**Основы статического временного анализа.**

**Часть 6: Multicycle Path Constraint.**

**Введение.**

В данной статье представлен временной анализ передачи сигналов между двумя тактовыми доменами. Показано несколько способов исключения путей из временного анализа. Рассмотрены инструменты Vivado, предназначенные для проверки корректности передачи данных между тактовыми доменами.

1. **Пересечение синхронных тактовых доменов.**

Доменами

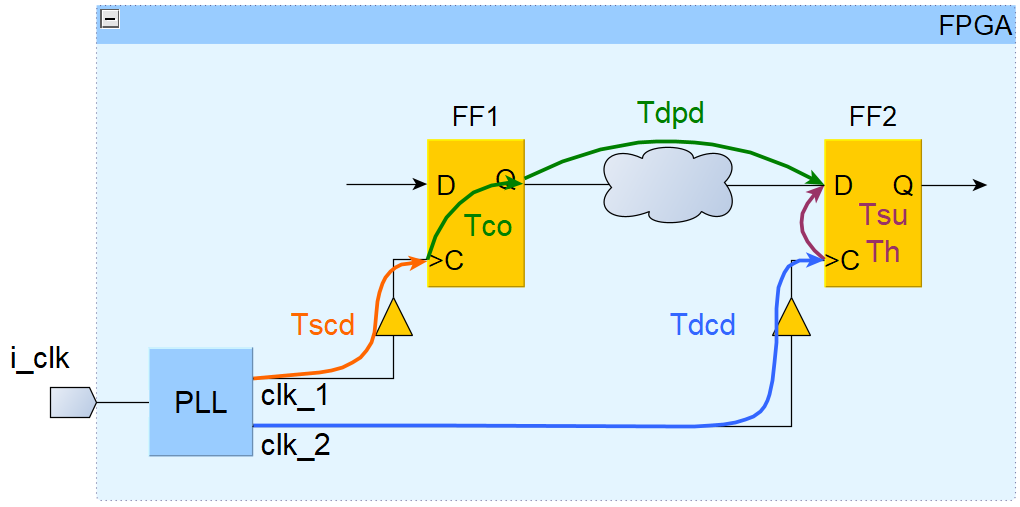


Рисунок 1. Путь с задержками для данных и тактовых сигналов.

Ниже даны определения задержек, представленных на рисунке 1.

* *Tscd* (***S****ource* ***C****lock* ***D****elay*) – задержка запускающего тактового сигнала от ножки CLK\_1 FPGA до тактового входа триггера FF1;
* *Tdcd* (***D****estination* ***C****lock* ***D****elay*) – задержка защелкивающего тактового сигнала от ножки CLK\_2 FPGA до тактового входа триггера FF2;
* *Tco* (***C****lock to* ***O****utput*) – интервал времени между приходом фронта на тактовый вход триггера и появлением данных на его выходе Q;
* *Tdpd* (***D****ata* ***P****ropagation* ***D****elay*) – задержка распространения данных по соединениям и через комбинационную логику;
* *Tsu* (***S****et****U****p time*) – время установки защелкивающего триггера;
* *Th* (***H****old time*) – время удержания защелкивающего триггера.

Подставим в него найденные ранее результаты и получим:

где *ΔTsu* – интервал времени между появлением запускающего фронта на ножке CLK\_1 и защелкивающего фронта на ножке CLK\_2:

Учитывая предыдущие результаты, можем записать выражение для *Slack* в следующем виде:

где *ΔTh* – интервал времени между появлением запускающего и защелкивающего фронтов:

каждой такой пары должны быть проверены два условия [2]:

1. данные, которые передаются по запускающему фронту CLK\_1 не должны быть приняты фронтом CLK\_2, который является предыдущим по отношению к текущим защелкивающему фронту CLK\_2;
2. данные, которые передаются фронтом CLK\_1, следующим после запускающего, не должны быть приняты текущим защелкивающим фронтом CLK\_2.

module top (

    input  logic i\_clk,

    input  logic i\_data,

    output logic o\_data

);

    logic clk\_1, clk\_2;

    logic data\_clk\_1, data\_clk\_2;

    // PLL для формирования синхронных сигналов

    clk\_wiz\_0 PLL (

        .i\_clk(i\_clk),

        .clk\_1(clk\_1),

        .clk\_2(clk\_2)

    );

    // передающий домен

    always\_ff @(posedge clk\_1) begin

        data\_clk\_1 <= i\_data;

    end

    // приемный домен

    always\_ff @(posedge clk\_2) begin

        data\_clk\_2 <= data\_clk\_1;

        o\_data <= data\_clk\_2;

    end

endmodule

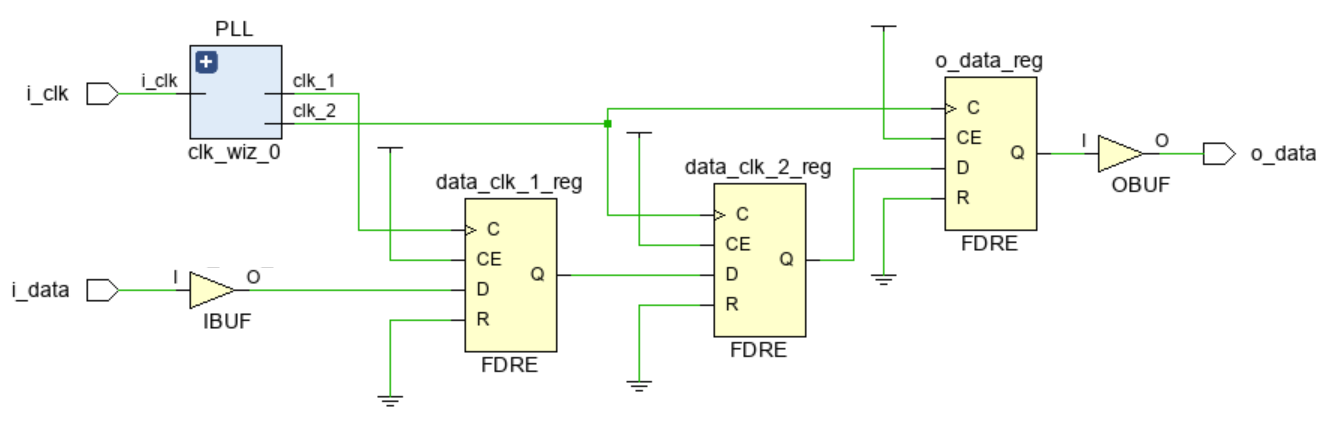


Рисунок 2. Схема проекта.

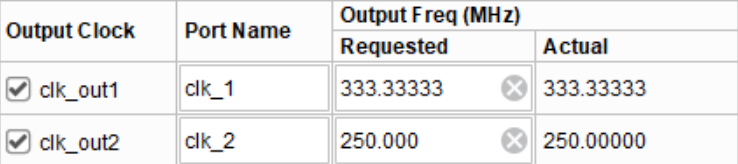


Рисунок 3. Настройки PLL.

# период входного тактового сигнала

create\_clock -period 10.0 -name i\_clk [get\_ports i\_clk]

# объявление переменных, которые указывают места формирования тактовых сигналов

set pll\_input    [get\_pins PLL/inst/mmcm\_adv\_inst/CLKIN1]

set pll\_output\_1 [get\_pins PLL/inst/mmcm\_adv\_inst/CLKOUT0]

set pll\_output\_2 [get\_pins PLL/inst/mmcm\_adv\_inst/CLKOUT1]

# обновление имен сгенерированных с помощью PLL тактовых сигналов

create\_generated\_clock -name clk\_1 -source $pll\_input -master\_clock [get\_clocks i\_clk] $pll\_output\_1

create\_generated\_clock -name clk\_2 -source $pll\_input -master\_clock [get\_clocks i\_clk] $pll\_output\_2

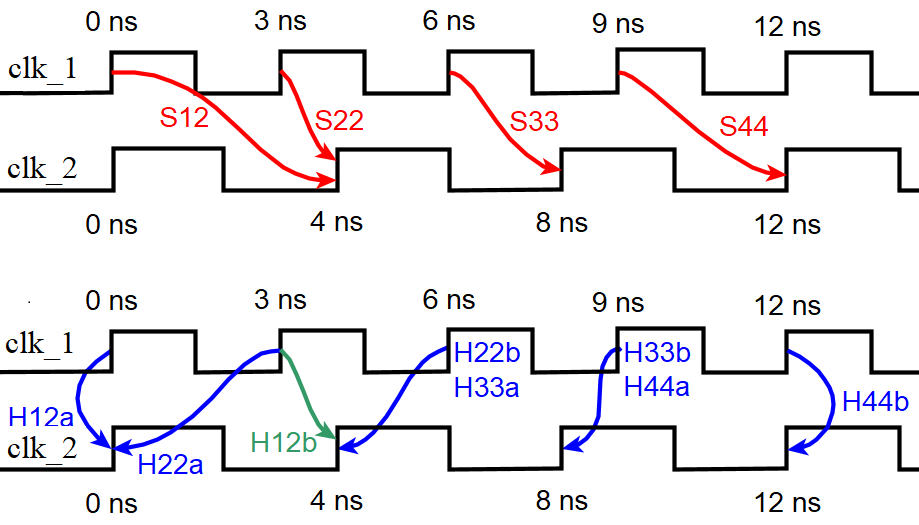


Рисунок 4. Диаграммы сигналов для временного анализа.

