

# **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 передачи данных из FPGA во внешнее устройство 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 будут совпадать с материалом, представленным в [1]. Это сделано намеренно 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 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 на 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 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 иногда называть Device. Далее будет представлен временной анализ передачи данных с выхода ODATA FPGA на вход SER 74HC595. На рисунке 1 данный путь отмечен красным цветом.

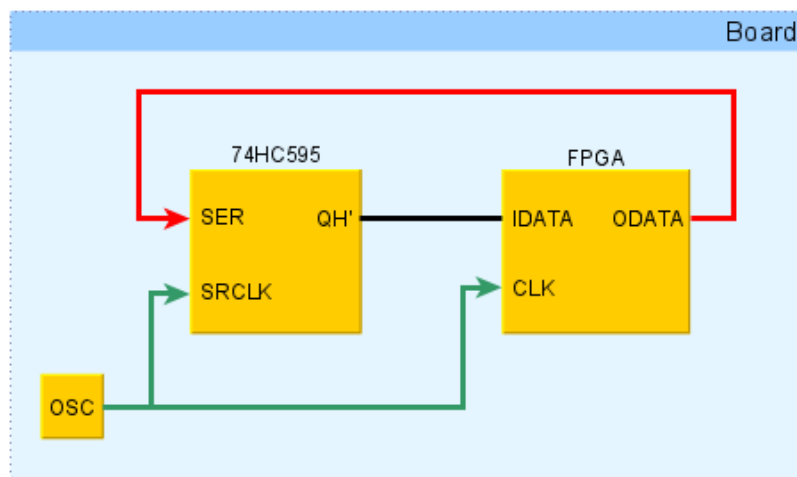


Рисунок 1. Схема соединения устройств на плате.

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

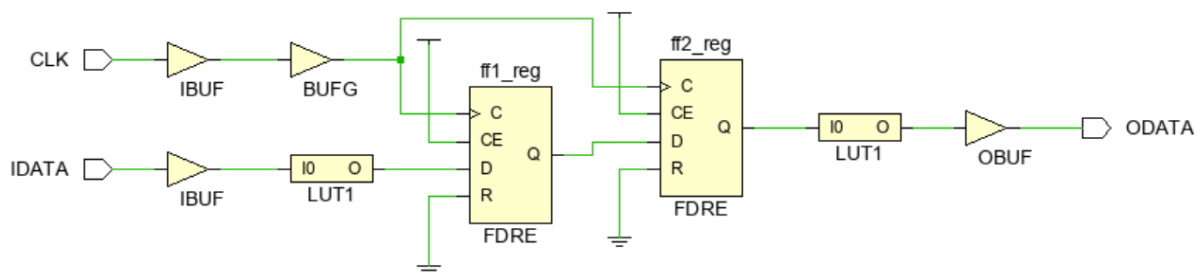


Рисунок 2. Схема FPGA проекта.

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.

