**Основы статического временного анализа.**

**Часть 2: System Synchronous** **Input Delay Constraint.**

**Введение.**

Это вторая статья из серии статей по временным ограничениям в FPGA. Основная цель – познакомить начинающих разработчиков с основами статического временного анализа. Ранее был рассмотрен временной анализ при передачи данных между двумя последовательными элементами внутри FPGA [1]. В данной статье будет представлен случай, когда данные поступают в FPGA от внешнего устройства.

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

Внутри синхронных устройствах данные передаются между двумя триггерами, которые разделены комбинационной логикой. Для корректной передачи данных должны быть выполнены ограничения на максимальное (*Setup*) и минимальное (*Hold*) время распространения.

На плате помимо FPGA присутствуют другие микросхемы, которые передают данные в FPGA. Чтобы вся плата работала на заданной частоте, передача данных между FPGA и внешним устройством (Device), также должна удовлетворять ограничениям по *Setup* и *Hold*.

Будем считать, что на плате помимо FPGA присутствует микросхема 74HC595 [2]. Данная микросхема представляет из себя последовательный регистр сдвига и часто поставляется в обучающих наборах для Arduino. Также на плате присутствует тактовый генератор (OSC), формирующий тактовый сигнала для FPGA и 74HC595. Схема соединения устройств на плате представлен на рисунке 1.

Данная конфигурация, когда тактовый сигнал для источника и приемника данных формируется внешним генератором, называется System Synchronous. Другой вариант, при котором источник помимо данный формирует тактовый сигналы, называется Source Synchronous.

На рисунке 1 отмечены только те ножки микросхемы 74HC595, которые будет нас интересовать в дальнейшем. Ножка *SER* соответствует входу регистра сдвига, ножка QH’ – его выходу. Регистр сдвига тактируется сигналом, поступающим на ножку SRCLK. В данной статье будет рассмотрен временной анализ для передачи данных с выхода QH’ микросхемы 74HC595 на вход IDATA FPGA. На рисунке 1 данный путь отмечен красным цветом.

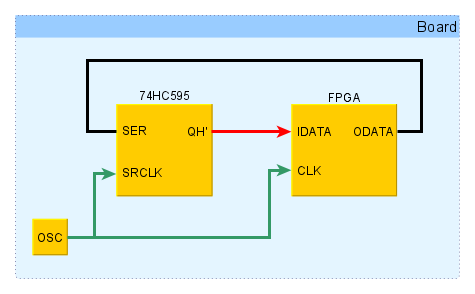


Рисунок 1. Схема соединения устройств на плате.

В качестве примера рассмотрим простейший FPGA проект, состоящий из двух триггеров и двух LUT, которые реализуют логическое отрицание. Схема проекта показана на рисунке 2. Описание 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

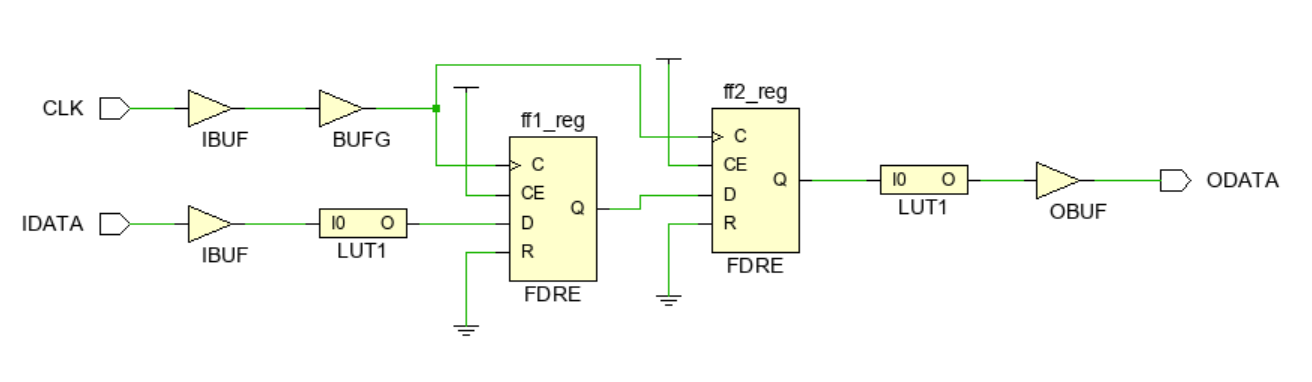


Рисунок 2. Схема проекта.