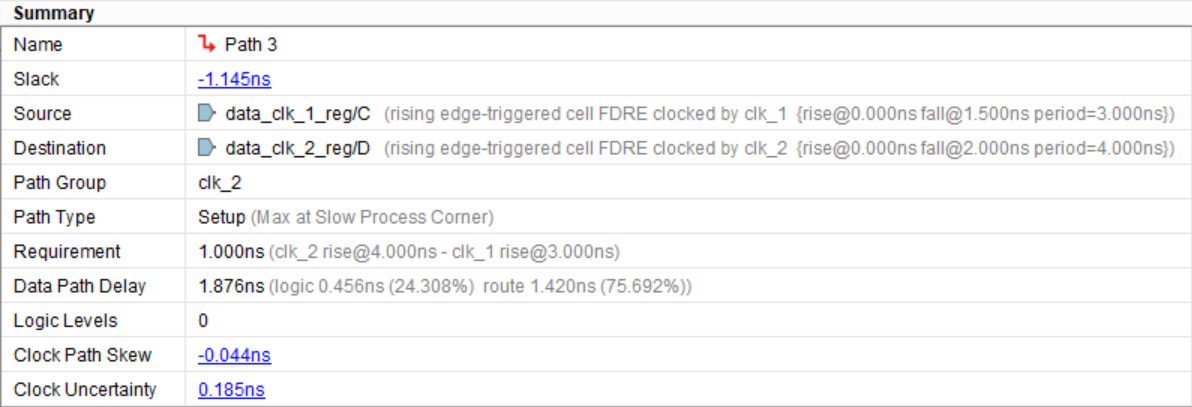


Рисунок 5. Раздел *Summary* временного отчета для анализа по Setup.

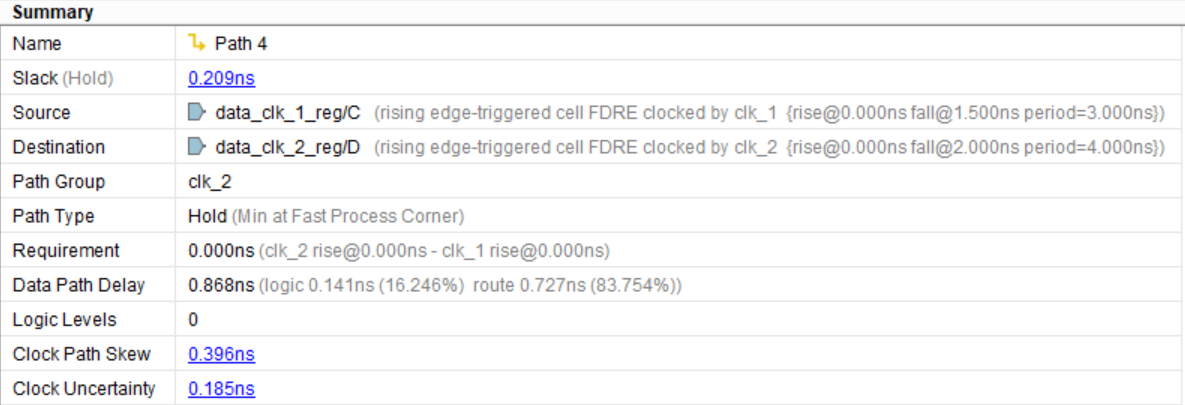


Рисунок 6. Раздел *Summary* временного отчета для анализа по Hold.

1. **Изменение положения защелкивающего фронта.**

доменами

# сдвиг защелкивающего фронта вправо для анализа по Setup

set\_multicycle\_path 2 -setup -end -from [get\_pins data\_clk\_1\_reg/C] -to [get\_pins data\_clk\_2\_reg/D]

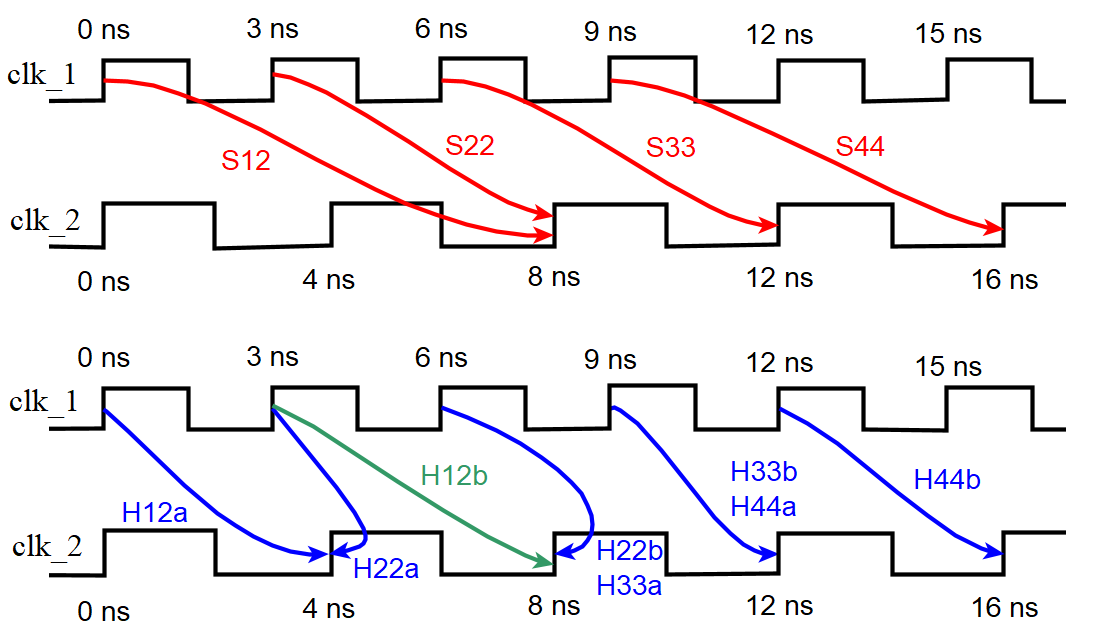


Рисунок 7. Диаграммы сигналов для временного анализа.

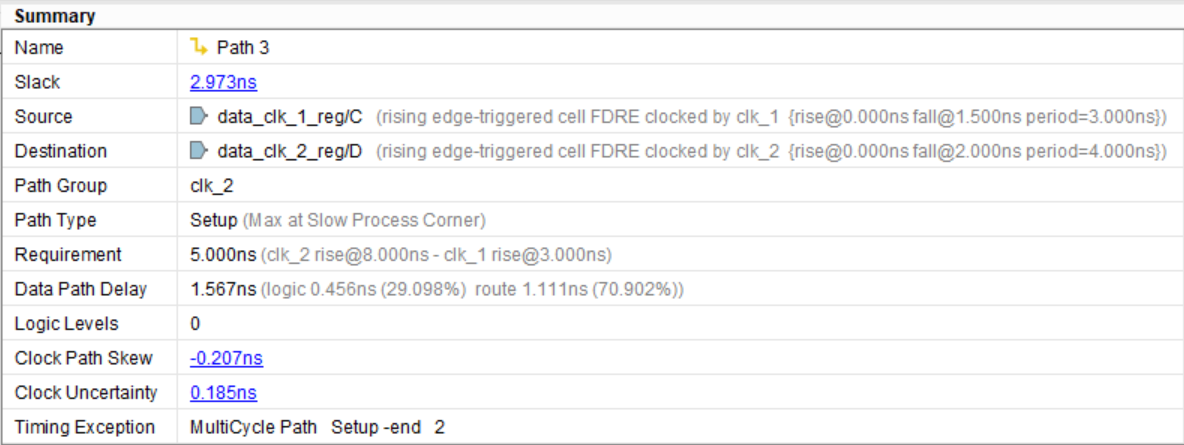


Рисунок 8. Раздел *Summary* временного отчета для анализа по Setup.

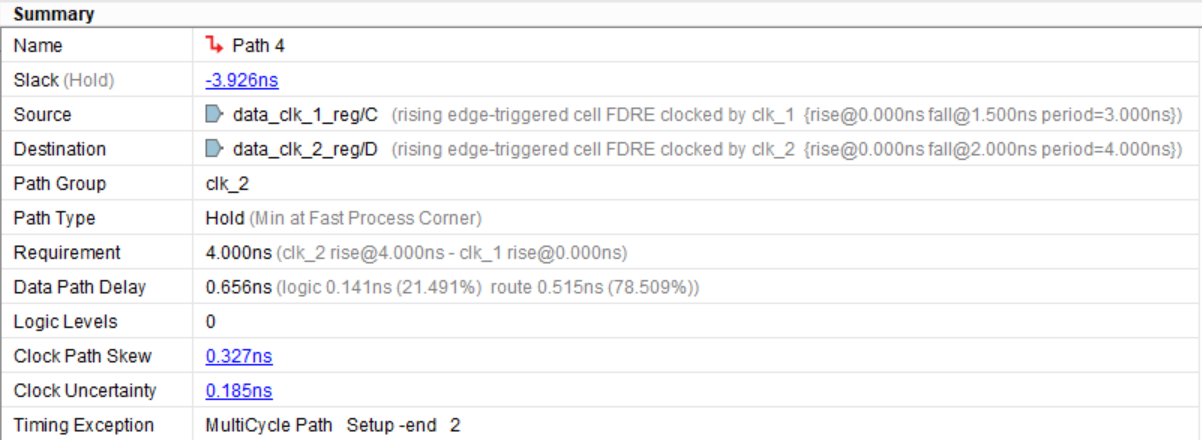


Рисунок 9. Раздел *Summary* временного отчета для анализа по Hold.

