**Основы статического временного анализа.**

**Часть 2.2: System Synchronous** **Output Delay Constraint.**

**Введение.**

Данная статья является продолжением серии статей по временным ограничениям в FPGA. Главная цель – познакомить начинающих разработчиков с основами статического временного анализа. Далее будет рассмотрен анализ при передачи данных в FPGA из внешнего устройства и показано два способа создания ограничений для выходных сигналов.

1. **Цель временных ограничений для выходных сигналов.**

В цифровых синхронных устройствах данные передаются между двумя триггерами, которые разделены комбинационной логикой. Ранее в [1] был представлен временной анализ для входных сигналов. В данной статье будут рассмотрены ограничения для выходных сигналов. Часть параграфов будет совпадать с материалом, представленным в статье [1]. Это сделано намеренно для уменьшения количества ссылок и, чтобы данные работы можно было читать независимо друг от друга.

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

В качестве примера будем рассматривать устройство, схема которого показана рисунке 1. Помимо FPGA на плате присутствует микросхема 74HC595 [3], которая представляет из себя обычный регистр сдвига и часто поставляется в составе обучающих наборов для Arduino. Также на плате располагается генератор (OSC), формирующий тактовый сигнал   
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 данных также формирует тактовый сигнал, называется Source Synchronous.

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

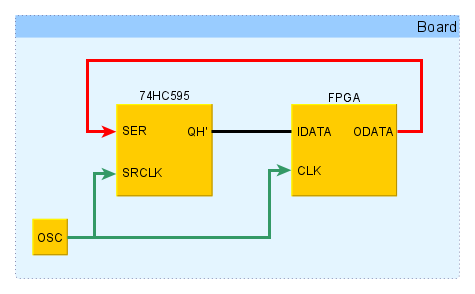


Рисунок 1. Схема соединения устройств на плате.

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

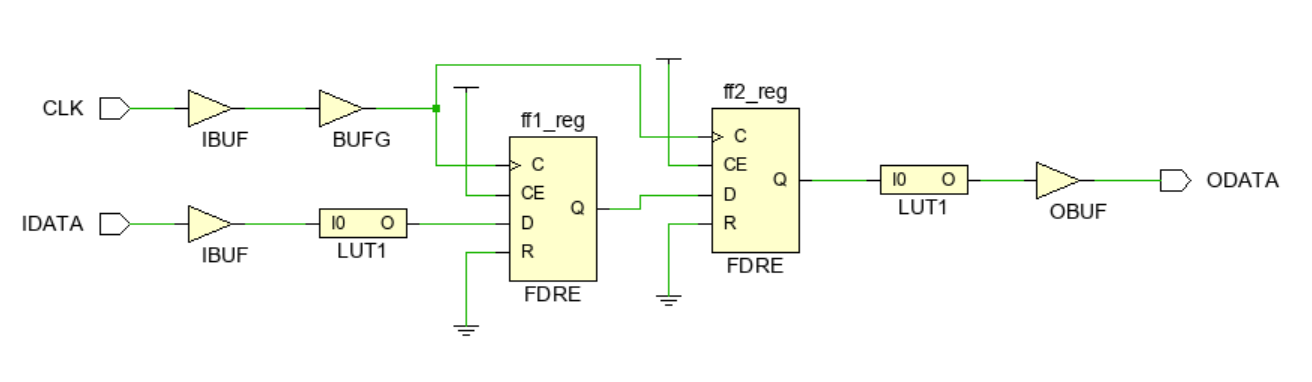


Рисунок 2. Схема FPGA проекта.

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 выходных сигналов.

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

Анализ передачи данных между FPGA и Device мало отличается от случая, рассмотренного в [2] 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.