Данный проект не имеет какой-либо практической ценности, однако его примере можно продемонстрировать, как проводится временной анализ для входных сигналов 74HC595.

1. **Временной анализ для входных сигналов.**

Анализ передачи данных между микросхемой 74HC595 и FPGA мало отличается от анализа для двух триггеров внутри FPGA. Особенностью является то, что запускающий триггер располагается в одном, а защелкивающий в другом. На рисунке 3 представлен анализируемый путь, на который нанесены задержки для данных и тактового сигнала.

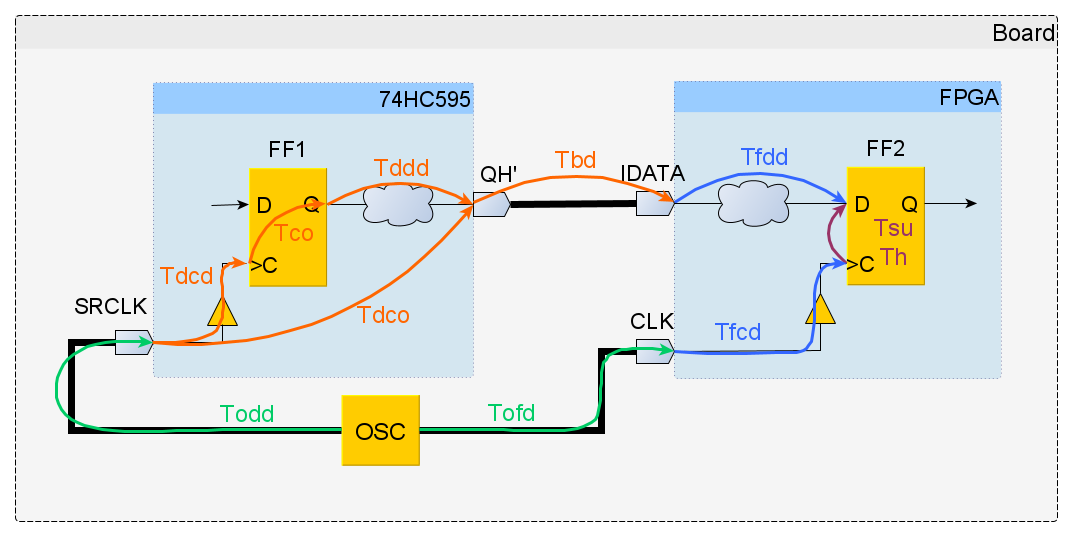


Рисунок 3. Путь с задержками для входных данных и тактового сигнала.

Ниже даны определения задержек, представленных на рисунке 3.

* (***O****sc to* ***D****evice* ***D****elay*) – задержка тактового сигнала от генератора до ножки микросхемы 74HC595;
* (***O****sc to* ***F****PGA* ***D****elay*) – задержка тактового сигнала от генератора до ножки FPGA;
* (***D****evice* ***C****lock* ***D****elay*) – задержка тактового сигнала от ножки 74HC595 до тактового входа 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 между триггером и ножкой микросхемы 74HC595;
* (***B****oard* ***D****elay*) – задержка распространения данных по дорожкам на плате;
* (***F****PGA* ***C****lock* ***D****elay*) – задержка тактового сигнала от ножки FPGA до тактового входа триггера;
* (***F****PGA* ***D****ata* ***D****elay*) – задержка распространения данных между триггером и ножкой FPGA;
* (***S****et****U****p time*) – время установки.
* (***H****old time*) – время удержания.

Период тактового сигнала будем обозначать как *Tclk*. Красным и зеленым цветом на рисунке 3 представлены задержки, которые неизвестны и которые необходимо указать для успешного проведения временного анализа.

1. **Максимальное время распространения**

Для начала рассмотрим, каким образом выполняется анализ для проверки ограничения на максимальное время распространения (*Setup*). Проделаем те же шаги, что и для анализа передачи между двумя триггерами, который был проведен в предыдущей статье [1].