# сдвиг защелкивающего фронта влево для анализа по Hold

set\_multicycle\_path 1 -hold  -end -from [get\_pins data\_clk\_1\_reg/C] -to [get\_pins data\_clk\_2\_reg/D]

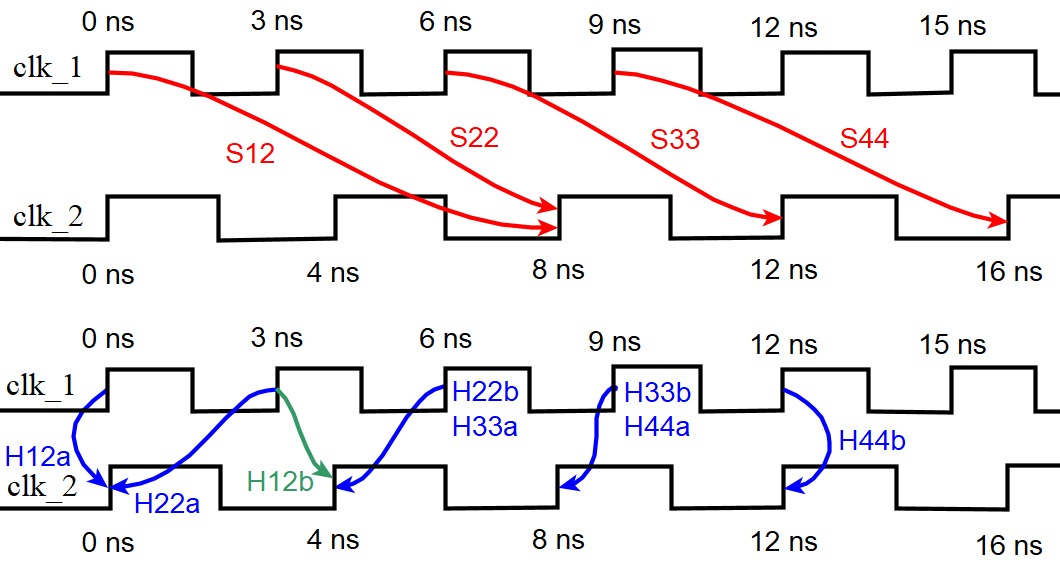


Рисунок 10. Диаграммы сигналов для временного анализа.

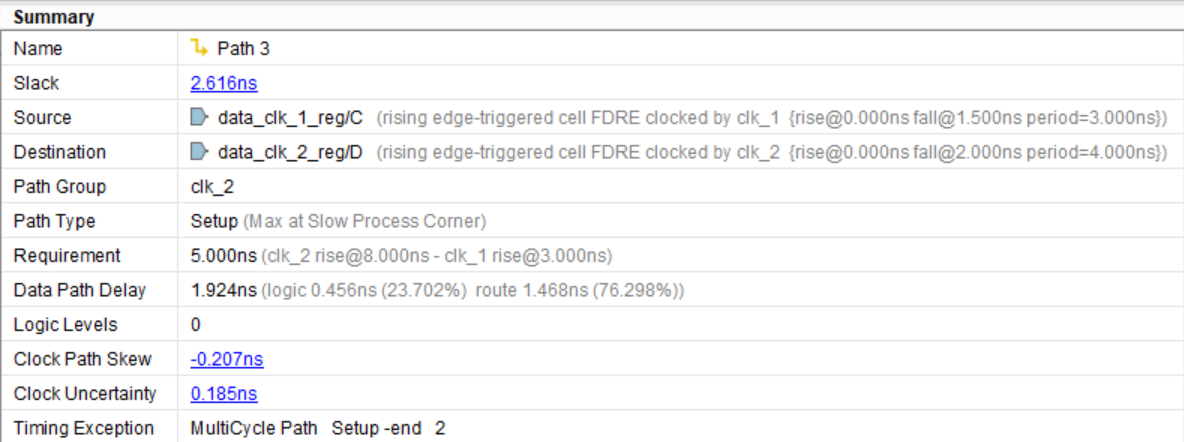


Рисунок 11. Раздел *Summary* временного отчета для анализа по Setup.

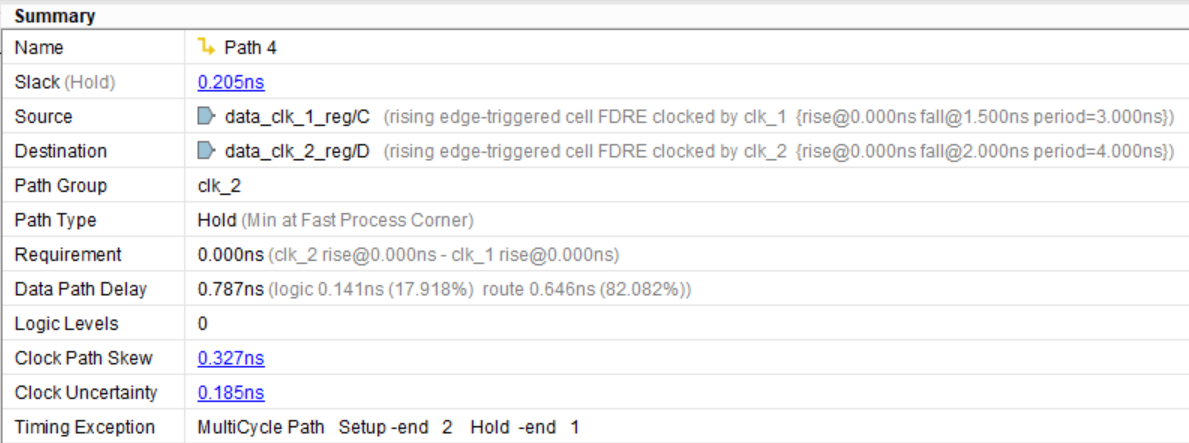


Рисунок 12. Раздел *Summary* временного отчета для анализа по Hold.

1. **Изменение положения запускающего фронта.**

Доменами

# сдвиг запускающего фронта влево для анализа по Setup

set\_multicycle\_path 2 -setup -start -from [get\_pins data\_clk\_1\_reg/C] -to [get\_pins data\_clk\_2\_reg/D]

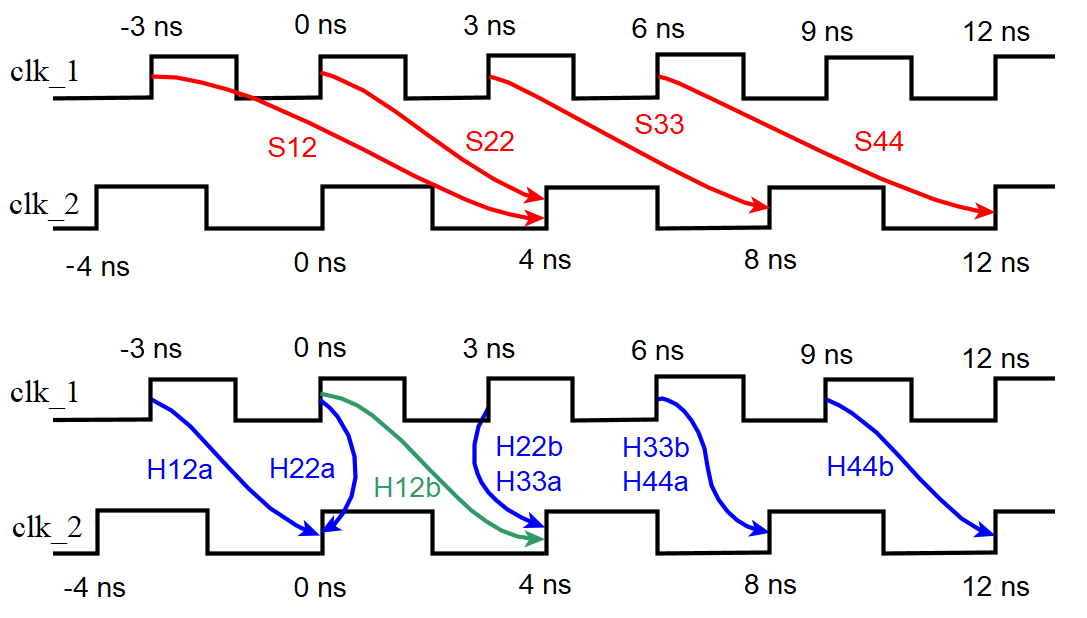


Рисунок 13. Диаграммы сигналов для временного анализа.

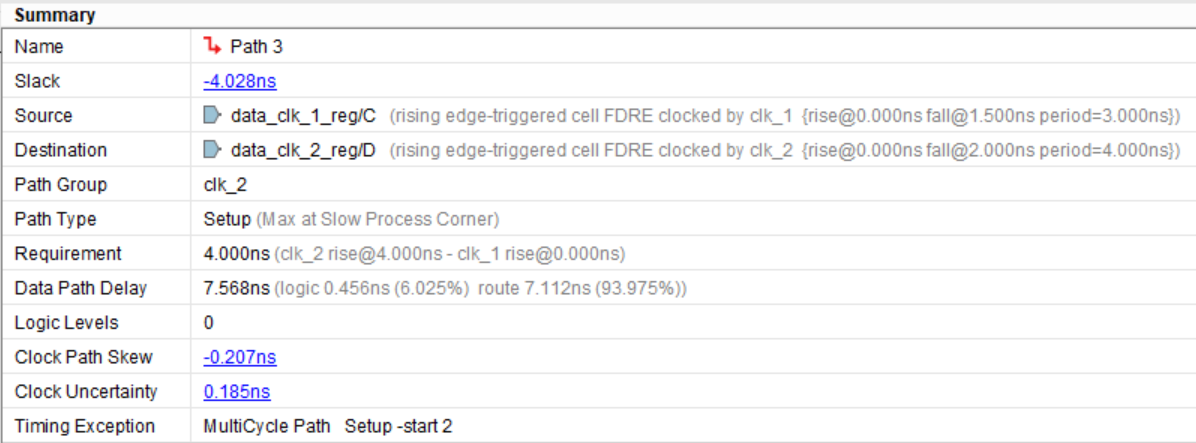


Рисунок 14. Раздел *Summary* временного отчета для анализа по Setup.