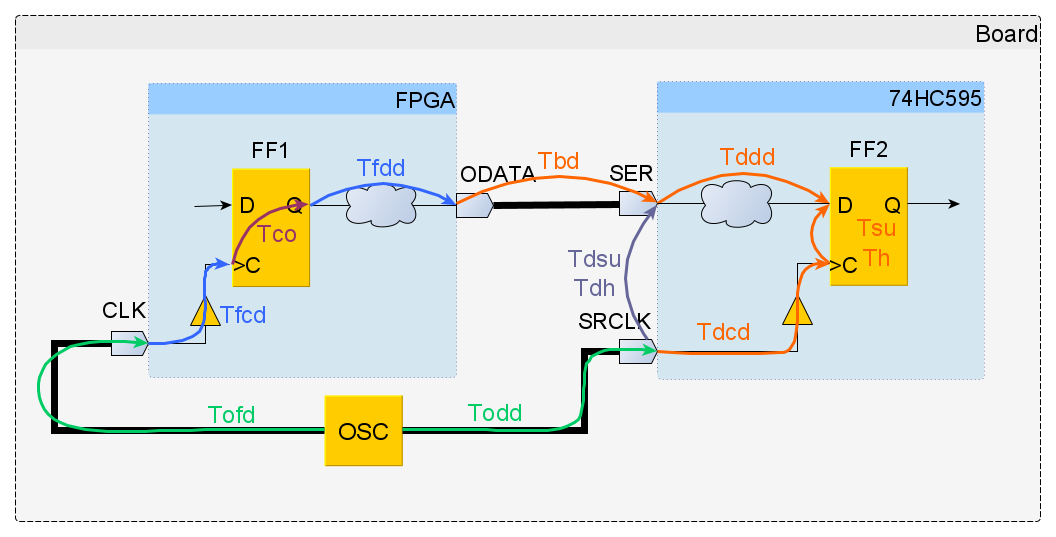


Рисунок 3. Путь с задержками для выходных данных и тактового сигнала.

Ниже даны определения задержек, представленных на рисунке 3.

* *Todd* (***O****sc to* ***D****evice* ***D****elay*) – задержка тактового сигнала от генератора до ножки SRCLK микросхемы 74HC595;
* *Tofd* (***O****sc to* ***F****PGA* ***D****elay*) – задержка тактового сигнала от генератора до ножки CLK FPGA;
* *Tdcd* (***D****evice* ***C****lock* ***D****elay*) – задержка тактового сигнала от ножки SRCLK микросхемы 74HC595 до тактового входа защелкивающего триггера;
* *Tbd* (***B****oard* ***D****elay*) – задержка распространения данных по дорожкам платы между FPGA и микросхемой 74HC595;
* *Tddd* (***D****evice* ***D****ata* ***D****elay*) – задержка распространения данных от ножки SER микросхемы 74HC595 до защелкивающего триггера;
* *Tsu* (***S****et****U****p time*) – время установки защелкивающего триггера;
* *Th* (***H****old time*) – время удержания защелкивающего триггера;
* *Tfcd* (***F****PGA* ***C****lock* ***D****elay*) – задержка тактового сигнала от ножки CLK FPGA до тактового входа запускающего триггера;
* *Tfdd* (***F****PGA* ***D****ata* ***D****elay*) – задержка распространения данных от запускающего триггера до ножки ODATA FPGA;
* *Tco* (***C****lock to* ***O****utput*) – интервал времени между приходом фронта на тактовый вход триггера и появлением данных на его выходе Q.

Период тактового сигнала будем обозначать *Tclk*. Оранжевым и зеленым цветом на рисунке 3 представлены задержки для участков пути, которые располагаются вне FPGA. Данные задержки необходимо 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 защелкивающего триггера, расположенного внутри микросхемы 74HC595, считая, что запускающий фронт появляется в нулевой момент времени. Уравнения для расчета представлены ниже (см. рисунок 3):

* 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 внутри микросхемы 74HC595 (***D****ata* ***A****rrival time*):

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

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

* Время прибытия фронта к защелкивающему триггеру внутри микросхемы 74HC595 (***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.

При анализе по *Setup* величина запаса (*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 записать полное выражение для расчета *Slack*:

Введем обозначение *Tdsu* (***D****evice* ***S****et****U****p time*):

Задержка *Tdsu* соответствует времени установки для сигнала на   
входе SER микросхемы 75HC595 относительно тактового входа SRCLK. Тогда уравнение для *Slack* можно записать в виде:

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 запускающему триггеру   
  (***S****ource* ***С****lock* ***A****rrival time*):
* Задержка распространения данных (***D****ata* ***D****elay*):
* Время прибытия данных на вход защелкивающего триггера внутри микросхемы 75HC595 (***D****ata* ***A****rrival time*):

Далее представлены уравнения для вычисления требуемого времени прибытия данных:

* Время прибытия фронта к защелкивающему триггеру внутри микросхемы 75HC595 (***D****estination* ***C****lock* ***A****rrival time*):
* Требуемое время прибытия данных (***D****ata* ***R****equired time*):

Защелкивающий фронт для предыдущих данных появляется в тот же момент времени, что и запускающий фронт для следующих данных. По этой причине в задержке распространения *Tdca* отсутствует слагаемое, равное периоду тактового сигнала.

