**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.

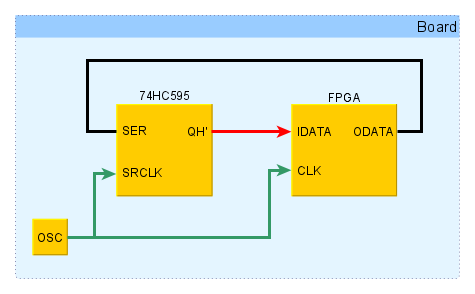


Рисунок 1. Схема соединения устройств на плате.

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

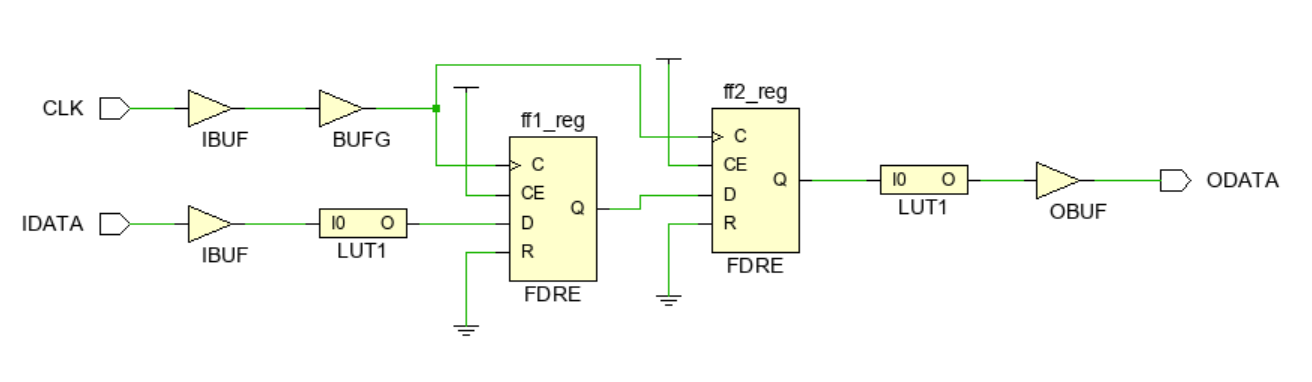


Рисунок 2. Схема проекта.

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.

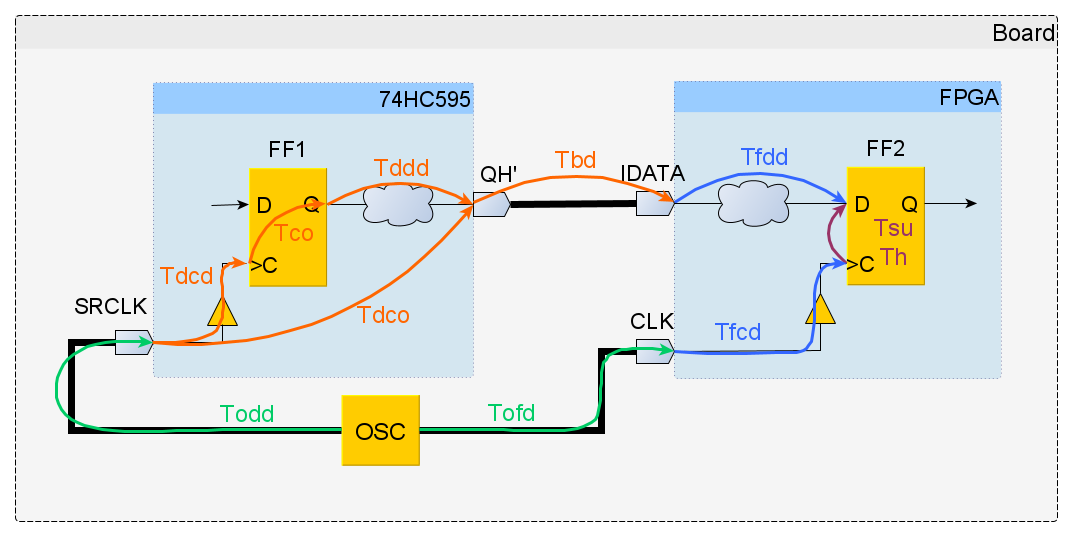


Рисунок 3. Путь с задержками для входных данных и тактового сигнала.

