**Основы статического временного анализа.**

**Часть 5: False Path Constraint.**

**Введение.**

1. **Пересечение тактовых доменов.**

Рассказать, что такое CDC. Провести расчеты Slack. Сослаться на Xilinx Guide.

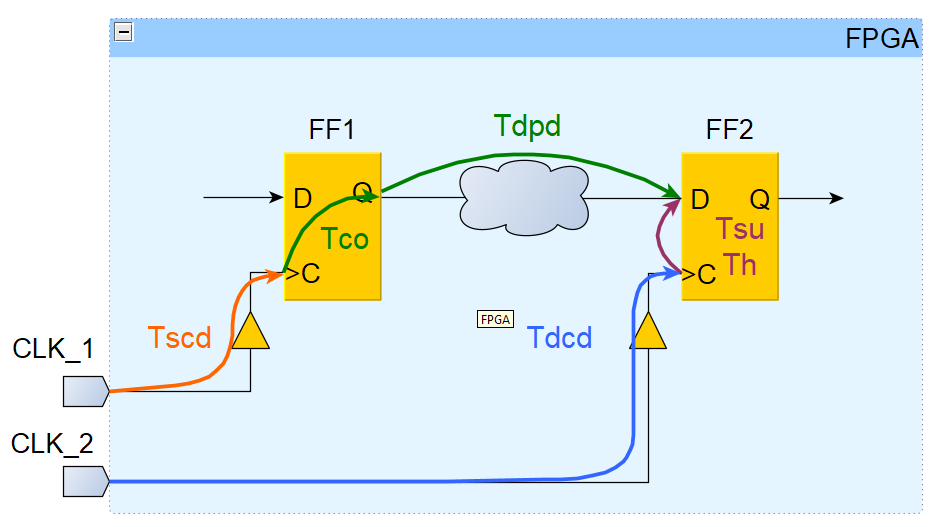


Рисунок 1. Путь с задержками для данных и тактового сигнала.

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 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.

Анализ по Setup

Пусть запускающий фронт появляется на ножке CLK\_1 в момент времени *Tclk\_1*, а защелкивающий фронт появляется на ножке CLK\_2 в момент времени *Tclk\_2*.

* Время прибытия запускающего фронта к триггеру FF1  
  (***S****ource* ***С****lock* ***A****rrival time*):
* Время прибытия данных на вход триггера FF1(***D****ata* ***A****rrival time*):
* Время прибытия защелкивающего фронта к триггеру FF2   
  (***D****estination* ***C****lock* ***A****rrival time*):
* Требуемое время прибытия данных (***D****ata* ***R****equired time*):

Анализ по Hold

* Время прибытия запускающего фронта к триггеру FF1  
  (***S****ource* ***С****lock* ***A****rrival time*):
* Время прибытия данных на вход триггера FF1(***D****ata* ***A****rrival time*):
* Время прибытия защелкивающего фронта к триггеру FF2  
  (***D****estination* ***C****lock* ***A****rrival time*):
* Требуемое время прибытия данных (***D****ata* ***R****equired time*):

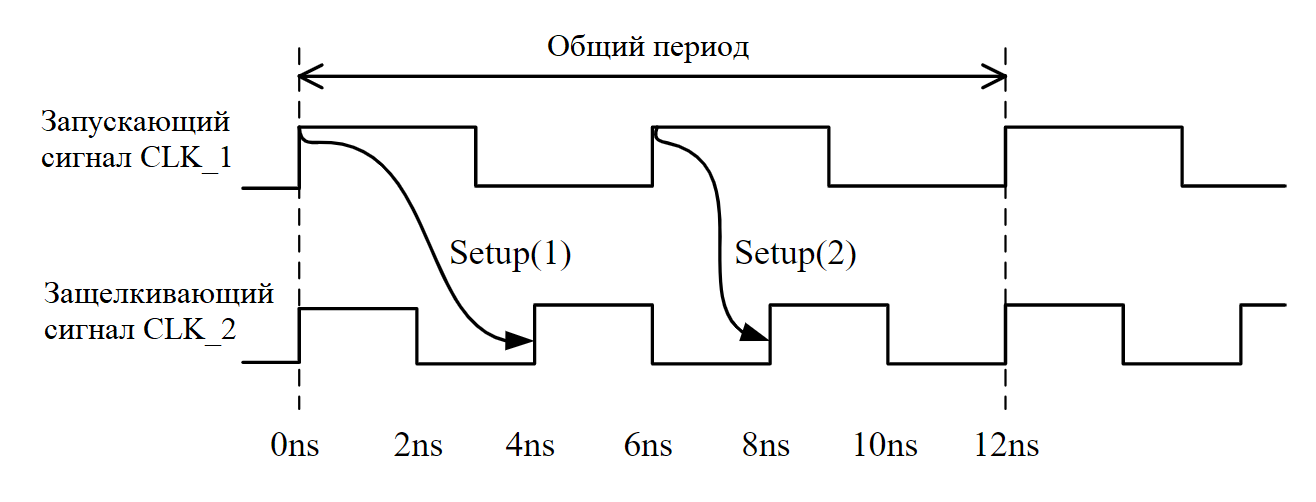


Рисунок 2. Временная диаграмма сигналов для анализа по *Setup*.

