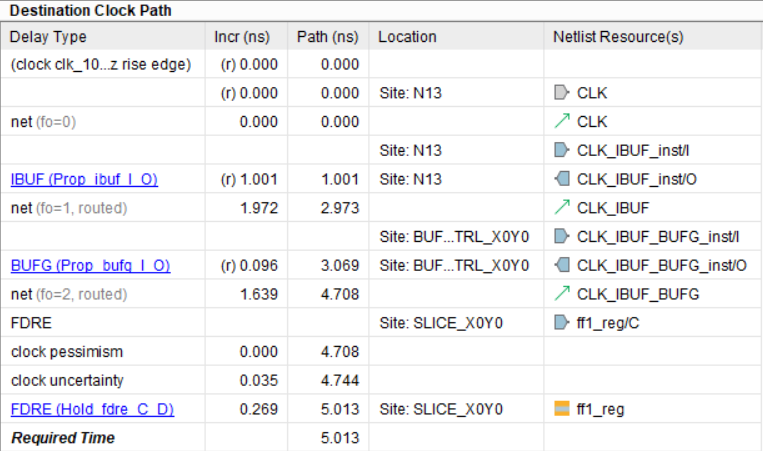
Аналогично можно рассмотреть результаты анализа по Hold, представленные на рисунках 7 и 8.

Из рисунка 8 можно увидеть, что защелкивающий фронт попадает на ножку FPGA в нулевой момент времени. С учетом задержки распространения это означает, что он появился на выходе генератора на нс. раньше. Для анализа по Hold запускающий и защелкивающий фронты появляются в одно и тоже время. Тогда запускающий фронт появится на входе 74HC595 в момент времени нс. Спустя нс. данные появятся на выходе QH’ микросхемы 74HC595. Далее они будут распространятся по дорожке платы в течении нс. и попадут на ножку FPGA в момент времени нс. Данная задержка представлена на рисунке 6 под названием Input Delay. После чего данные проходят через входной буфер и LUT попадают на вход защёлкивающего триггера в момент времени 19.265 нс.

Защелкивающий фронт проходит через входной и тактовый буферы и попадает на тактовый вход защелкивающего триггера в момент времени 4.708 нс. С учетом неопределенности тактового сигнала и времени удержания данные должны прийти к защелкивающему триггеру в момент времени 5.013 нс.

**

Рисунках 7. Расчет фактического времени прибытия данных.

**

Рисунках 8. Расчет фактического времени прибытия данных.

Такой подход прост тем, что можно просто представить все неизвестные задержки в виде одной величины . Однако, как можно было увидеть, это приводит к тому, что временные отсчеты достаточно сложно интерпретировать.

1. **Второй способ создания временных ограничения в Vivado.**

Теперь рассмотрим второй способ задания временных ограничений для входного сигнала. В Vivado есть возможность отдельно указывать задержки распространения тактового сигнала по дорожкам печатной платы. Это делается с помощью команды set\_clock\_latency . Ниже представлены команды xdc-файла для указания задержек тактового сигнала:

# ограничение на период тактового сигнала, поступающего в FPGA

create\_clock -period 100 -name clk\_10MHz [get\_ports CLK]

# задержки распространения тактового сигнала от генератора до FPGA

set\_clock\_latency -source -early $Tofd\_min [get\_clocks clk\_10MHz]

set\_clock\_latency -source -late $Tofd\_max [get\_clocks clk\_10MHz]

Как и раньше с помощью create\_clock  задается ограничение на период тактового сигнала, который поступает на ножку FPGA. Далее первая команда set\_clock\_latency с опцией -early  указывает, что минимальное время распространения равно . Конструкция [get\_clocks clk\_10MHz] указывает тактовый сигнал, для которого задаются задержки. Вторая команда set\_clock\_latency с опцией -late  задает максимальное время распространения равным . Опция -source  в обоих командах означает, что указана задержка от источника тактового сигнала до ножки FPGA.