## 2. 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.

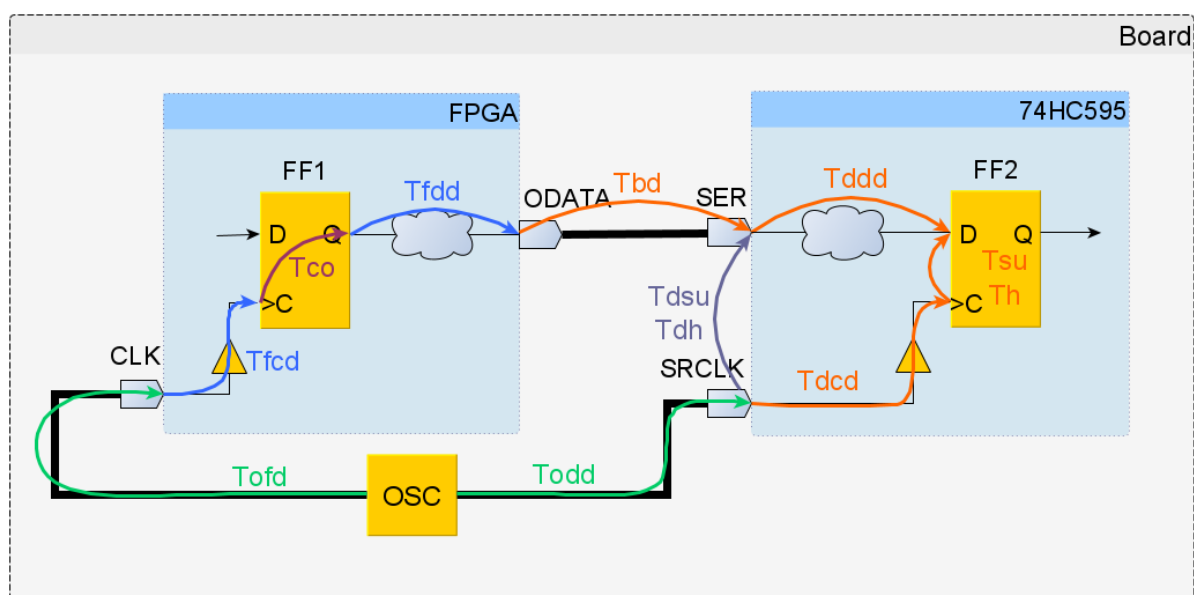


Рисунок 3. Путь с задержками для выходных данных и тактового сигнала.

Ниже даны определения задержек, представленных на рисунке 3.

- *Todd* (*Osc to Device Delay*) – задержка тактового сигнала от генератора до ножки SRCLK микросхемы 74HC595;
- *Tofd* (*Osc to FPGA Delay*) – задержка тактового сигнала от генератора до ножки CLK FPGA;
- *Tdcd* (*Device Clock Delay*) – задержка тактового сигнала от ножки SRCLK микросхемы 74HC595 до тактового входа защелкивающего триггера;
- *Tbd* (*Board Delay*) – задержка распространения данных по дорожкам платы между FPGA и микросхемой 74HC595;
- *Tddd* (*Device Data Delay*) – задержка распространения данных от ножки SER микросхемы 74HC595 до защелкивающего триггера;
- *Tsu* (*SetUp time*) – время установки защелкивающего триггера;
- *Th* (*Hold time*) – время удержания защелкивающего триггера;
- *Tfcd* (*FPGA Clock Delay*) – задержка тактового сигнала от ножки CLK FPGA до тактового входа запускающего триггера;
- *Tfdd* (*FPGA Data Delay*) – задержка распространения данных от запускающего триггера до ножки ODATA FPGA;
- *Tco* (*Clock to Output*) – 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.

### 3. 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. Будем считать, что запускающий фронт появляется в нулевой момент времени. Уравнения для расчета представлены ниже (см. рисунок 3):

- Время прибытия фронта к запускающему триггеру (*Source Clock Arrival time*):

$$T_{sca\_max} = T_{ofd\_max} + T_{fcd\_max}.$$

- Задержка распространения данных (*Data Delay*):

$$T_{dd\_max} = T_{co\_max} + T_{fdd\_max} + T_{bd\_max} + T_{ddd\_max}.$$

- Время прибытия данных на вход защелкивающего триггера внутри микросхемы 74HC595 (*Data Arrival time*):

$$T_{da\_max} = T_{sca\_max} + T_{dd\_max}.$$

Подставив предыдущие результаты в уравнение для  $T_{da\_max}$ , получим:

$$T_{da\_max} = T_{ofd\_max} + T_{fcd\_max} + T_{co\_max} + \\ + T_{fdd\_max} + T_{bd\_max} + T_{ddd\_max}.$$

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

- Время прибытия фронта к защелкивающему триггеру внутри микросхемы 74HC595 (*Destination Clock Arrival time*):

$$T_{dca\_min} = T_{odd\_min} + T_{dcd\_min} + T_{clk}.$$

- Требуемое время прибытия данных (*Data Required time*):

$$T_{dr\_min} = T_{dca\_min} - T_{su} = T_{odd\_min} + T_{dcd\_min} + T_{clk} - T_{su}.$$

В предыдущем уравнении учитывается, что данные на входе защелкивающего триггера должны быть стабильны в течении времени установки  $T_{su}$  до прихода фронта тактового сигнала.