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 рассчитаем время прибытия данных на вход защелкивающего триггера, который располагается внутри FPGA. Глядя на рисунок 3, можно получить следующие уравнения:

* Время прибытия фронта к запускающему триггеру в микросхеме 75HC595 (***S****ource* ***С****lock* ***A****rrival time*):
* Задержка распространения данных (***D****ata* ***D****elay*):
* Время прибытия данных на вход защелкивающего триггера в 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’.С учетом этого уравнение для ***D****ata* ***A****rrival time* можно записать в виде:

Далее 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 (***D****estination* ***C****lock* ***A****rrival time*):
* Требуемое время прибытия данных (***D****ata* ***R****equired time*):

В предыдущем уравнении учитывается, что данные должны прийти на вход защелкивающего триггера на время *Tsu* раньше защелкивающего фронта.

Запаса для задержки данных (*Slack*) вычисляется по формуле

Напомним, что положительный *Slack* указывает на то, что данные доходят до места назначения раньше, чем это требуется. Отрицательное значение *Slack* означает нарушение ограничения по *Setup*. 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*). Напомним, что анализ по *Hold* проводится для самого пессимистичного случая, при котором запускающий фронт и данные распространяются наиболее быстро, а защелкивающий фронт – максимально медленно.

Глядя на рисунок 3, можно рассчитать время прибытия данных на вход защелкивающего триггера:

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

В данном уравнении учитывается введенное ранее обозначение:

Далее представлены уравнения для требуемого времени прибытия данных.

* Время прибытия фронта к защелкивающему триггеру (***D****estination* ***C****lock* ***A****rrival time*):

Защелкивающий фронт для предыдущих данных появляется в тот же момент времени, что и запускающий фронт для следующих данных, поэтому к задержке распространения период тактового сигнала не добавляется.

* Требуемое время прибытия данных (***D****ata* ***R****equired time*):

Слагаемое в уравнении для учитывает, что данные не должны изменяться в течении времени удержания после защелкивающего фронта.

При анализе по *Hold* выражение для вычисления *Slack* имеет вид:

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

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

Рассмотрим первый способ создания временных ограничений для входного сигнала IDATA. Для начала рассмотрим создание ограничений по Setup. В уравнении (10) слагаемые, выделенные зеленым и оранжевым цветом, являются неизвестными для Vivado, так как описывают задержки вне FPGA. Также неизвестным является период тактового сигнала .

Создание ограничений на период тактового сигнала выполняется с помощью create\_clock и было рассмотрено в предыдущей статье [1]. Будем считать, что микросхема 74HC595 питается от источника питания с напряжением 4.5 В. Тогда в таблице 7.6 из datasheet для 74HC595 можно увидеть, что максимальная рабочая частота равна 31 МГц. Пускай требуется, что вся плата могла работать на частоте 10 МГц, тогда ограничение на тактовый сигнал можно задать в виде:

# ограничение на период тактового сигнала

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

В данном случае объявляется тактовый сигнал с именем clk\_10MHz, период которого равен 100 наносекунд. Ножка FPGA, на которую поступает тактовый сигнал называется CLK.

Все оставшиеся неизвестные в уравнении (1) можно объединить в одно выражение, которое будет иметь вид:

Тогда уравнение для можно записать в виде:

Слагаемые , и задают задержки сигнала из-за распространения по дорожкам печатной платы. Данные задержки зависят материала подложки, от длины дорожки и ее типа (полосковая, микрополосковая и т.д.). Для каждого типа дорожки существуют приближенные выражения для оценки скорость распространения сигнала. Зная скорость и длину дорожки, можно оценить задержку распространения сигнала. Пример приближенного расчета для полосковых и микрополосковых линий показан в [1].

Будем считать, что мы смогли оценить минимальные и максимальные задержки распространяя по дорожкам печатной платы и их значения записаны в xdc-файл:

# 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

Задержка , соответствует интервалу времени между приходом фронта на тактовый вход микросхемы 75HC595 и появлением данных на ее выходе QH’. Значение для данной задержки можно найти в таблице 7.7 из datasheet для 74HC595. Оно обозначено как и при напряжении питания 4.5 В может изменяться от 17 до 32 наносекунд. Данные задержки также внесем 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.