Так как задержка распространения тактового сигнала до микросхемы 74HC595 имеет значение отличное от задержки до FPGA, в xdc-файле требуется создать еще один тактовый сигнал. Команды xdc-файла представлены ниже:

# ограничение на период виртуального тактового сигнала, поступающего

# в микросхему 74HC595

create\_clock -period 100 -name clk\_10MHz\_Dev

# задержки распространения тактового сигнала от генератора до микросхемы 74HC595

set\_clock\_latency -source -early $Todd\_min [get\_clocks clk\_10MHz\_Dev]

set\_clock\_latency -source -late $Todd\_max [get\_clocks clk\_10MHz\_Dev]

С помощью команды create\_clock  создается тактовый сигнал с именем clk\_10MHz\_Dev и периодом 100 нс. Обратите внимание, что, так как этот сигнал поступает только в микросхему 74HC595, у данного сигнала не указана ножка FPGA. Тактовый сигнал, который присутствует на плате, но не поступает в FPGA называется виртуальным (virtual clock). Для данного виртуального сигнала минимальная и максимальная задержки задаются равными и соответственно.

Также теперь можно упростить выражения для и , которые будут содержать только задержки распространения данных и примут вид:

Временные ограничения для сигнала IDATA также задаются с помощью команды set\_input\_delay в следующем виде:

# временные ограничение для входного сигнала IDATA

set idelay\_max [expr $Tdco\_max + $Tbd\_max]

set\_input\_delay -clock clk\_10MHz\_Dev -max $idelay\_max [get\_ports IDATA];

set idelay\_min [expr $Tdco\_min + $Tbd\_min]

set\_input\_delay -clock clk\_10MHz\_Dev -min $idelay\_min [get\_ports IDATA];

Как и раньше опция -max  указывает, что задержки предназначены для анализа по Setup, а опция -min - для анализа по Hold. Обратите внимание, что теперь запускающий триггер внутри микросхемы 74HC595 тактируется виртуальным тактовым сигналом, что указано с помощью опции -clock clk\_10MHz\_Dev .

Полное содержимое xdc-файла представлено ниже:

# задержка распространения между приходом тактового сигнала и появлением

# данных на выходе QH' микросхемы 74HC595

set Tdco\_min 17

set Tdco\_max 32

# минимальное и максимальное время распространения данных по дорожкам платы

set Tbd\_max 0.6

set Tbd\_min 0.5

# минимальное и максимальное время распространения тактового сигнала от

# генератора до микросхемы 74HC595

set Todd\_max 0.4

set Todd\_min 0.2

# минимальное и максимальное время распространения тактового сигнала от

# генератора до FPGA

set Tofd\_max 0.3

set Tofd\_min 0.2

# ограничение на период тактового сигнала, поступающего в FPGA

create\_clock -period 100 -name clk\_10MHz [get\_ports CLK]

# задержки распространения тактового сигнала от генератора до FPGA

set\_clock\_latency -source -early $Tofd\_min [get\_clocks clk\_10MHz]

set\_clock\_latency -source -late $Tofd\_max [get\_clocks clk\_10MHz]

# ограничение на период виртуального тактового сигнала, поступающего

# в микросхему 74HC595

create\_clock -period 100 -name clk\_10MHz\_Dev

# задержки распространения тактового сигнала от генератора до микросхемы 74HC595

set\_clock\_latency -source -early $Todd\_min [get\_clocks clk\_10MHz\_Dev]

set\_clock\_latency -source -late $Todd\_max [get\_clocks clk\_10MHz\_Dev]

# временные ограничение для входного сигнала IDATA

set idelay\_max [expr $Tdco\_max + $Tbd\_max]

set\_input\_delay -clock clk\_10MHz\_Dev -max $idelay\_max [get\_ports IDATA];

set idelay\_min [expr $Tdco\_min + $Tbd\_min]

set\_input\_delay -clock clk\_10MHz\_Dev -min $idelay\_min [get\_ports IDATA];

На рисунке 9 представлены общие сведения о входном пути для анализа по Setup. Обратите внимание, что теперь в строчке Source указано, что входной сигнал на ножке IDATA тактируется виртуальным сигналом clk\_100MHz\_Dev.