При анализе по *Setup* величина запаса (*Slack*) вычисляется по формуле:

$$Slack = T_{dr\_min} - T_{da\_max}.$$

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

$$Slack = T_{clk} + T_{odd\_min} - T_{ofd\_max} - T_{co\_max} - T_{fcd\_max} - T_{fdd\_max} - T_{bd\_max} + T_{dcd\_min} - T_{ddd\_max} - T_{su}.$$

Введем обозначение *Tdsu* (*Device SetUp time*):

$$T_{dsu} = T_{su} + T_{ddd\_max} - T_{dcd\_min}.$$

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

$$Slack = T_{clk} + T_{odd\_min} - T_{ofd\_max} - T_{co\_max} - T_{fcd\_max} - T_{fdd\_max} - T_{bd\_max} - T_{dsu}. \quad (1)$$

#### 4. 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 (*Source Clock Arrival time*):

$$T_{sca\_min} = T_{ofd\_min} + T_{fcd\_min}.$$

- Задержка распространения данных (*Data Delay*):

$$T_{dd\_min} = T_{co\_min} + T_{fdd\_min} + T_{bd\_min} + T_{ddd\_min}.$$

- Время прибытия данных на вход защелкивающего триггера внутри микросхемы 75HC595 (*Data Arrival time*):

$$T_{da\_min} = T_{sca\_min} + T_{dd\_min} = T_{ofd\_min} + T_{fcd\_min} + T_{co\_min} + T_{fdd\_min} + T_{bd\_min} + T_{ddd\_min}.$$

Далее представлены уравнения для вычисления требуемого времени прибытия данных:

- Время прибытия фронта к защелкивающему триггеру внутри микросхемы 75HC595 (*Destination Clock Arrival time*):

$$Tdca\_max = Todd\_max + Tdcd\_max.$$

- Требуемое время прибытия данных (*Data Required time*):

$$Tdr\_max = Tdca\_max + Th = Todd\_max + Tdcd\_max + Th.$$

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

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

$$Slack = Tda\_min - Tdr\_max.$$

Введем обозначение  $Tdh$  (*Device Hold time*):

$$Tdh = Th + Tdcd\_max - Tddd\_min.$$

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

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

$$Slack = Tofd\_min - Todd\_max + Tco\_min + Tgcd\_min + Tfdd\_min + Tbd\_min - Tdh. \quad (2)$$

## 5. Первый способ создания временных ограничений 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  $T_{clk}$ . 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 частоте 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:

$$Slack = T_{clk} - output\_delay\_max - T_{co\_max} - T_{fcd\_max} - T_{fdd\_max},$$

где

$$output\_delay\_max = T_{ofd\_max} - T_{odd\_min} + T_{bd\_max} + T_{dsu}. \quad (3)$$

В уравнении (3) слагаемые  $T_{odd\_max}$ ,  $T_{ofd\_min}$  и  $T_{bd\_max}$  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  $t_{su}$  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:

$$Slack = output\_delay\_min + T_{co\_min} + T_{fcd\_min} + T_{fdd\_min},$$

где

$$output\_delay\_min = T_{ofd\_min} - T_{odd\_max} + T_{bd\_min} - T_{dh}. \quad (4)$$

Значение задержки  $T_{dh}$  можно получить из таблицы 7.6 datasheet для 74HC595 [3]. Данная задержка обозначена как  $t_h$  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, создающие временное ограничение, 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.

Summary	
Name	Path 1
Slack	69.319ns
Source	#f2_reg/C (rising edge-triggered cell FDRE clocked by clk_10MHz {rise@0.000ns fall@50.000ns period=100.000ns})
Destination	ODATA (output port clocked by clk_10MHz {rise@0.000ns fall@50.000ns period=100.000ns})
Path Group	clk_10MHz
Path Type	Max at Slow Process Corner
Requirement	100.000ns (clk_10MHz rise@100.000ns - clk_10MHz rise@0.000ns)
Data Path Delay	5.238ns (logic 3.390ns (64.714%) route 1.848ns (35.286%))
Logic Levels	2 (LUT1=1 OBUF=1)
Output Delay	20.700ns
Clock Path Skew	-4.708ns
Clock Uncertainty	0.035ns

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 через входной и тактовый буферы попадает на вход триггера и через 4.708 нс запускает передачу данных. В свою очередь данные проходят через LUT1 и выходной буфер и появляются на выходной ножке ODATA в момент времени 9.946 нс. Эта задержка показана на рисунке 5 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 74НС595. Время распространения данных вне FPGA учитывается с помощью задержки *output delay*. Ее значение вычисляется по формуле (3) и равно  $0.3 - 0.2 + 0.6 + 20 = 20.7$  нс.