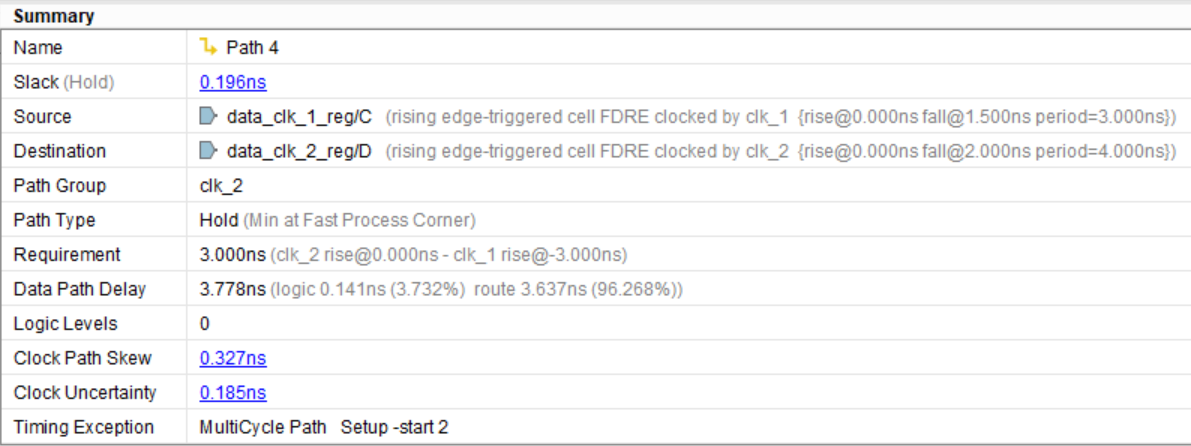


Рисунок 15. Раздел *Summary* временного отчета для анализа по Hold.

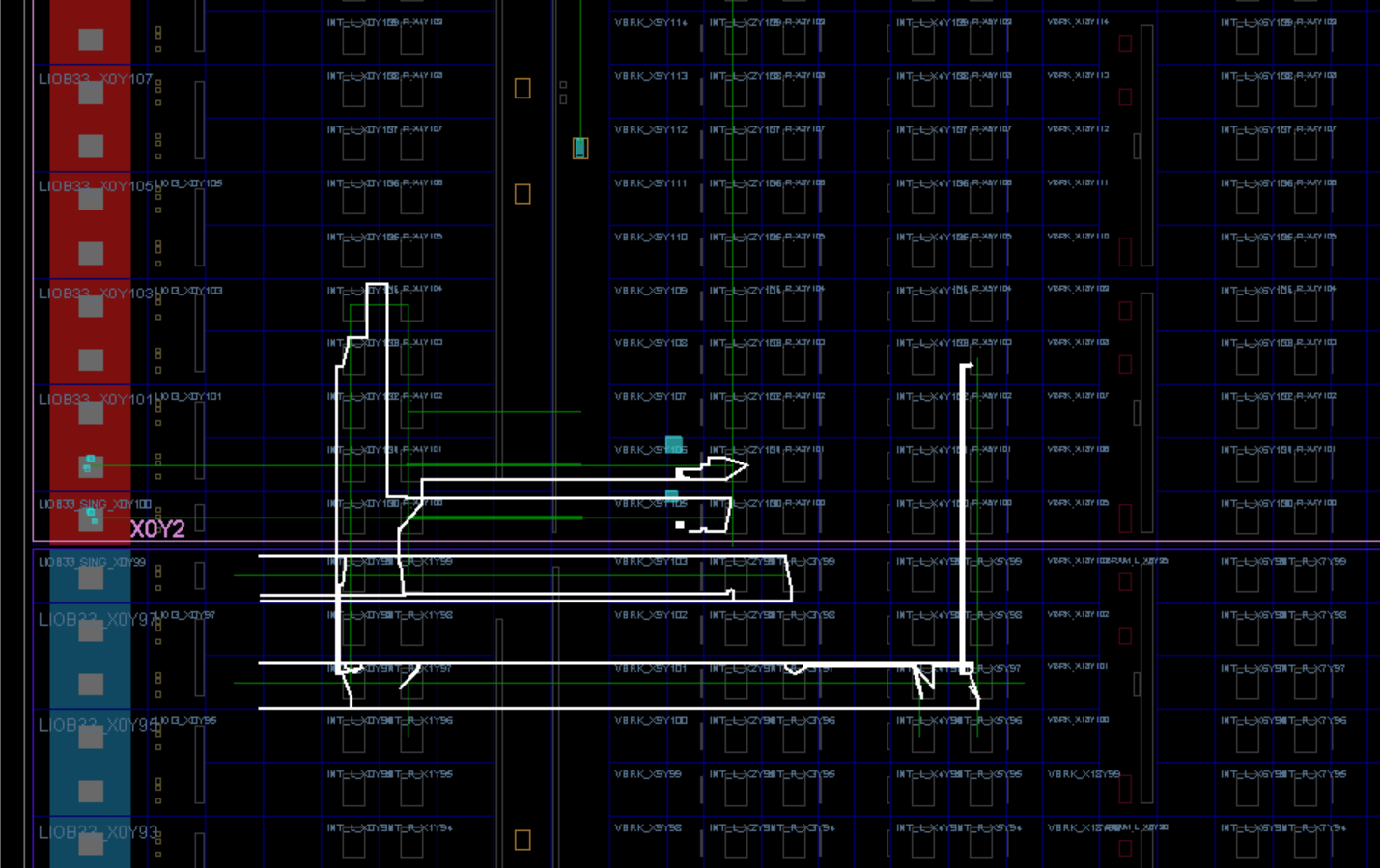


Рисунок 16. Результат размещения и трассировки проекта.

# сдвиг запускающего фронта вправо для анализа по Hold

set\_multicycle\_path 1 -hold  -start -from [get\_pins data\_clk\_1\_reg/C] -to [get\_pins data\_clk\_2\_reg/D]

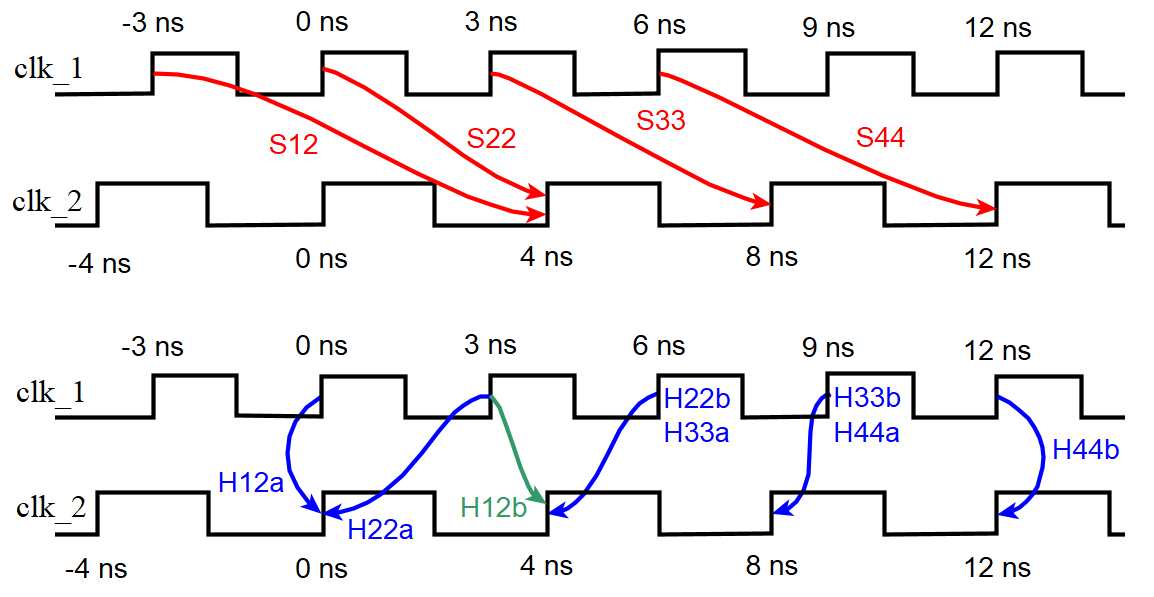


Рисунок 17. Диаграммы сигналов для временного анализа.

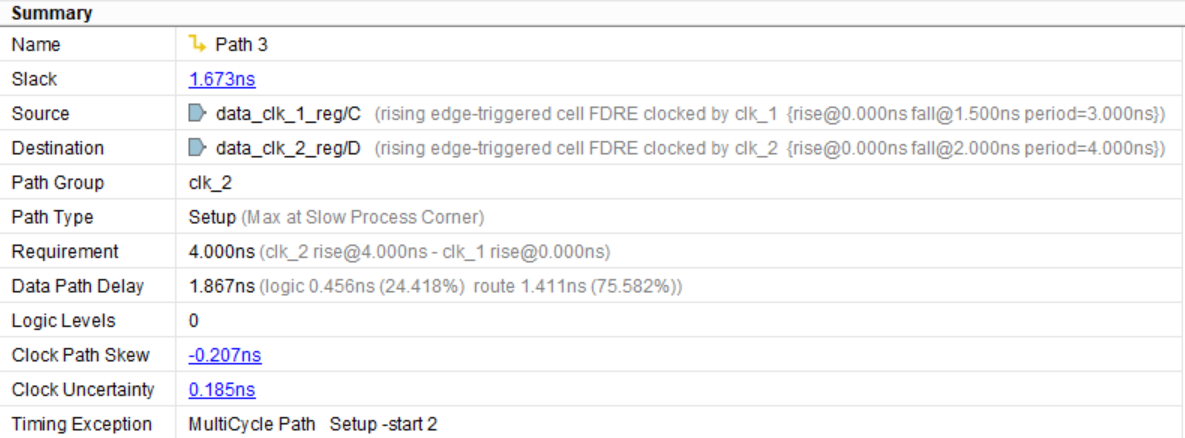


Рисунок 18. Раздел *Summary* временного отчета для анализа по Setup.