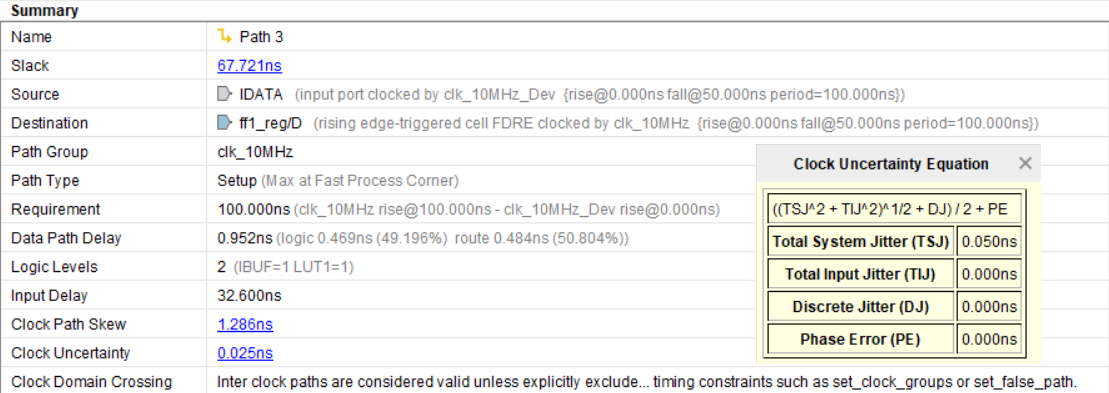
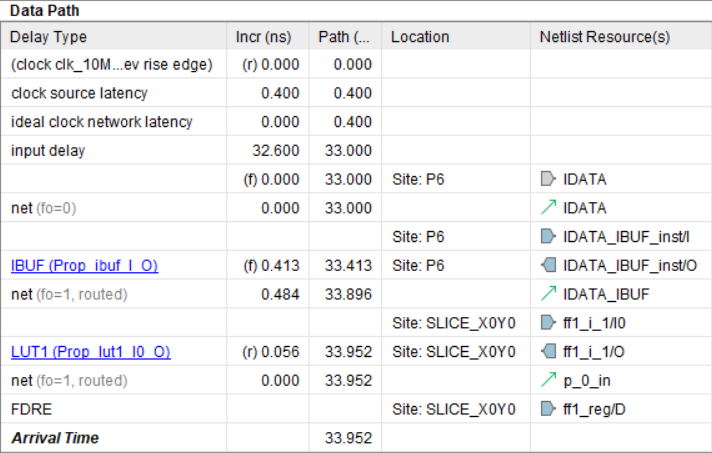
**

Рисунок 9. Общие сведения о входном пути

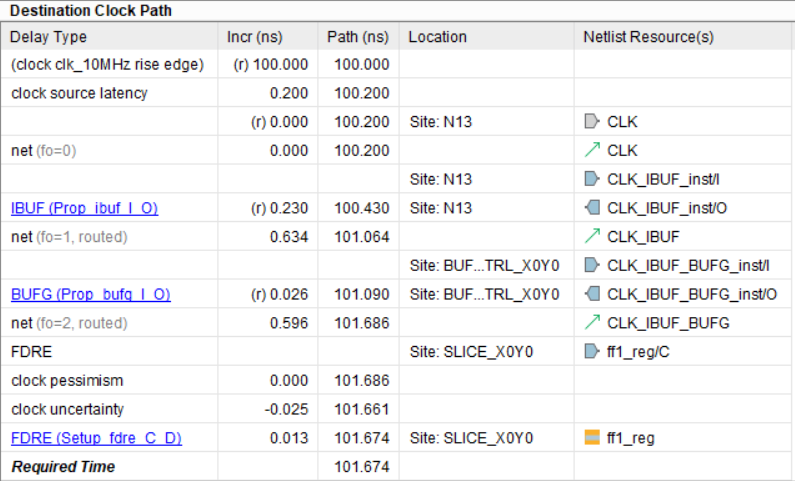
На рисунке 10 и 11 показаны временные отчеты для фактического и требуемого и времени прибытия данных при анализе по Setup. На рисунке 10 можно увидеть, что запускающий фронт появляется на выходе генератора в нулевой момент времени. Далее указывается, что тактовый сигнал попадает на вход 74HC595 c задержку в 0.4 нс. (clock source latency), что соответствует .