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

Введем обозначение *Tdh* (***D****evice* ***H****old time*):

Задержка *Tdh* соответствует времени удержания для сигнала на   
входе SER микросхемы 75HC595 относительно тактового входа SRCLK. То есть, данные на входе SER не должны изменяться в течении времени *Tdh* после защелкивающего фронта.

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

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

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

В уравнении (1) слагаемые, выделенные зеленым и оранжевым цветом, неизвестны анализатору Vivado, так как они описывают задержки для участков пути вне FPGA. Также неизвестным является значение периода *Tclk*. Создание ограничений на период тактового сигнала с помощью команды create\_clock было рассмотрено в [2].

Для определенности будем считать, что напряжение источника питания микросхемы 74HC595 равно 4.5 В. Из таблицы 6.6 datasheet для 74HC595 [3] находим, что в этом случае максимальная рабочая частота равна 31 МГц. Пусть требуется, чтобы плата на рисунке 1 могла работать при частота тактового генератора (OSC) равной 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 объявляет тактовый сигнал clk\_10MHz с периодом 100 нс, который поступает в FPGA через ножку CLK.

Объединим все оставшиеся неизвестные слагаемые из уравнения (1) в одну переменную *output\_delay\_max*. Тогда выражение для *Slack* можно переписать в виде:

где

В уравнении (3)слагаемые *Todd\_max*, *Tofd\_min* и *Tbd\_max* описывают задержки, обусловленные распространением сигнала по дорожкам печатной платы. Их значения зависят от многих факторов, например, материала подложки, длины дорожек, типа дорожек (полосковые,   
микрополосковые и т.д.).

Для каждого типа дорожек существуют приближенные выражения для вычисления скорости распространения сигнала. Зная скорость сигнала и длину дорожки, можно оценить задержку распространения. Примеры приближенных расчетов для различных типов дорожек можно найти в [4].

Будем считать, что мы смогли оценить минимальные и максимальные задержки распространения сигналов по дорожкам печатной платы. В качестве примера примем следующие значения в наносекундах: *Tbd\_max* = 0.6,  
 *Tbd\_min* = 0.5, *Todd\_max* = 0.4, *Todd\_min* = 0.2, *Tofd\_max* = 0.3 и   
*Tofd\_min* = 0.2. 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 до FPGA

set Tofd\_max 0.3

set Tofd\_min 0.2

Рассмотрим задержку *Tdsu* из уравнения (3), которая соответствует времени установки для сигнала на входе SER микросхемы 75HC595 относительно тактового входа SRCLK. Ее значение можно получить из таблицы 7.6 datasheet для 74HC595 [3]. Данная задержка обозначена как *tsu* и при напряжении питания 4.5 В равна 20 нс. Это значение также записывается в xdc-файл:

# время установки для сигнала на входе SER микросхемы 74HC595

# относительно тактового входа SRCLK

set Tdsu 20

Теперь можно создать временное ограничение на выходной сигнал для анализа по *Setup*. Для этого в файл с ограничениями нужно внести следующие команды [5]:

# временное ограничение для выходного сигнала ODATA

set odelay\_max [expr $Tofd\_max - $Todd\_min + $Tbd\_max + $Tdsu]

set\_output\_delay -clock clk\_10MHz -max $odelay\_max [get\_ports ODATA];

В первой строке объявлена переменная odelay\_max, значение которой приравнивается *output\_delay\_max* из уравнения (3). Далее с помощью команды set\_output\_delay создается ограничение для выходного сигнала.   
Опция -max $odelay\_max задает задержку для анализа по *Setup*. Конструкция [get\_ports ODATA] указывает, что ограничение накладывается на выходной сигнал, выдаваемый из FPGA через ножку ODATA.

Важно отметить, что анализатору Vivado необходимо указать, каким сигналом тактируется защелкивающий триггер, так как он находится   
вне FPGA. Это делается с помощью опции -clock clk\_10MHz. Данный тактовый сигнал был создан ранее с помощью команды create\_clock. Запускающий триггер располагается внутри FPGA, поэтому временному анализатору его тактовый сигнал известен.