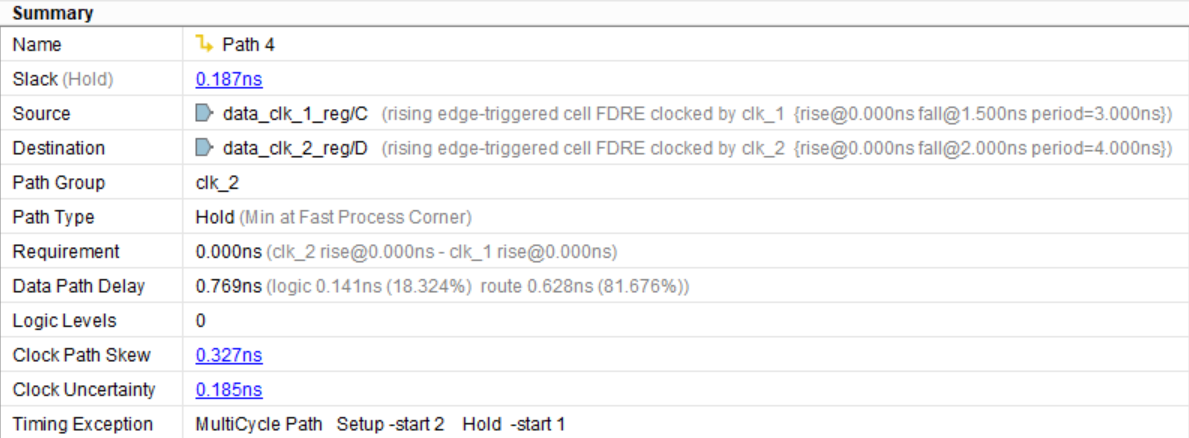


Рисунок 19. Раздел *Summary* временного отчета для анализа по Hold.

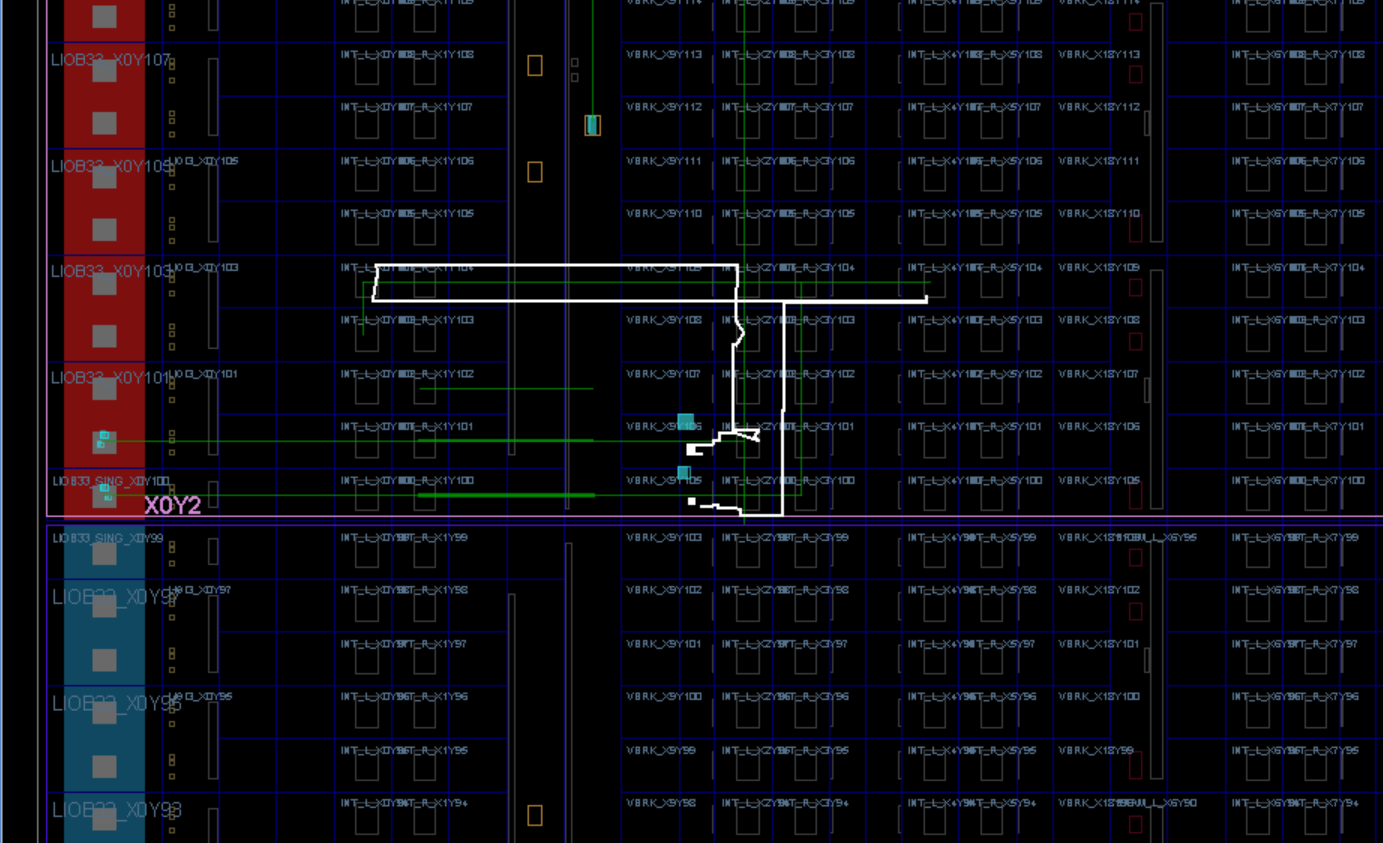


Рисунок 20. Результат размещения и трассировки проекта.

1. **Сдвиг фазы тактового сигнала.**

Доменами

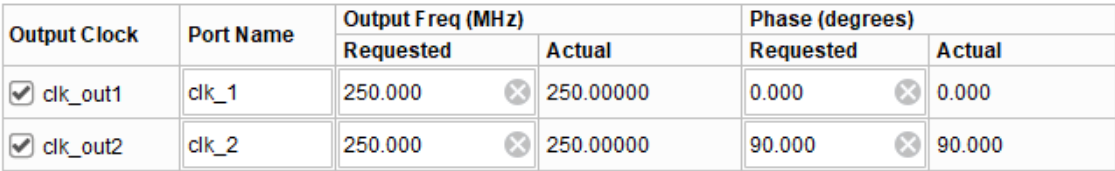


Рисунок 21. Настройки PLL.

# период входного тактового сигнала

create\_clock -period 10.0 -name i\_clk [get\_ports i\_clk]

# объявление переменных, которые указывают места формирования тактовых сигналов

set pll\_input    [get\_pins PLL/inst/mmcm\_adv\_inst/CLKIN1]

set pll\_output\_1 [get\_pins PLL/inst/mmcm\_adv\_inst/CLKOUT0]

set pll\_output\_2 [get\_pins PLL/inst/mmcm\_adv\_inst/CLKOUT1]

# обновление имен сгенерированных с помощью PLL тактовых сигналов

create\_generated\_clock -name clk\_1 -source $pll\_input -master\_clock [get\_clocks i\_clk] $pll\_output\_1

create\_generated\_clock -name clk\_2 -source $pll\_input -master\_clock [get\_clocks i\_clk] $pll\_output\_2

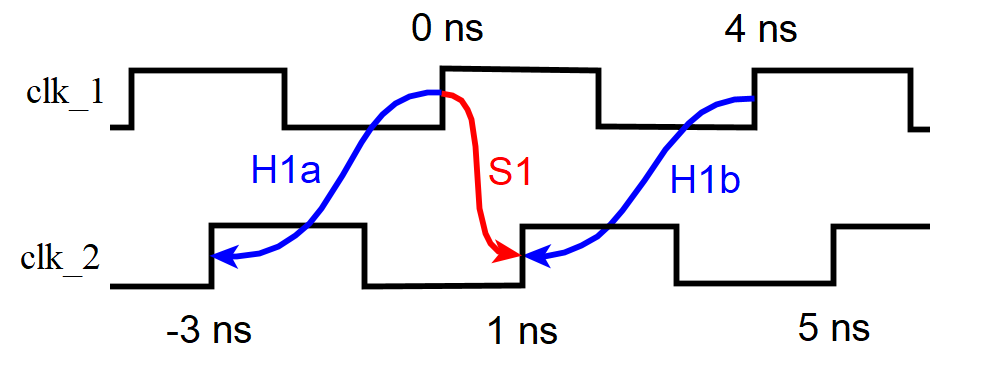


Рисунок 22. Диаграммы сигналов для временного анализа.

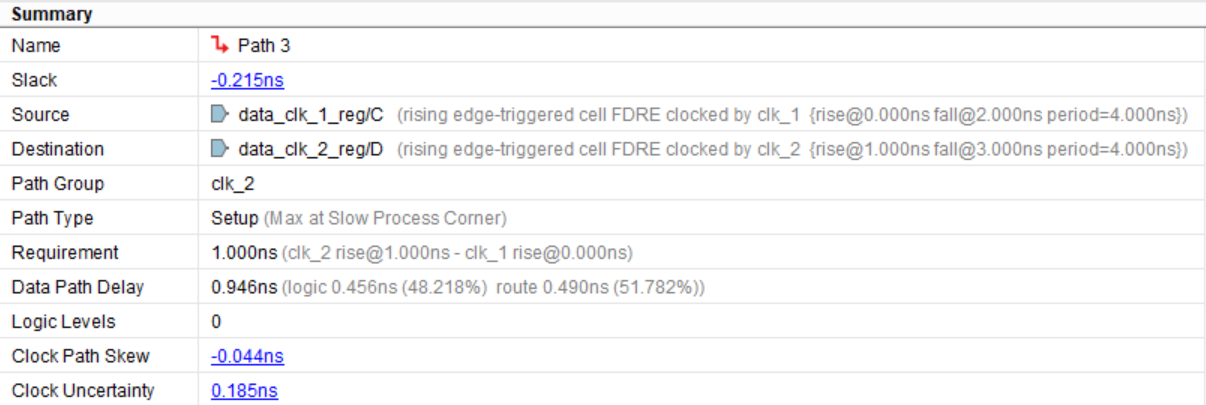


Рисунок 23. Раздел *Summary* временного отчета для анализа по Setup.