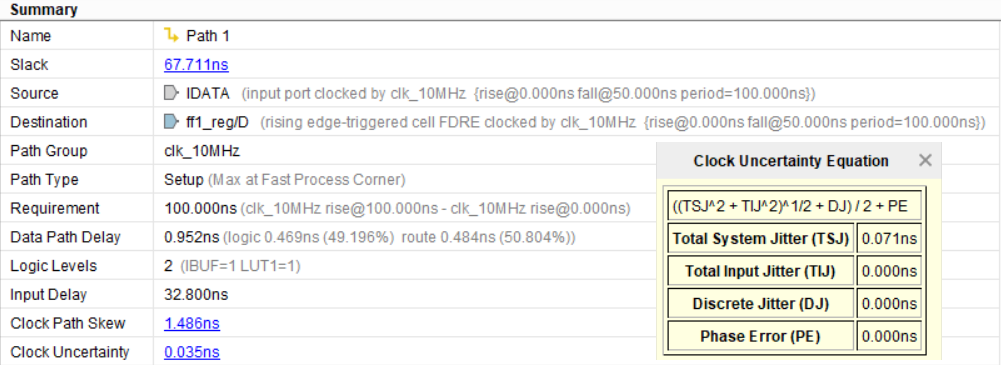
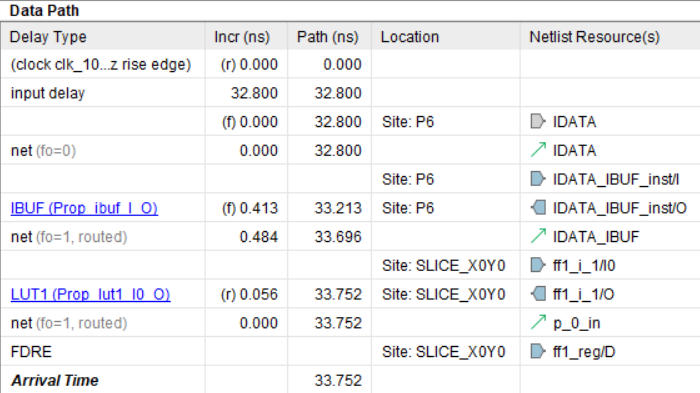
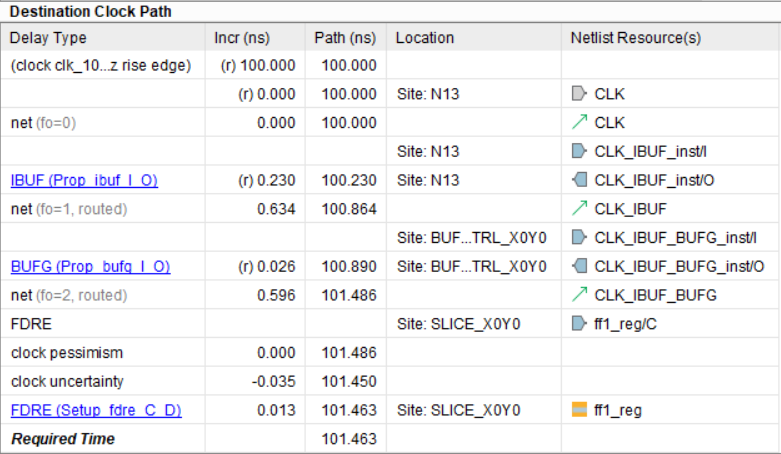
**

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.

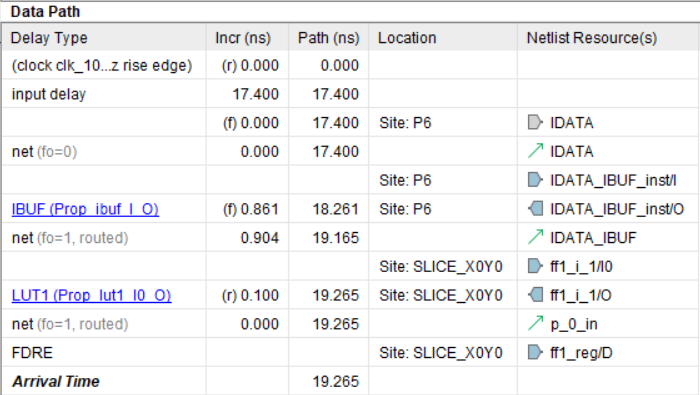
Поступление этого фронта запустит данные триггера внутри 74HC595. Спустя задержку нс., то есть в момент времени 32.8 нс. эти данные попадут на ножку FPGA. На рисунке 5 этой задержке соответствует параметр Input Delay во второй строке. Внутри FPGA данные проходят через входной буфер и LUT и попадают на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

Отсюда Slack равен нс.