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.**

# 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]

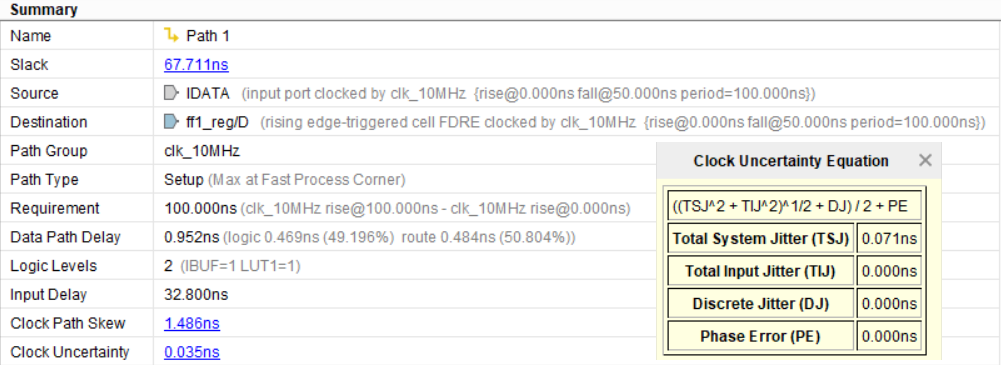
# временные ограничение для входного сигнала IDATA

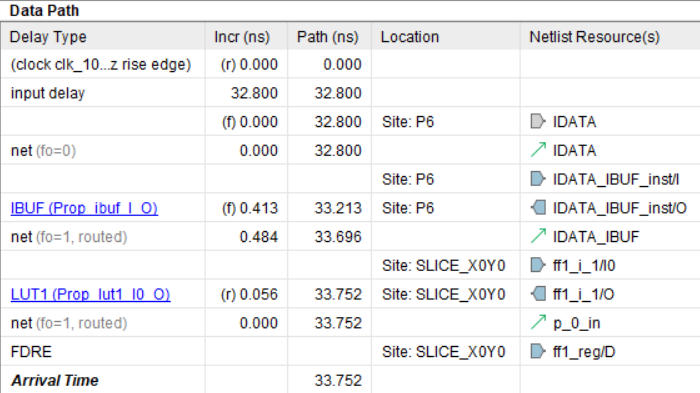
set idelay\_max [expr $Todd\_max - $Tofd\_min + $Tdco\_max + $Tbd\_max]

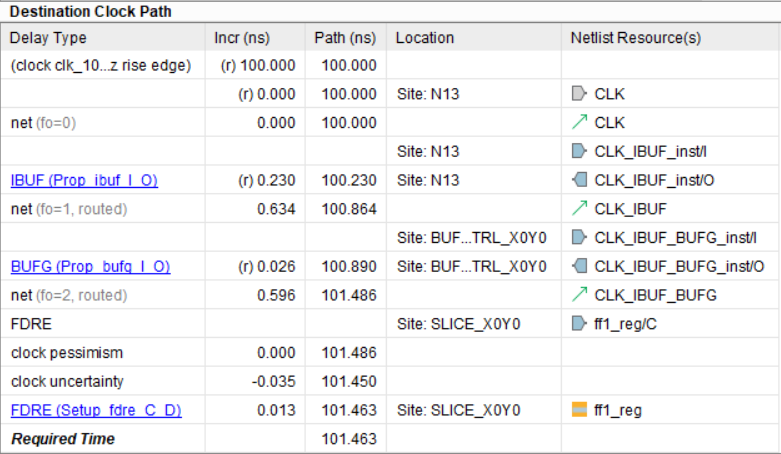
set\_input\_delay -clock clk\_10MHz -max $idelay\_max [get\_ports IDATA];

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

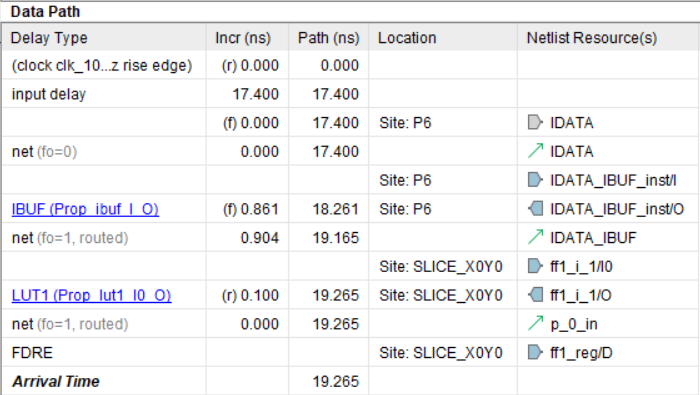
set\_input\_delay -clock clk\_10MHz -min $idelay\_min [get\_ports IDATA];