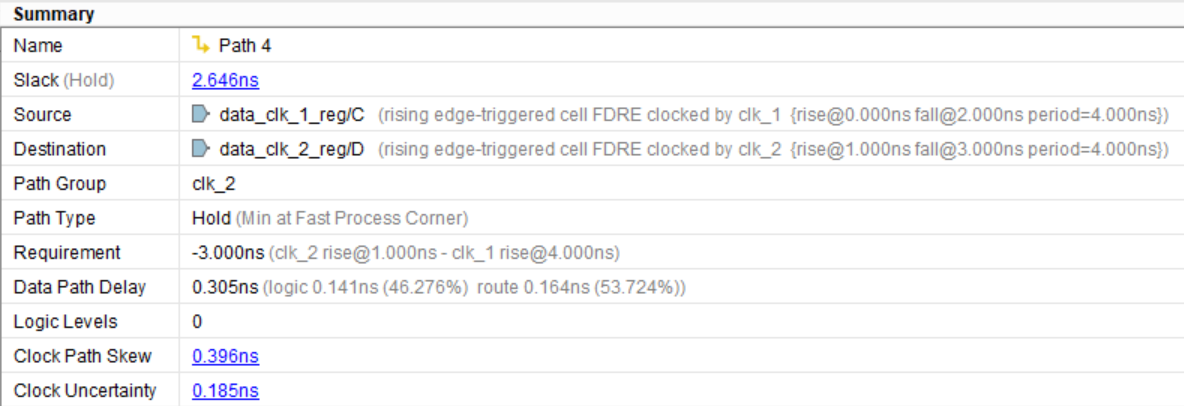


Рисунок 24. Раздел *Summary* временного отчета для анализа по Hold.

# сдвиг защелкивающего фронта вправо для анализа по Setup

set\_multicycle\_path 2 -setup -end -from [get\_pins data\_clk\_1\_reg/C] -to [get\_pins data\_clk\_2\_reg/D]

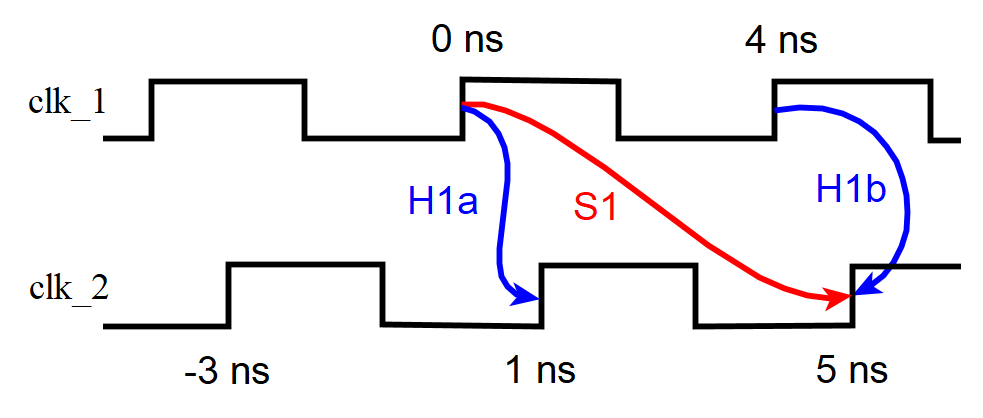


Рисунок 25. Диаграммы сигналов для временного анализа.

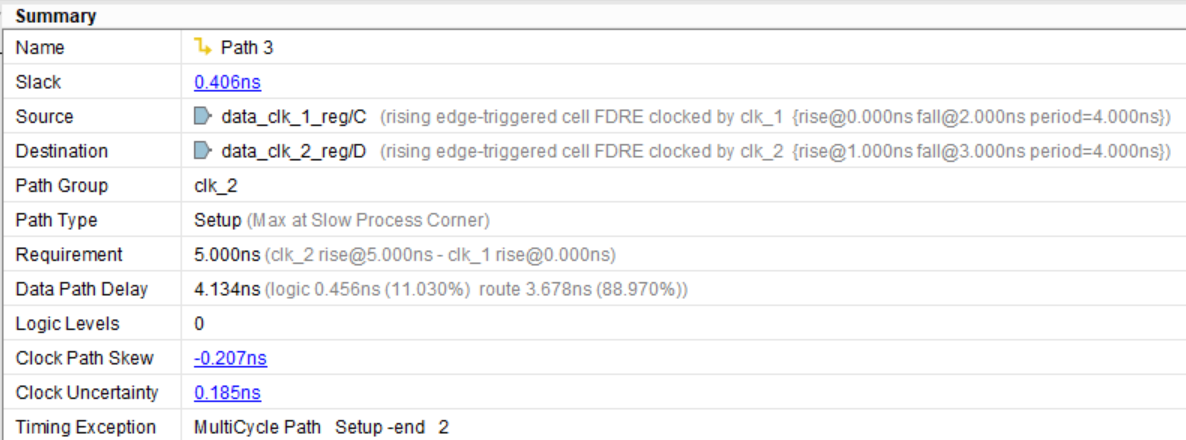


Рисунок 26. Раздел *Summary* временного отчета для анализа по Setup.

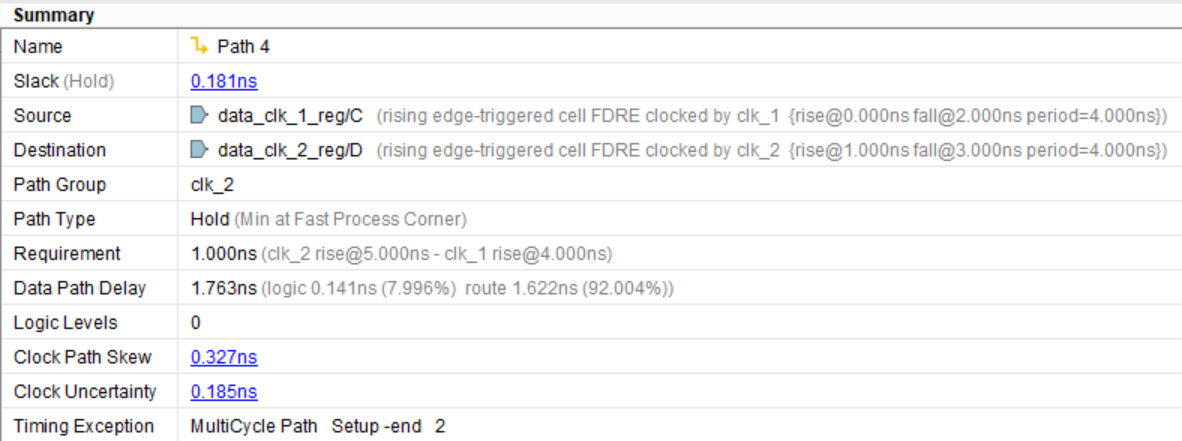


Рисунок 27. Раздел *Summary* временного отчета для анализа по Hold.

1. **Работа модуля по строб-сигналу**

доменами

module top\_2 (

    input  logic        clk,

    input  logic        reset,

    output logic [31:0] q

);

    // строб-сигнал для счетчика

    logic counter\_ce;

    // формирование строб сигнала

    always\_ff @(posedge clk)

        if (reset)

            counter\_ce <= 1'b0;

        else

            counter\_ce <= ~counter\_ce;

    // счетчик

    always\_ff @(posedge clk)

        if (reset)

            q <= '0;

        else if (counter\_ce)

            q <= q + 1'b1;

endmodule

# период входного тактового сигнала

create\_clock -period 2.5 -name clk [get\_ports clk]

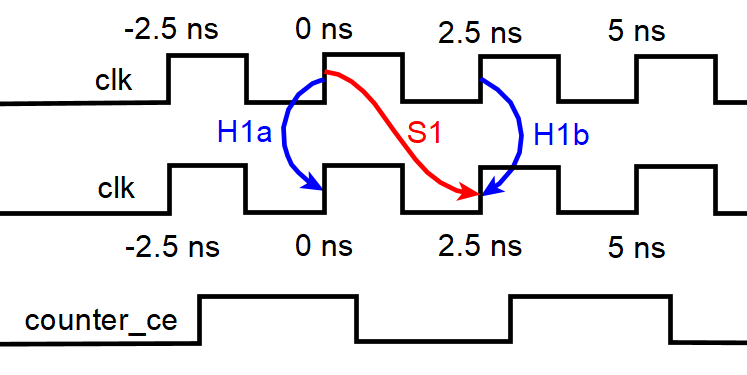


Рисунок 28. Диаграммы сигналов для временного анализа.

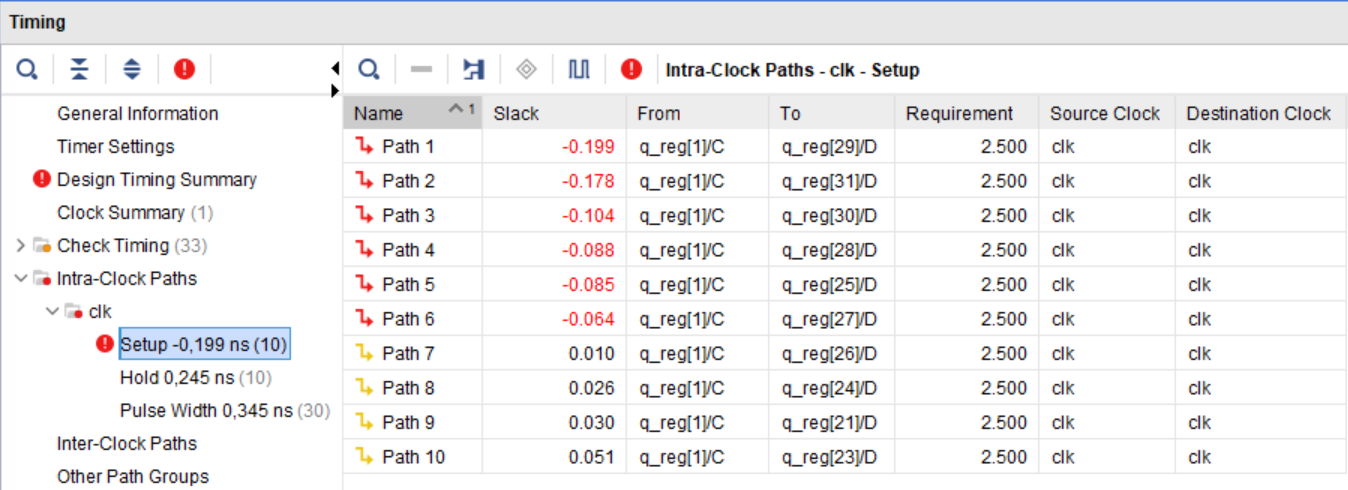


Рисунок 29. Результаты временного анализа.