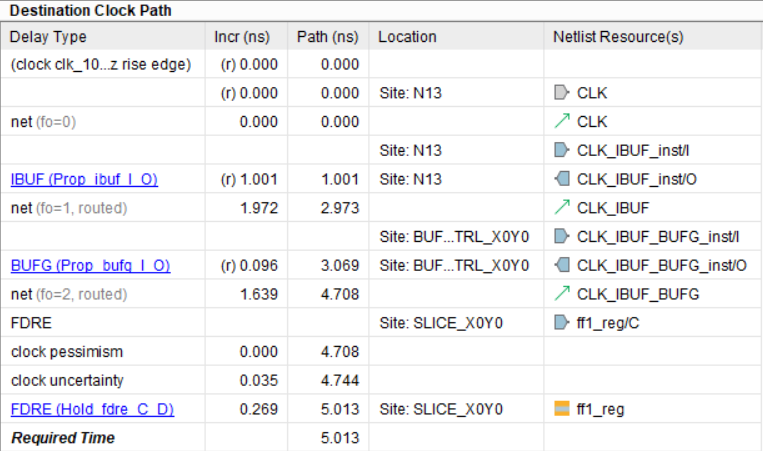
Аналогично можно рассмотреть результаты анализа по Hold, представленные на рисунках 7 и 8.

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 6. Расчет фактического времени прибытия данных.

**

Рисунках 7. 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]

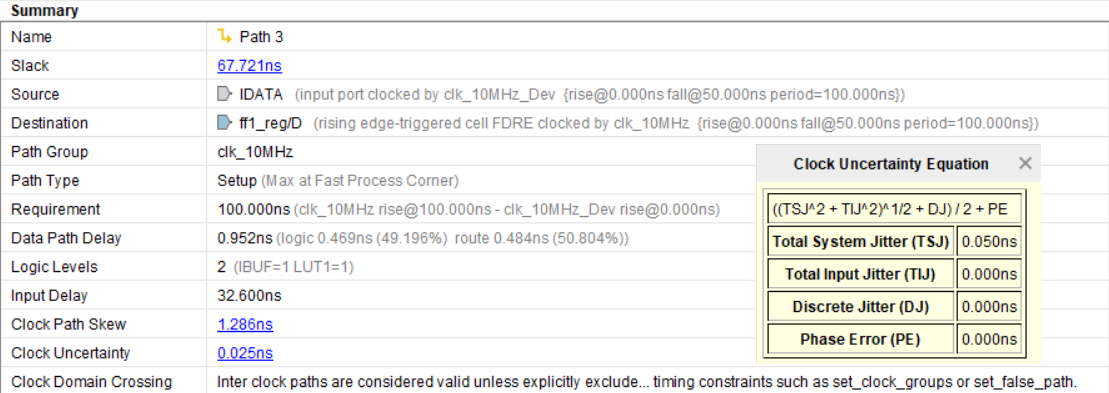
# временные ограничение для входного сигнала IDATA

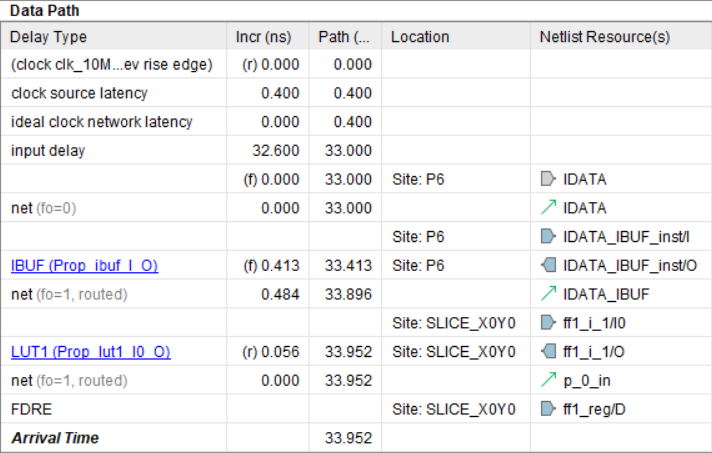
set idelay\_max [expr $Tdco\_max + $Tbd\_max]