Аналогичным образом создаются ограничения для анализа по *Hold*. Объединив все неизвестные слагаемые в уравнении (2) в одну переменную, выражение для *Slack* можно записать в виде:

где

Значение задержки *Tdh* можно получить из таблицы 7.6 datasheet для 74HC595 [3]. Данная задержка обозначена как *th* и равна 0 нс. Ее также следует записать в xdc-файл:

# время удержания для сигнала на входе SER микросхемы 74HC595

# относительно тактового входа SRCLK

set Tdh 0

Команды, которые требуется добавить в xdc-файл, представлены ниже:

# временное ограничение для выходного сигнала ODATA

set odelay\_min [expr $Tofd\_min - $Todd\_max + $Tbd\_min - $Tdh]

set\_output\_delay -clock clk\_10MHz -min $odelay\_min [get\_ports ODATA];

Как и ранее, сначала объявляется переменная, значение которой равно *output\_delay\_min*, после чего с помощью команды set\_output\_delay создается временное ограничение. Опция –min указывает, что ограничение предназначено для проведения анализа по *Hold*.

Полное содержимое xdc-файла представлено ниже:

# время установки для сигнала на входе SER микросхемы 74HC595

# относительно тактового входа SRCLK

set Tdsu 20

# время удержания для сигнала на входе SER микросхемы 74HC595

# относительно тактового входа SRCLK

set Tdh 0

# 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

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

# временное ограничение для выходного сигнала ODATA

set odelay\_max [expr $Tofd\_max - $Todd\_min + $Tbd\_max + $Tdsu]

set\_output\_delay -clock clk\_10MHz -max $odelay\_max [get\_ports ODATA];

set odelay\_min [expr $Tofd\_min - $Todd\_max + $Tbd\_min - $Tdh]

set\_output\_delay -clock clk\_10MHz -min $odelay\_min [get\_ports ODATA];

Рассмотрим, как введенные ограничения будут отражены во временных отчетах, полученных после размещения и трассировки проекта. На рисунке 4 представлен раздел *Summary* для анализа по *Setup*, в котором указан запускающий триггер (ff2\_reg), выходной порт FPGA (ножка ODATA), задержка данных внутри FPGA (Data Path Delay) и количество уровней логики   
(Logic Levels). Также можно увидеть полученный *Slack*, расфазировку   
(Clock Path Skew) и неопределенность (Clock Uncertainty) тактового сигнала.

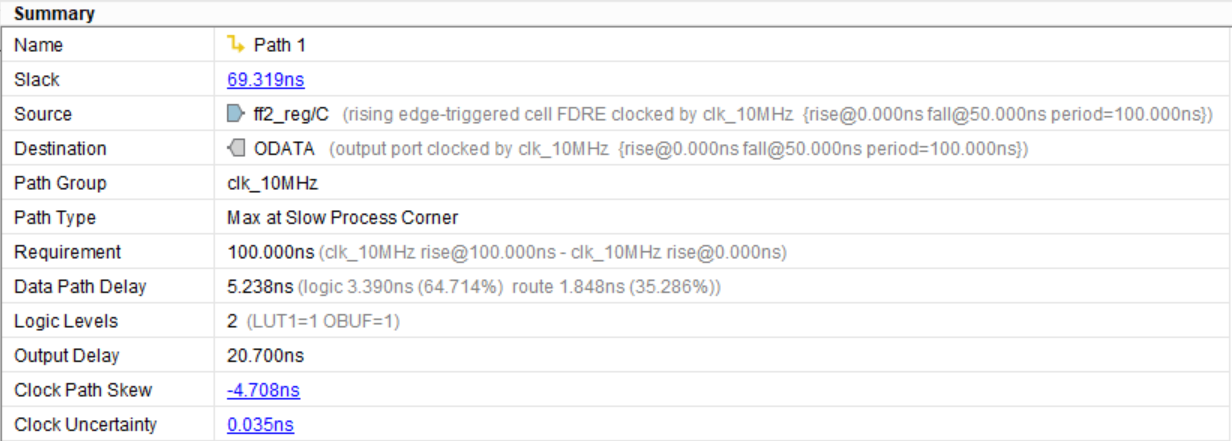
**

Рисунок 4. Общие сведения о входном пути (*Setup*).