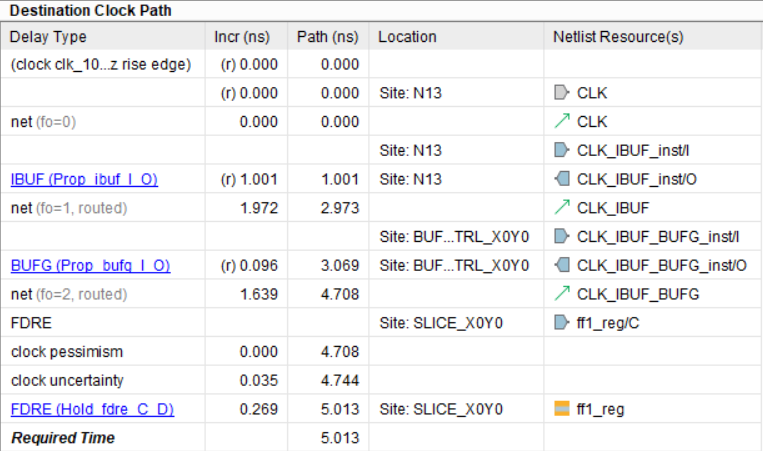
**

**

**

*Setup*

**

**

*Hold*

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

# 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

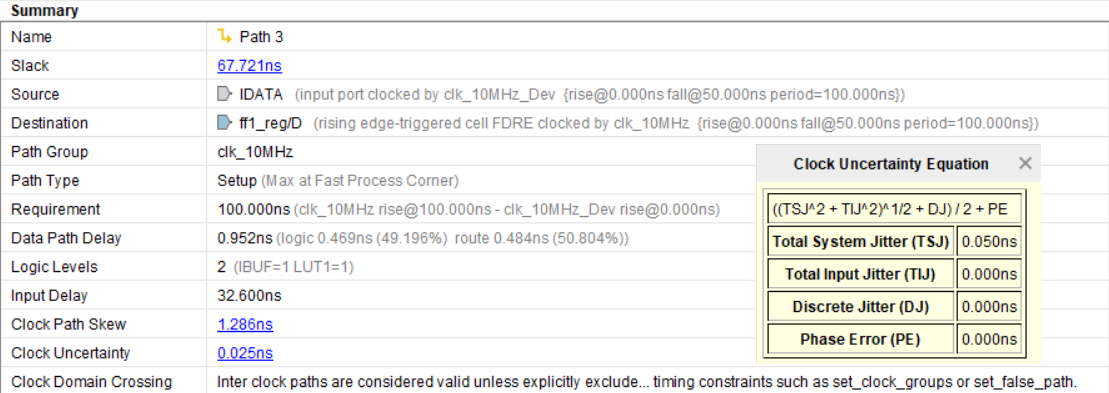
set\_clock\_latency -source -early $Todd\_min [get\_clocks clk\_10MHz\_Dev]

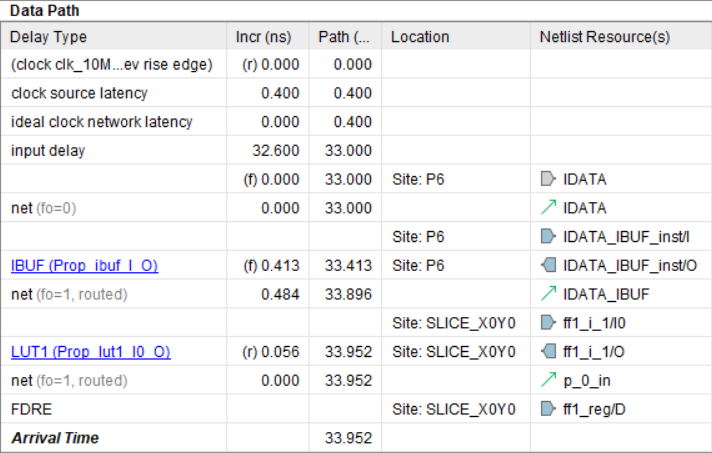
set\_clock\_latency -source -late $Todd\_max [get\_clocks clk\_10MHz\_Dev]