Таким образом, с учетом неопределенности тактового сигнала в 0.035 нс 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.

Source Clock Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_10MHz rise edge)	(r) 0.000	0.000		
	(r) 0.000	0.000	Site: N13	CLK
net (fo=0)	0.000	0.000		CLK
<a href="#">IBUF (Prop_ibuf I O)</a>	(r) 1.001	1.001	Site: N13	CLK_IBUF_inst/O
net (fo=1, routed)	1.972	2.973		CLK_IBUF
<a href="#">BUFG (Prop_bufg I O)</a>	(r) 0.096	3.069	Site: BUF...TRL_X0Y0	CLK_IBUF_BUFG_inst/O
net (fo=2, routed)	1.639	4.708		CLK_IBUF_BUFG
FDRE			Site: SLICE_X0Y1	ff2_reg/C
Data Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
<a href="#">FDRE (Prop_fdre C Q)</a>	(f) 0.419	5.127	Site: SLICE_X0Y1	ff2_reg/Q
net (fo=1, routed)	0.162	5.289		ff2
<a href="#">LUT1 (Prop_lut1 I0 O)</a>	(r) 0.299	5.588	Site: SLICE_X0Y1	ODATA_OBUF_inst_i_1/O
net (fo=1, routed)	1.686	7.275		ODATA_OBUF
<a href="#">OBUF (Prop_obuf I O)</a>	(r) 2.672	9.946	Site: R7	ODATA_OBUF_inst/O
net (fo=0)	0.000	9.946		ODATA
			Site: R7	ODATA
<b>Arrival Time</b>		9.946		
Destination Clock Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_10MHz rise edge)	(r) 100.000	100.000		
clock pessimism	0.000	100.000		
clock uncertainty	-0.035	99.965		
output delay	-20.700	79.265		
<b>Required Time</b>		79.265		

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 на ножке ODATA появляются данные. Для анализа по *Hold* запускающий и защелкивающий фронты формируются 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  $0.2 - 0.4 + 0.5 - 0 = 0.3$  нс. Учитывая неопределенность тактового сигнала получаем, что данные должны появиться на ножке ODATA в момент времени -0.265 нс. Отрицательное значение *Required Time* означает, что данные должны появиться на выходе ODATA FPGA раньше, чем генератор сформирует защелкивающий фронт.

Source Clock Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_10...z rise edge)	(r) 0.000	0.000		
	(r) 0.000	0.000	Site: N13	CLK
net (fo=0)	0.000	0.000		CLK
<a href="#">IBUF (Prop_ibuf I O)</a>	(r) 0.230	0.230	Site: N13	CLK_IBUF_inst/O
net (fo=1, routed)	0.634	0.864		CLK_IBUF
<a href="#">BUFG (Prop_bufg I O)</a>	(r) 0.026	0.890	Site: BUF...TRL_X0Y0	CLK_IBUF_BUFG_inst/O
net (fo=2, routed)	0.596	1.486		CLK_IBUF_BUFG
FDRE			Site: SLICE_X0Y1	ff2_reg/C
Data Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
<a href="#">FDRE (Prop_fdre C Q)</a>	(f) 0.128	1.614	Site: SLICE_X0Y1	ff2_reg/Q
net (fo=1, routed)	0.061	1.674		ff2
<a href="#">LUT1 (Prop_lut1 I0 O)</a>	(r) 0.099	1.773	Site: SLICE_X0Y1	ODATA_OBUF_inst_i_1/O
net (fo=1, routed)	0.351	2.125		ODATA_OBUF
<a href="#">OBUF (Prop_obuf I O)</a>	(r) 1.188	3.313	Site: R7	ODATA_OBUF_inst/O
net (fo=0)	0.000	3.313		ODATA
			Site: R7	ODATA
<b>Arrival Time</b>		3.313		
Destination Clock Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_10...z rise edge)	(r) 0.000	0.000		
clock pessimism	0.000	0.000		
clock uncertainty	0.035	0.035		
output delay	-0.300	-0.265		
<b>Required Time</b>		-0.265		

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 отсутствии точного соответствия между временными отчетами и реальными задержками на плате. Например, считает, что фронты тактового сигнала доходят до запускающего триггера в нулевой момент времени.