# получение триггеров счетчика по имени ячеек netlist

set counter\_flops [get\_cells q\_reg[\*]]

# добавление multicycle path

set\_multicycle\_path 2 -setup -end -from $counter\_flops -to $counter\_flops

set\_multicycle\_path 1 -hold  -end -from $counter\_flops -to $counter\_flops

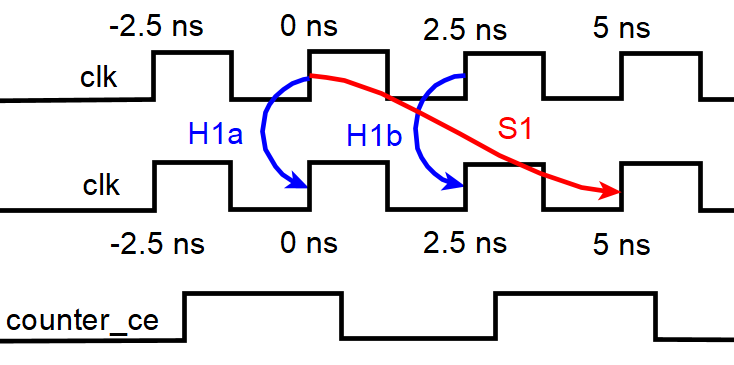


Рисунок 30. Диаграммы сигналов для временного анализа.

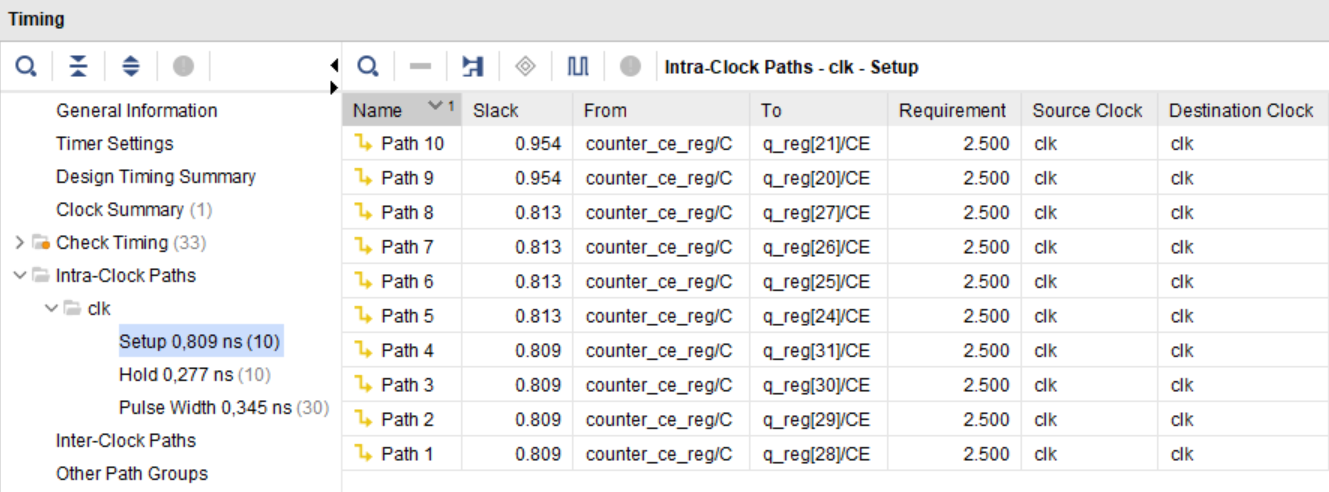


Рисунок 31. Результаты временного анализа.

# получение триггеров счетчика по имени переменной и типу

set counter\_flops [get\_cells q\* -filter {PRIMITIVE\_TYPE =~ FLOP\_LATCH.flop.\*}]

# получение триггеров счетчика по сигналу counter\_ce

set ce\_net [get\_nets counter\_ce]

set flops\_ce\_pins [get\_pins -of\_objects $ce\_net -filter {NAME =~ "\*CE"}]

set counter\_flops [get\_cells -of\_objects $flops\_ce\_pins]

1. **Ограничения для входных и выходных сигналов.**

Доменами

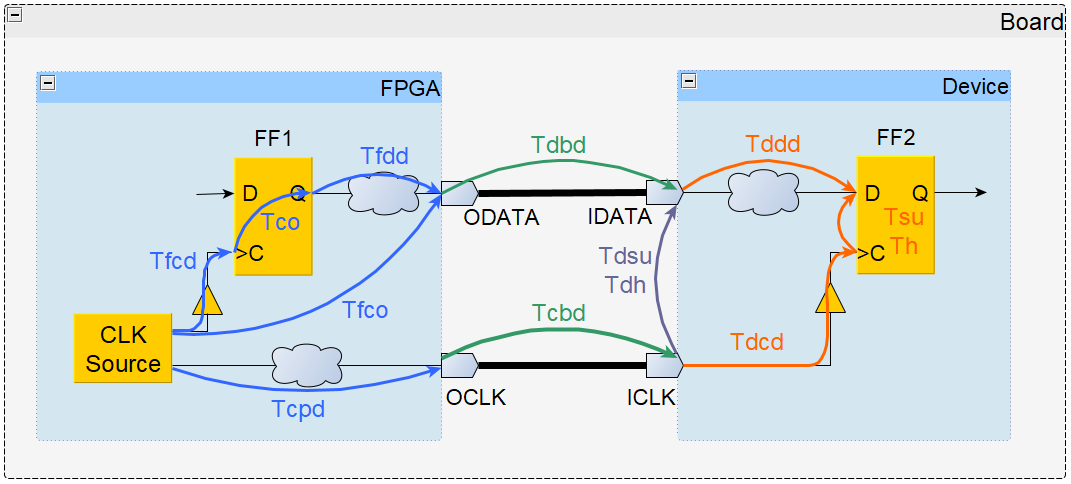


Рисунок 32. Путь с задержками для данных и тактовых сигналов.

module top\_3 (

    input  logic i\_clk,

    input  logic i\_data,

    output logic o\_clk,

    output logic o\_data

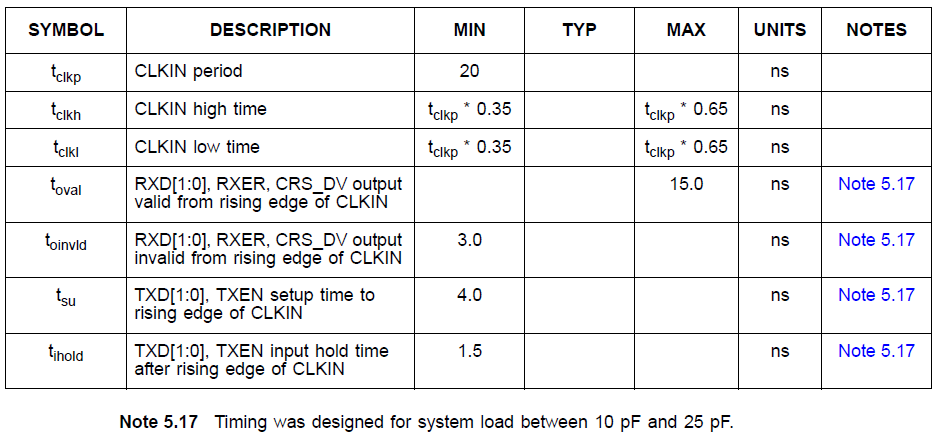
);

    always\_ff @(posedge i\_clk)

        o\_data <= i\_data;

    assign o\_clk = i\_clk;

endmodule



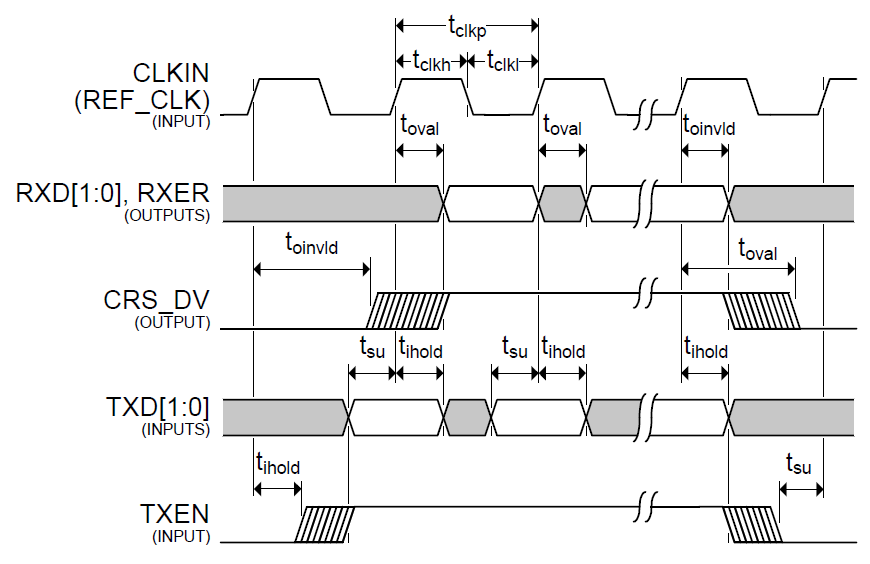


Рисунок 33. Задержки и временные диаграммы для LAN8740A.