На рисунке 5 показаны задержки для запускающего фронта, защелкивающего фронта и данных. Представленные результаты можно интерпретировать следующим образом. Из раздела *Source Clock Path* получаем, что защелкивающий фронт приходит на ножку CLK FPGA в нулевой момент времени. Далее он распространяется через входной и тактовый буферы, попадает на запускающий триггер и запускает передачу данных через 4.708 нс. В свою очередь данные проходят через LUT1 и выходной буфер и появляются в выходной ножке FPGA в момент   
времени 9.946 нс. Эти задержки показаны в разделе *Data Path*.

Из раздела *Destination Clock Path* видно, что спустя один период тактового сигнала, равный 100 нс, защелкивающий фронт попадает на вход триггера внутри микросхемы 74HC595. Задержка распространения данных вне FPGA учитывается c помощью задержки output delay. Ее значение вычисляется по формуле (3) и равно 0.3 - 0.2 + 0.6 + 20 = 20.7нс. Знак минус указывает, что данная задержка при расчете *Slack* должна вычитаться из общей суммы.

Задержка *output delay* указывает, что данные должны появится выходной ножке FPGA на 20.7нс раньше, чем следующий тактовый фронт достигнет защелкивающего триггера. Таким образом, с учетом неопределенности тактового сигнала данные должны дойти до ножки ODATA в момент времени 100 - 20.7 *-* 0.035 = 79.265 нс.

Отсюда получаем, что *Slack* равен 79.265 - 9.946 = 69.319 нс, что соответствует значению, представленному на рисунке 4.