## 6. Второй способ создания временных ограничений в Vivado.

Рассмотрим второй способ создания временных ограничений для выходных сигналов. В xdc-файле можно отдельно указывать задержки распространения тактового сигнала по дорожкам печатной платы. Для этого используется команда `set_clock_latency` [5]:

```
# 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]
```

С помощью команды `create_clock` создается ограничение на период тактового сигнала, который поступает на ножку CLK FPGA. Далее первая команда `set_clock_latency` с опцией `-early` устанавливает минимальную задержку распространения равной *Tofd\_min*. Конструкция `[get_clocks clk_10MHz]` указывает тактовый сигнал, для которого формируются задержки.

Вторая команда `set_clock_latency` с опцией `-late` задает максимальное время распространения. Опция `-source` означает, что задержка указывается от источника тактового сигнала до ножки FPGA.

Так как задержка распространения до микросхемы 74HC595 отличаются от задержки до FPGA, в файле с временными ограничениями требуется создать еще один тактовый сигнал:

```
# 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 этот сигнал поступает только в микросхему 74HC595, поэтому команда `get_ports`, указывающая ножку FPGA, не используется. Тактовый сигнал, который не попадает в FPGA, но присутствует на плате, называется виртуальным (virtual clock). Для виртуального сигнала `clk_10MHz_Dev` минимальная и максимальная задержки распространения заданы равными *Todd\_min* и *Todd\_max* соответственно.



Так как задержки для тактового сигнала уже установлены, уравнения для *output\_delay\_max* и *output\_delay\_min* упростятся и будут содержать только задержки распространения данных:

$$\begin{aligned} output\_delay\_max &= Tbd\_max + Tdsu; \\ output\_delay\_min &= Tbd\_min - Tdh. \end{aligned} \quad (5)$$

Временные ограничения для выходного сигнала создаются с помощью команды `set_output_delay` в виде:

```
# временные ограничения для выходного сигнала ODATA
set odelay_max [expr $Tbd_max + $Tdsu]
set_output_delay -clock clk_10MHz_Dev -max $odelay_max [get_ports ODATA];

set odelay_min [expr $Tbd_min - $Tdh]
set_output_delay -clock clk_10MHz_Dev -min $odelay_min [get_ports ODATA];
```

Опция `-max` указывает, что задержки предназначены для анализа по *Setup*, а опция `-min` – для анализа по *Hold*. Обратите внимание, что теперь защелкивающий триггер, расположенный внутри микросхемы 74HC595, тактируется виртуальным сигналом, поэтому опция `-clock` задана с сигналом `clk_10MHz_Dev`.

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, 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 ограничения для выходного сигнала ODATA
set odelay_max [expr $Tbd_max + $Tdsu]
set_output_delay -clock clk_10MHz_Dev -max $odelay_max [get_ports ODATA];

set odelay_min [expr $Tbd_min - $Tdh]
set_output_delay -clock clk_10MHz_Dev -min $odelay_min [get_ports ODATA];
```

Рассмотрим, что изменилось во временных отчетах при использовании данного способа создания ограничений. На рисунке 7 представлены общие сведения о входном пути для анализа по *Setup*. В третьей строке указано, что данные для микросхемы 74HC595 появляются на выходе ODATA FPGA и защелкиваются виртуальным сигналом clk\_10MHz\_Dev.

Summary	
Name	↳ Path 3
Slack	<a href="#">69.329ns</a>
Source	ff2_reg/C (rising edge-triggered cell FDRE clocked by clk_10MHz {rise@0.000ns fall@50.000ns period=100.000ns})
Destination	ODATA (output port clocked by clk_10MHz_Dev {rise@0.000ns fall@50.000ns period=100.000ns})
Path Group	clk_10MHz_Dev
Path Type	Max at Slow Process Corner
Requirement	100.000ns (clk_10MHz_Dev rise@100.000ns - clk_10MHz rise@0.000ns)
Data Path Delay	5.238ns (logic 3.390ns (64.714%) route 1.848ns (35.286%))
Logic Levels	2 (LUT1=1 OBUF=1)
Output Delay	20.600ns
Clock Path Skew	<a href="#">-4.808ns</a>
Clock Uncertainty	<a href="#">0.025ns</a>
Clock Dom...Crossing	Inter clock paths are considered valid unless explicitly exclude... timing constraints such as set_clock_groups or set_false_path.