# период тактового сигнала

set Tclk 20

# время установки и удержания для микросхемы LAN8740A

set Tdsu 4.0

set Tdh 1.5

# минимальное и максимальное время распространения данных по дорожкам платы

set Tdbd\_max 0.15

set Tdbd\_min 0.1

# минимальное и максимальное время распространения тактового сигнала по дорожкам платы

set Tcbd\_max 0.12

set Tcbd\_min 0.07

# ограничение на период входного тактового сигнала

create\_clock -period $Tclk -name i\_clk [get\_ports i\_clk]

# ограничение на период выходного тактового сигнала

create\_generated\_clock -name o\_clk -source [get\_ports i\_clk] -divide\_by 1 [get\_ports o\_clk]

# временные ограничение для выходного сигнала o\_data

set\_output\_delay -clock o\_clk

-max [expr $Tclk + $Tdbd\_max + $Tdsu - $Tcbd\_min] [get\_ports o\_data]

set\_output\_delay -clock o\_clk

-min [expr $Tclk + $Tdbd\_min - $Tdh  - $Tcbd\_max] [get\_ports o\_data]

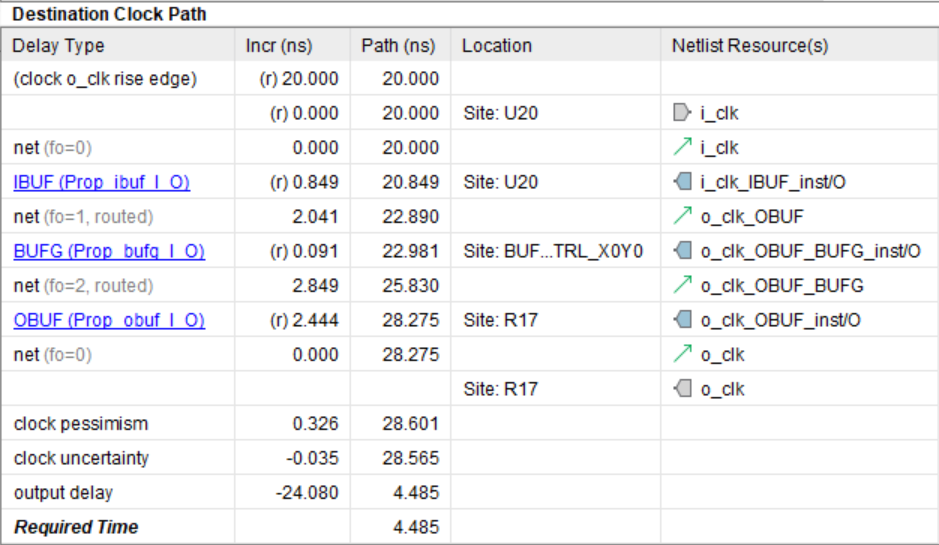


Рисунок 34. Задержки для защелкивающего фронта при анализе по Setup.

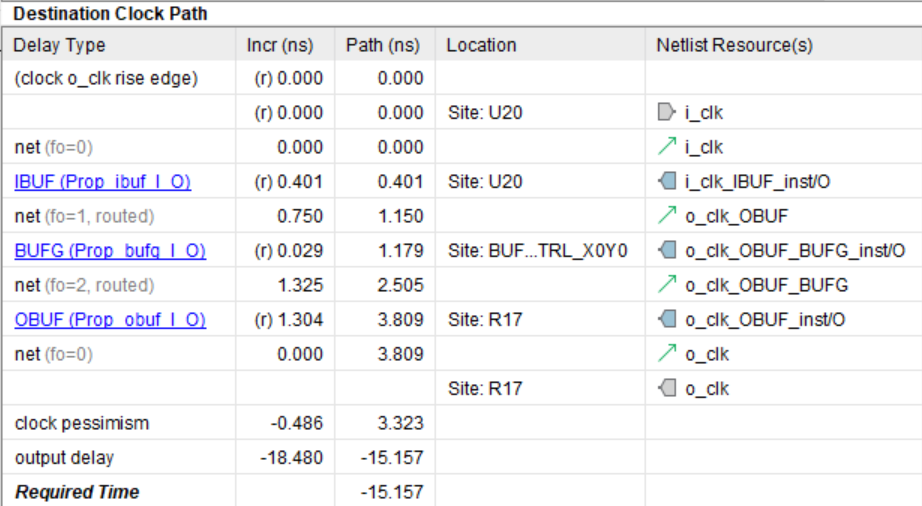


Рисунок 35. Задержки для защелкивающего фронта при анализе по Hold.

# временные ограничение для выходного сигнала o\_data

set\_output\_delay -clock o\_clk

-max [expr $Tdbd\_max + $Tdsu - $Tcbd\_min] [get\_ports o\_data]

set\_output\_delay -clock o\_clk

-min [expr $Tdbd\_min - $Tdh  - $Tcbd\_max] [get\_ports o\_data]

# добавление multicycle path

set\_multicycle\_path 0 -setup -end -to [get\_ports o\_data]

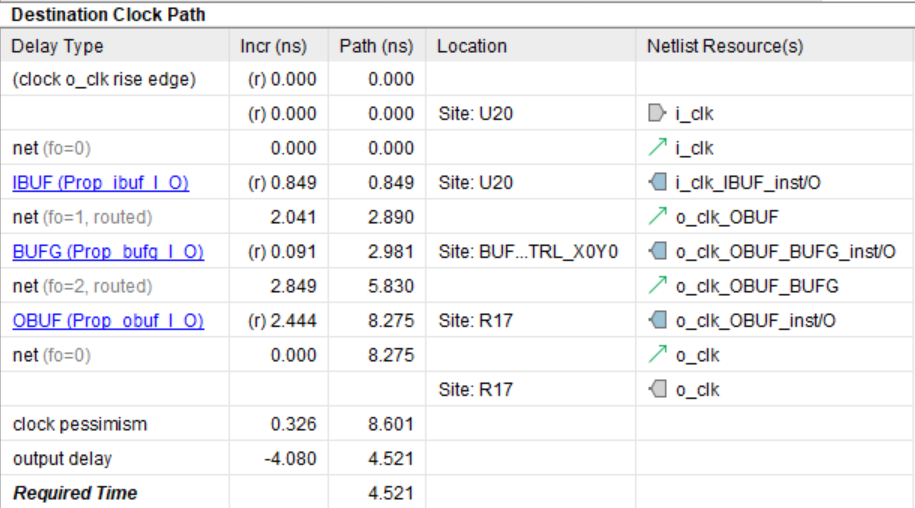


Рисунок 36. Задержки для защелкивающего фронта при анализе по Setup.

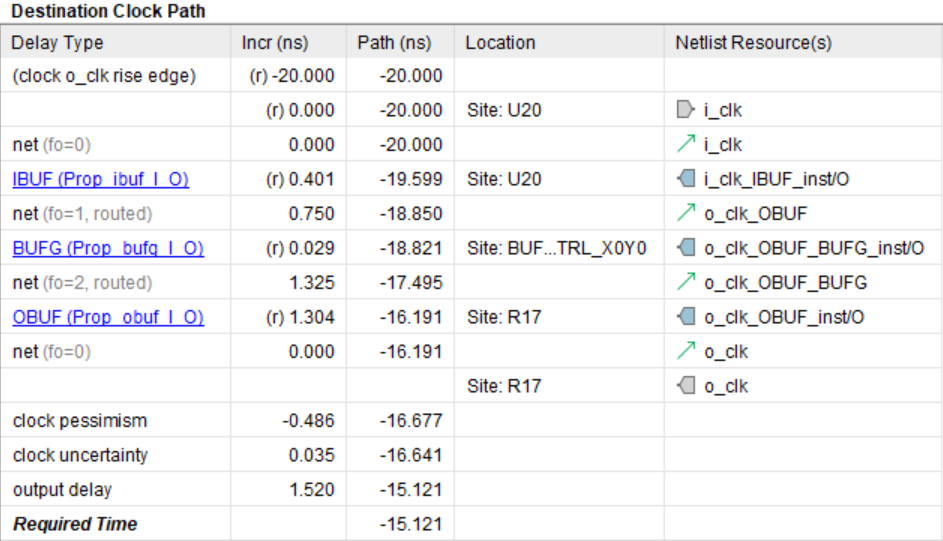


Рисунок 37. Задержки для защелкивающего фронта при анализе по Hold.

**Заключение.**

В статье представлен временной анализ при передаче данных между тактовыми доменами. Рассказано о возможных проблемах, возникающих   
из-за нарушения временных ограничений для асинхронных тактовых сигналов. Рассмотрены несколько способов исключения пути из временного анализа. Кратко описан принцип работы синхронизатора на основе сдвоенных триггеров.

**Ссылки.**

1. [Основы статического временного анализа. Часть 1: Period Constraint](https://fpga-systems.ru/static-timing-analysis-part-1-period-constraints)
2. [Vivado Design Analysis and Closure Techniques (UG 906)](https://docs.xilinx.com/v/u/2017.3-English/ug906-vivado-design-analysis)
3. [Xilinx Support Forum](https://support.xilinx.com/s/question/0D52E00006mibS0SAI/constraining-2-external-clocks-that-are-synchronous?language=en_US)
4. [Vivado Using Constraints (UG 903)](https://docs.xilinx.com/v/u/2013.1-English/ug903-vivado-using-constraints)
5. [Vivado Synthesis (UG 901)](https://docs.xilinx.com/v/u/2021.1-English/ug901-vivado-synthesis)