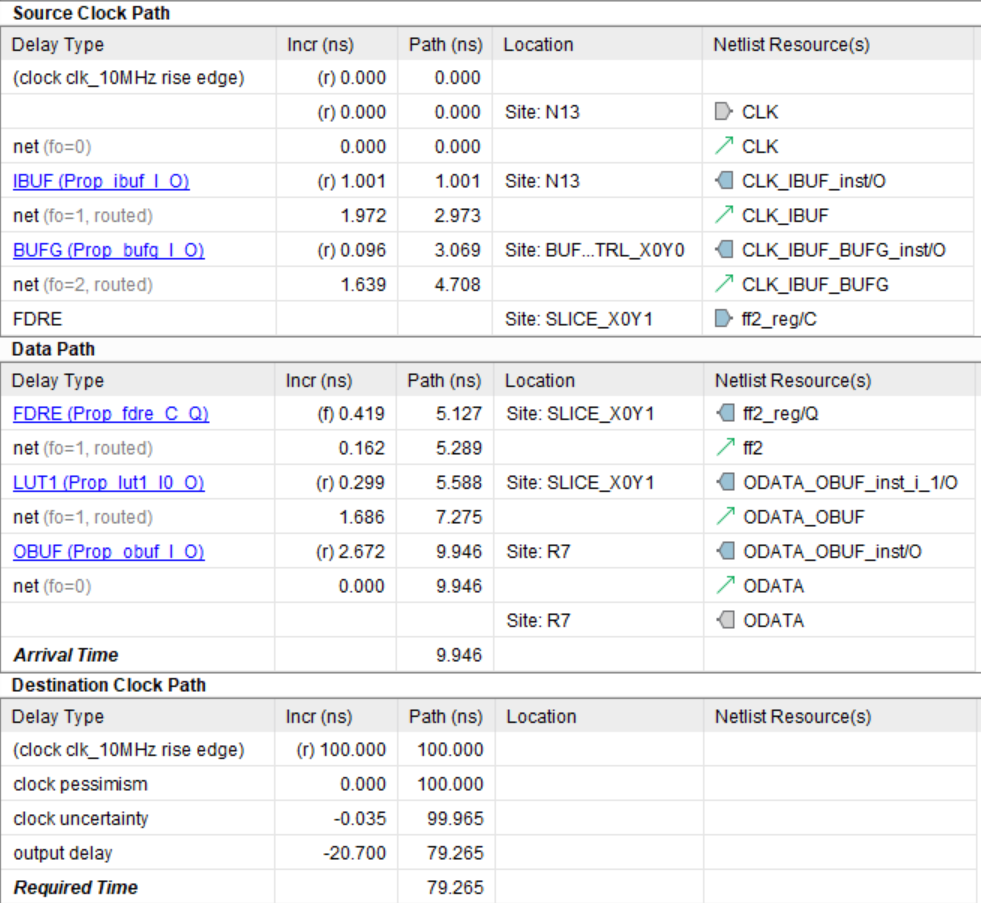
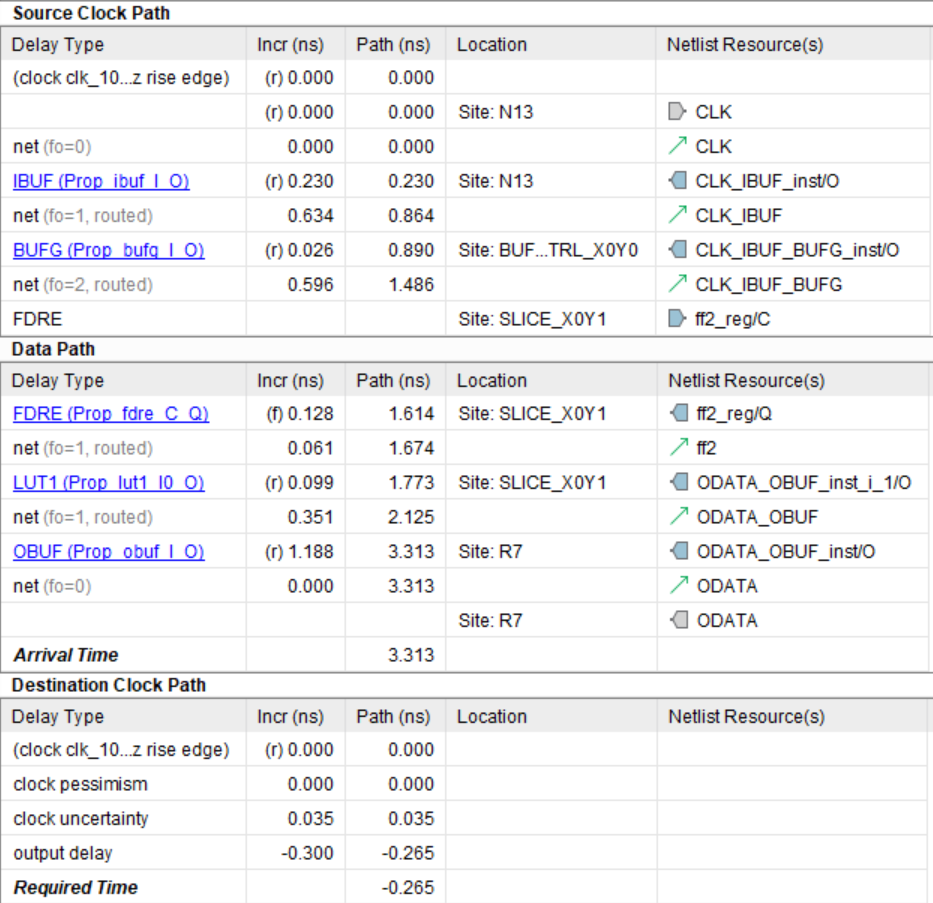
**

Рисунок 5. Задержки тактового сигнала и данных (*Setup*).

Ту же картину получаем для анализа по *Hold*. На рисунке 6 можно увидеть, что запускающий фронт достигает FPGA в нулевой момент времени, после чего спустя 3.313 нс данные появятся на ножке IDATA. Для анализа по *Hold* запускающий и защелкивающий фронты появляются одновременно,   
поэтому на вход SRCLK микросхемы 74HC595 фронт приходит также в нулевой в момент времени.

Значение задержки *output delay* вычисляется по формуле (4) и равно 0.2 - 0.4 + 0.5 - 0 = 0.3нс. Учитывая неопределенность тактового сигнала получаем, что данные должны появится на ножку FPGA на 0.265 нс позже защелкивающего тактового фронта.

**

Рисунках 6. Задержки тактового сигнала и данных (*Hold*).

Данный подход прост тем, что достаточно представить все неизвестные задержки в виде одной величины *output\_delay*. Объединение всех задержек в одну переменную имеет и негативный эффект, который заключается в менее наглядной форме представления результатов временного анализа.

1. **Output Delay Constraint в Vivado.**

**Заключение.**

**Ссылки.**