Рисунок 7. Общие сведения о выходном пути (*Setup*).

Расчет задержек для данных и тактового сигнала представлен на рисунке 8. В разделе *Source Clock Path* можно увидеть, что запускающий фронт появляется на выходе генератора в нулевой момент времени и спустя 0.3 нс (clock source latency) достигает тактовой ножки FPGA. Это соответствует задержке *Tofd\_max*.

Далее после прохождения через входной и тактовый буферы фронт доходит до запускающего триггера. Это событие начинает передачу данных, которые, как показано в разделе *Data Path*, появляются на выходе ODATA FPGA в момент времени 10.246 нс.

Source Clock Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_10MHz rise edge)	(r) 0.000	0.000		
clock source latency	0.300	0.300		
	(r) 0.000	0.300	Site: N13	CLK
net (fo=0)	0.000	0.300		CLK
<a href="#">IBUF (Prop_ibuf I O)</a>	(r) 1.001	1.301	Site: N13	CLK_IBUF_inst/O
net (fo=1, routed)	1.972	3.273		CLK_IBUF
<a href="#">BUFG (Prop_bufg I O)</a>	(r) 0.096	3.369	Site: BUF...TRL_X0Y0	CLK_IBUF_BUFG_inst/O
net (fo=2, routed)	1.639	5.008		CLK_IBUF_BUFG
FDRE			Site: SLICE_X0Y1	ff2_reg/C
Data Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
<a href="#">FDRE (Prop_fdre C Q)</a>	(f) 0.419	5.427	Site: SLICE_X0Y1	ff2_reg/Q
net (fo=1, routed)	0.162	5.589		ff2
<a href="#">LUT1 (Prop_lut1 I0 O)</a>	(r) 0.299	5.888	Site: SLICE_X0Y1	ODATA_OBUF_inst_i_1/O
net (fo=1, routed)	1.686	7.575		ODATA_OBUF
<a href="#">OBUF (Prop_obuf I O)</a>	(r) 2.672	10.246	Site: R7	ODATA_OBUF_inst/O
net (fo=0)	0.000	10.246		ODATA
			Site: R7	ODATA
<b>Arrival Time</b>		10.246		
Destination Clock Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_10MHz_Dev rise edge)	(r) 100.000	100.000		
clock source latency	0.200	100.200		
ideal clock network latency	0.000	100.200		
clock pessimism	0.000	100.200		
clock uncertainty	-0.025	100.175		
output delay	-20.600	79.575		
<b>Required Time</b>		79.575		

Рисунок 8. Задержки для данных и тактового сигнала (*Setup*).

Из раздела *Destination Clock Path* рисунка 8 можно увидеть, что защелкивающий фронт тактового сигнала появляется на выходе генератора спустя один период в 100 нс и через *Todd\_min* = 0.2 нс достигает

микросхемы 74HC595. Слагаемое *output delay*, равное *output\_delay\_max* из уравнения (5), указывает, что данным требуется 20.6 нс, чтобы дойти от выхода FPGA до входа микросхемы 74HC595 и оставаться стабильными в течении времени установки. Поэтому с учетом неопределённости тактового сигнала данные должны появиться на ножке ODATA FPGA в момент времени 79.575 нс.

Задержки для анализа по *Hold* показаны на рисунке 8. Можно увидеть, что запускающий фронт формируется на выходе генератора в нулевой момент времени и через *Tofd\_min* = 0.2 нс доходит до тактовой ножки FPGA. Далее в момент времени 3.513 нс на выходе ODATA появятся данные.