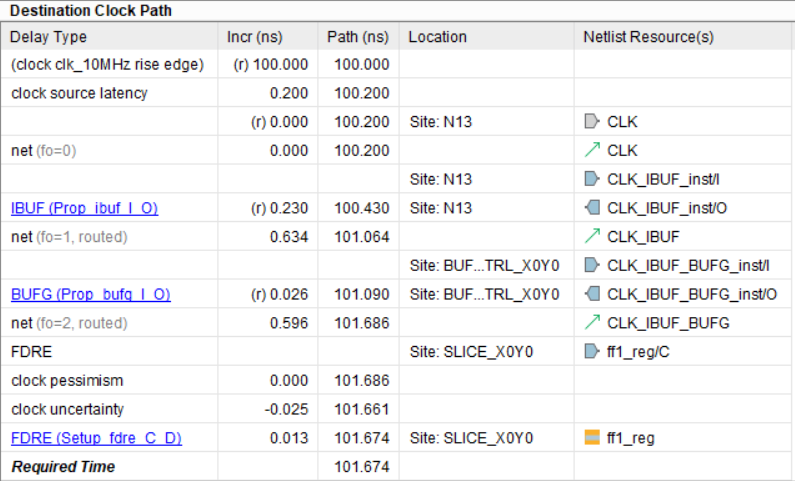
# временные ограничение для входного сигнала IDATA

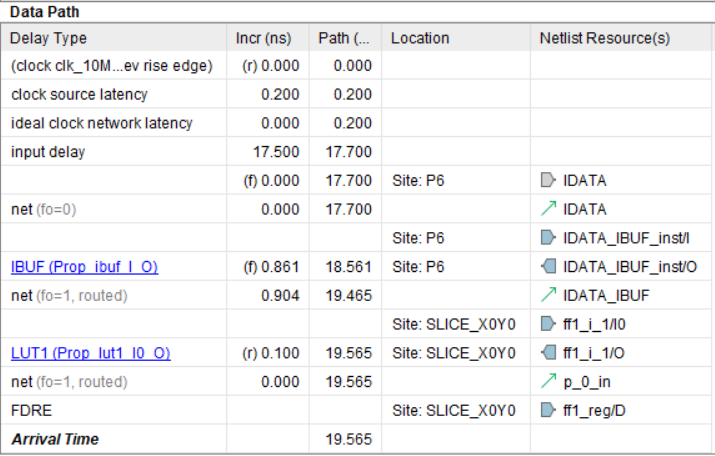
set\_input\_delay -clock clk\_10MHz\_Dev -max [expr $Tdco\_max + $Tbd\_max] [get\_ports IDATA];

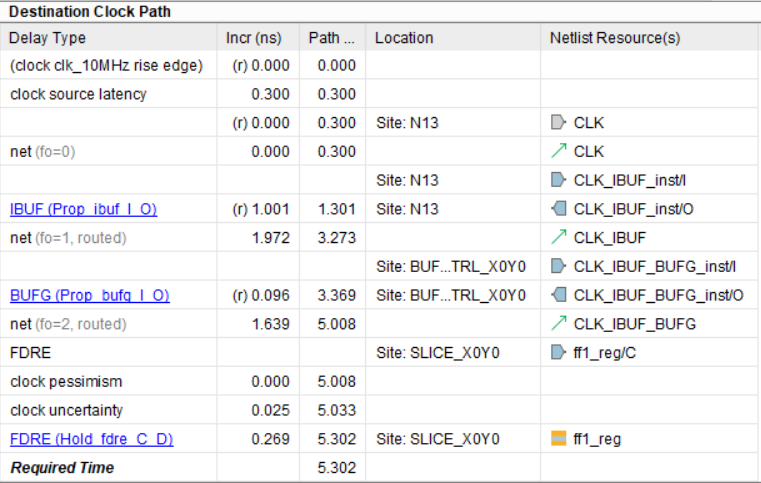
set\_input\_delay -clock clk\_10MHz\_Dev -min [expr $Tdco\_min + $Tbd\_min] [get\_ports IDATA];

**

**

**

**

**

# 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.**