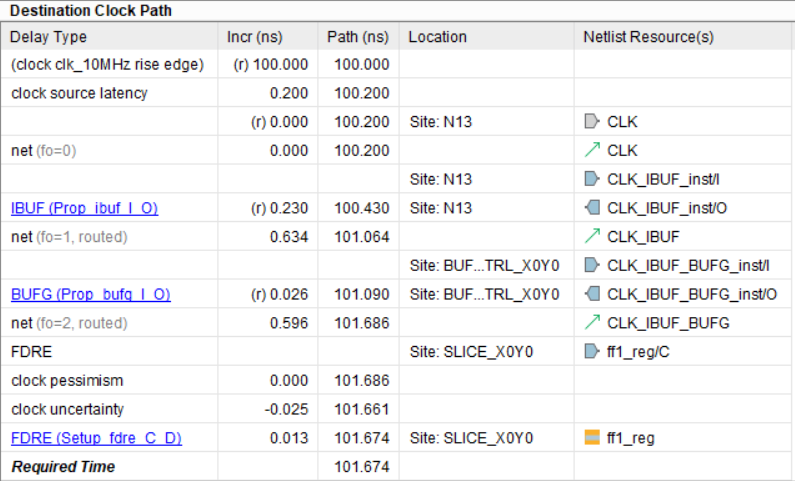
set\_input\_delay -clock clk\_10MHz\_Dev -max $idelay\_max [get\_ports IDATA];

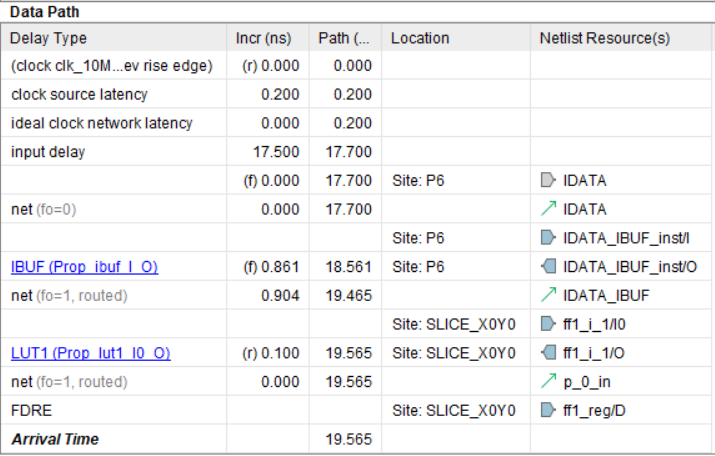
set idelay\_min [expr $Tdco\_min + $Tbd\_min]

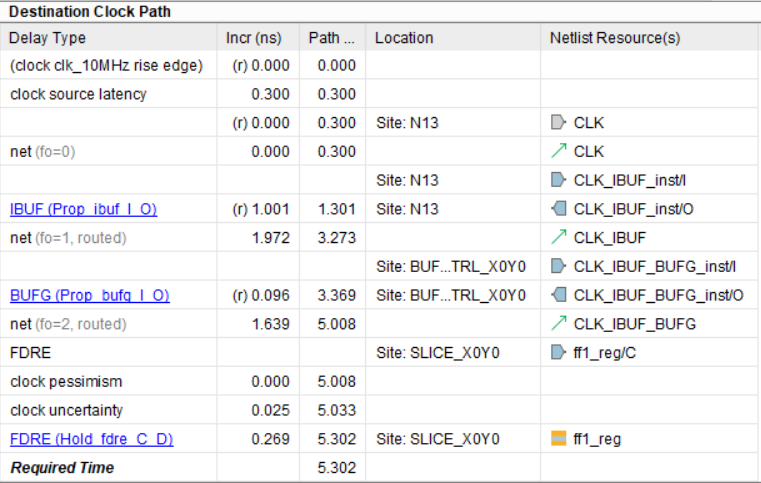
set\_input\_delay -clock clk\_10MHz\_Dev -min $idelay\_min [get\_ports IDATA];

**

**

**

**

**

# 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];

**Заключение.**

**Ссылки.**