Source Clock Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_10MHz rise edge)	(r) 0.000	0.000		
clock source latency	0.200	0.200		
	(r) 0.000	0.200	Site: N13	CLK
net (fo=0)	0.000	0.200		CLK
<a href="#">IBUF (Prop_ibuf I O)</a>	(r) 0.230	0.430	Site: N13	CLK_IBUF_inst/O
net (fo=1, routed)	0.634	1.064		CLK_IBUF
<a href="#">BUFG (Prop_bufg I O)</a>	(r) 0.026	1.090	Site: BUFGCTRL_X0Y0	CLK_IBUF_BUFG_inst/O
net (fo=2, routed)	0.596	1.686		CLK_IBUF_BUFG
FDRE			Site: SLICE_X0Y1	ff2_reg/C
Data Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
<a href="#">FDRE (Prop_fdre C Q)</a>	(f) 0.128	1.814	Site: SLICE_X0Y1	ff2_reg/Q
net (fo=1, routed)	0.061	1.874		ff2
<a href="#">LUT1 (Prop_lut1 I0 O)</a>	(r) 0.099	1.973	Site: SLICE_X0Y1	ODATA_OBUF_inst_i_1/O
net (fo=1, routed)	0.351	2.325		ODATA_OBUF
<a href="#">OBUF (Prop_obuf I O)</a>	(r) 1.188	3.513	Site: R7	ODATA_OBUF_inst/O
net (fo=0)	0.000	3.513		ODATA
			Site: R7	ODATA
<b>Arrival Time</b>		3.513		
Destination Clock Path				
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_10MHz_Dev rise edge)	(r) 0.000	0.000		
clock source latency	0.400	0.400		
ideal clock network latency	0.000	0.400		
clock pessimism	0.000	0.400		
clock uncertainty	0.025	0.425		
output delay	-0.500	-0.075		
<b>Required Time</b>		-0.075		

Рисунок 8. Задержки для данных и тактового сигнала (*Hold*).

Защелкивающий фронт появится на выходе генератора также в нулевой момент времени и дойдет до ножки SRCLK микросхемы 75HC595 через  $T_{odd\_max} = 0.4$  нс. Учитывая задержку распространения и неопределенность тактового сигнала, требуемое время появления данных на выходе ODATA FPGA равно -0.075 нс.

Использование виртуальных тактовых сигналов для создания ограничений несколько упрощает интерпретацию временных отчетов. Более того, данный способ рекомендуется Vivado. Если открыть вкладку XDC в Language Templates, то можно найти пример временных ограничений для выходного сигнала:

```
# Rising Edge System Synchronous Outputs
#
# A System Synchronous design interface is a clocking technique
# in which the same active-edge of a system clock is used for
# both the source and destination device.
#
# dest
# clk      _____|_____
#
#          (trce_dly_max+tsu) <-----|
#          (trce_dly_min-thd) <-|
#
# data     XXXXXXXXXXXXXXXX__DATA__XXXXXXXXXXXXX
#
set destination_clock <clock_name>; # Name of destination_clock clock
set tsu              0.000;         # Destination device setup time
set thd              0.000;         # Destination device hold time
set trce_dly_max     0.000;         # Maximum board trace delay
set trce_dly_min     0.000;         # Minimum board trace delay
set output_ports     <output_ports >; # List of output ports

# Output Delay Constraint
set_output_delay -clock $destination_clock -max
                 [expr $trce_dly_max + $tsu] [get_ports $output_ports];

set_output_delay -clock $destination_clock -min
                 [expr $trce_dly_min - $thd] [get_ports $output_ports];
```

В данном случае значение *output\_delay* для анализа по *Setup* задается в виде суммы задержек *tsu* (*Tdsu*) и *trce\_dly\_max* (*Tbd\_max*), а для анализа по *Hold* – как разность *trce\_dly\_min* (*Tbd\_min*) и *thd* (*Tdh*). Это полностью соответствует уравнениям (5). Заметим также, что если пренебречь задержками тактового сигнала *Todd* и *Tofd*, то оба рассмотренных выше способа создания ограничений будут полностью эквивалентны.

В заключении отметим следующий интересный момент. Можно увидеть, что значения *Slack*, представленные на рисунках 4 и 7, отличаются и равны 69.319 нс и 69.329 нс соответственно. Разница в 0.01 нс возникает из-за разной величины неопределенности тактового сигнала. Более подробно о причине такого несоответствия можно прочитать в [2].

## **Заключение.**

В данной статье был рассмотрен временной анализ для выходных сигналов FPGA. Показан вывод уравнений статического временного анализа. Представлено два способа создания временных ограничений, а также рассказано о виртуальных тактовых сигналах.

## **Ссылки.**

1. Основы статического временного анализа. Часть 2.1: System Synchronous Input Delay Constraint
2. [Основы статического временного анализа. Часть 1: Period Constraint](#)
3. [Datasheet 74HC595](#)
4. [How to Calculate Trace Length from Time Delay Value for High-speed Signals](#)
5. [Using Constraints \(UG 903\)](#)