**

Рисунках 10. Расчет фактического времени прибытия данных.

После этого еще спустя данные нс. появятся на ножке FPGA. Это произойдет в момент времени равный нс. С учетом задержек распространения внутри FPGA требуемое время прибытия данных составит 33.952 нс.

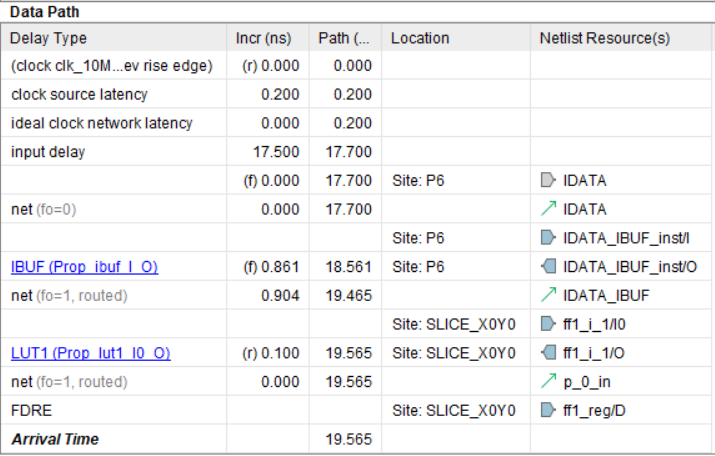
На рисунке 12 можно увидеть, что защелкивающий фронт появится на выходе генератора спустя один период в 100 нс. Далее через нс. этот фронт появится на ножке FPGA, пройдет через входной и тактовый буферы и попадет на защелкивающий триггер. Требуемое время прибытия данных составит 101.674 нс., а Slack для анализа по Setup будет равен 101.674-33.952=67.721 нс.

**

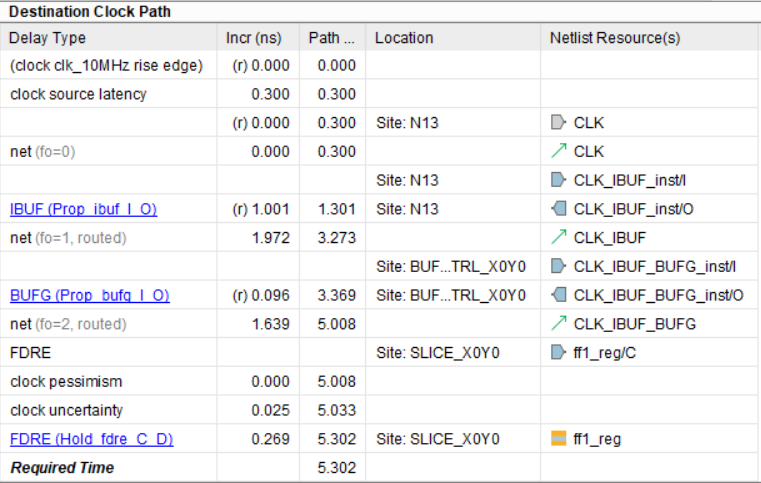
Рисунках 12. Расчет требуемого времени прибытия данных.

Аналогично для анализа по Hold на рисунке 13 можно увидеть, что запускающий фронт появляется на выходе генератора в нулевой момент времени. Спустя нс., он появится на входе SCLK микросхемы 75HC595, а еще спустя нс. данные появятся на ножке FPGA.

Из рисунка 14 можно увидеть, что защелкивающий фронт появится на выходе генератора также в нулевой момент времени и дойдет до ножки FPGA через нс.

**

Рисунках 13. Расчет фактического времени прибытия данных.

**

Рисунках 14. Расчет требуемого времени прибытия данных.

Использование виртуальных тактовых сигналов для создания ограничений на входные сигналы приводит к тому, что имеют более ясную интерпретацию и лучше соответствуют физическому распространению сигналов в плате.

Более того, это рекомендуется Vivado. Если открыть Language Templates в вкладках XDC и System Synchronous Input можно увидеть пример временных ограничений представленный ниже. В данном случае значение задается как сумма задержек tco () и trce\_dly (). Также отметим, что если пренебречь задержками и , то оба способа будет полностью эквивалентны.

# Rising Edge System Synchronous Inputs

#

# A Single Data Rate (SDR) System Synchronous interface is

# an interface where the external device and the FPGA use

# the same clock, and a new data is captured one clock cycle

# after being launched

#

# input      \_\_\_\_\_\_\_\_\_\_            \_\_\_\_\_\_\_\_\_\_

# clock   \_\_|          |\_\_\_\_\_\_\_\_\_\_|          |\_\_

#           |

#           |------> (tco\_min+trce\_dly\_min)

#           |------------> (tco\_max+trce\_dly\_max)

#         \_\_\_\_\_\_\_\_\_\_      \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# data    \_\_\_\_\_\_\_\_\_\_XXXXXX\_\_\_\_\_ Data \_\_\_\_\_XXXXXXX

#

set input\_clock     <clock\_name>;  # Name of input clock

set tco\_max         0.000;         # Maximum clock to out delay (external device)

set tco\_min         0.000;         # Minimum clock to out delay (external device)

set trce\_dly\_max    0.000;         # Maximum board trace delay

set trce\_dly\_min    0.000;         # Minimum board trace delay

set input\_ports     <input\_ports>; # List of input ports

# Input Delay Constraint

set\_input\_delay -clock $input\_clock -max [expr $tco\_max + $trce\_dly\_max]

[get\_ports $input\_ports];

set\_input\_delay -clock $input\_clock -min [expr $tco\_min + $trce\_dly\_min]

[get\_ports $input\_ports];

В заключении отметим следующий интересны момент. Можно увидеть, что значения Slack, представленные на рисунках 4 и 9, равны 67.711 нс. и 67.721 нс. соответственно. Рассмотрев временные отчеты можно увидеть, что данная разница возникает из-за различных значений неопределенности тактового сигнала (clock uncertainty).

Как отмечено в [1] неопределенности тактового сигнала может вохникать из-за джиттера. Значение общего джиттера *Ttsj* (***T****otal* ***S****ystem* ***J****itter*) рассчитывается по формуле:

где (***S****ystem* ***J****itter* ***S****ource*) – джиттер для запускающего триггера; (***S****ystem* ***J****itter* ***D****istention*) – джиттер для защелкивающего триггера. По умолчанию для тактовых сигналов, поступающих на ножку FPGA, значение равно 0.05 нс.

Тогда, при использовании первого способа задания временных ограничения, получаем, что нс., а неопределенность тактового сигнала равна нс. Эти значения можно увидеть на рисунке 4.

Если используется второй способ задания ограничения, то один из тактовых сигналов является виртуальным, и по умолчанию его джиттер считается равным нулю. То есть в этом случае поучаем, что нс. и нс. Неопределенность тактового сигнала нс., что можно увидеть на рисунке 9.

Чтобы значения Slack для обоих способов задания ограничений были равны можно использовать команды set\_system\_jitter и set\_input\_jitter.

**Заключение.**

В данной статье был рассмотрен временной анализ для входных сигналов FPGA. Показан вывод уравнений статического временного анализа. Представлено два способа задания временных ограничений. Рассказано о таком понятии как виртуальный тактовый сигнал. В следующей части предполагается использовать тот же подход для рассмотрения System Synchronous Output Delay.

**Ссылки.**