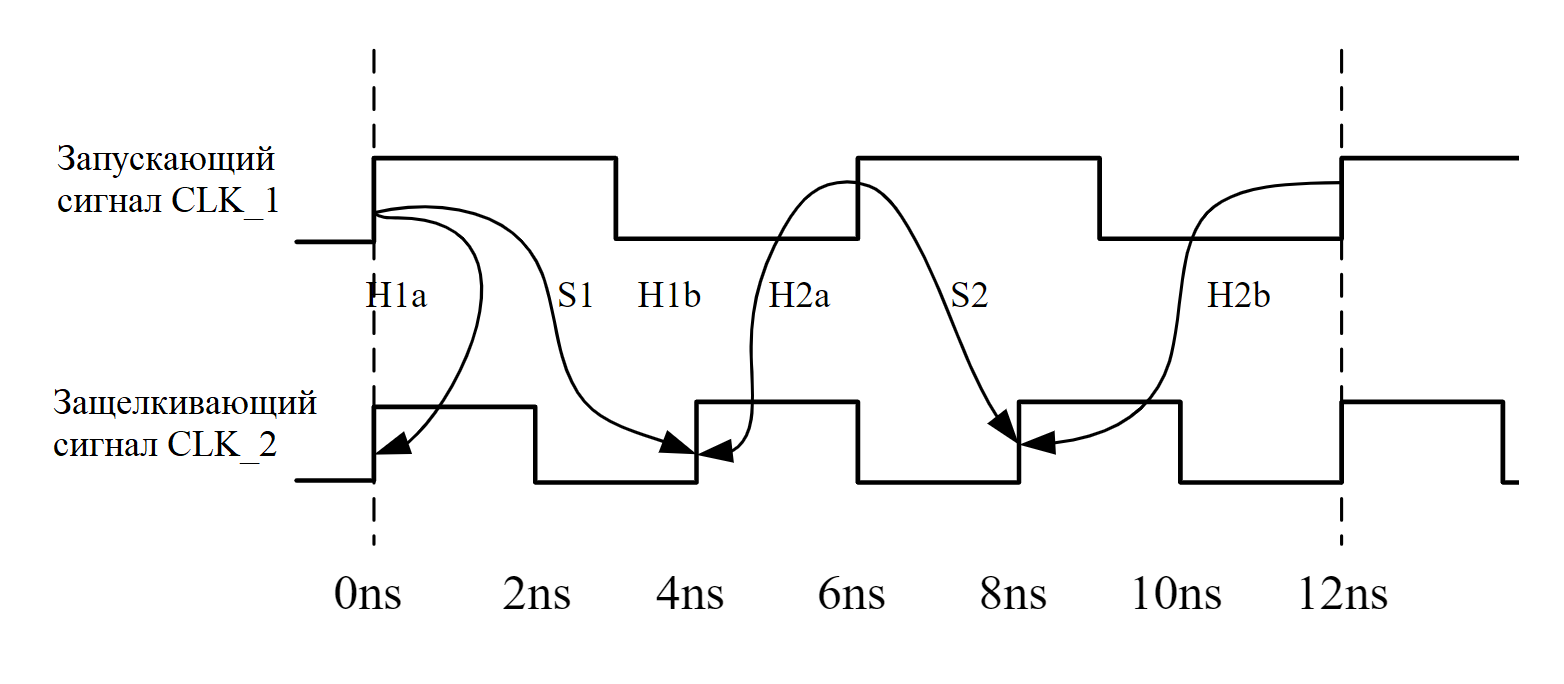


Рисунок 3. Временная диаграмма сигналов для анализа по *Hold*.

1. **Асинхронные тактовые сигналы.**

Рассказать про синхронизаторы и метастабильность. Примеры синхронных и асинхронных сигналов и отсчет clock interection (асинхронные, синхронные с PLL, синхронные с наружи).

module top\_1 (

    input  logic clk\_1,

    input  logic clk\_2,

    input  logic i\_data,

    output logic o\_data

);

    logic cdc\_data;

    // передающий тактовый домен

    always\_ff @(posedge clk\_1)

        cdc\_data <= i\_data;

    // принимающий тактовый домен

    always\_ff @(posedge clk\_2)

        o\_data <= cdc\_data;

endmodule

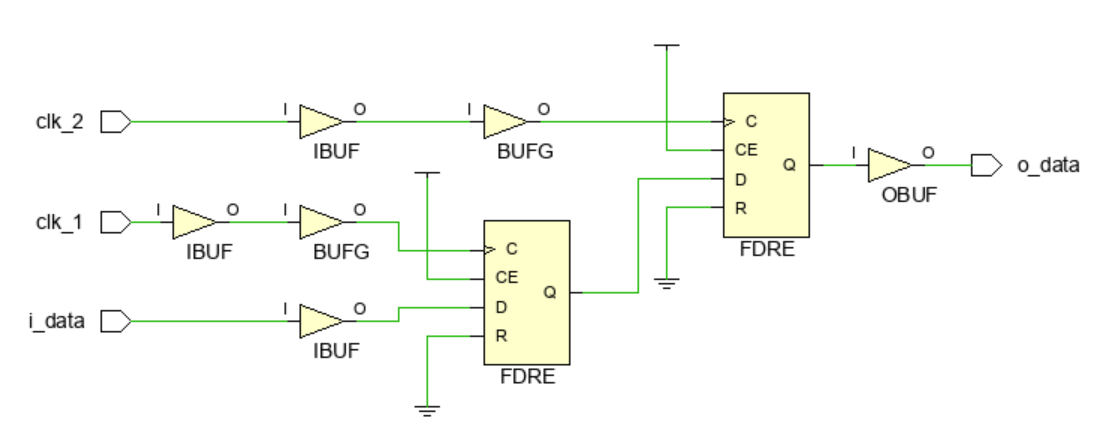


Рисунок 4. Схема проекта.

# период запускающего тактового сигнала

create\_clock -period 10.0 -name clk\_1 [get\_ports clk\_1]

# период защелкивающего тактового сигнала

create\_clock -period 5.0 -name clk\_2 [get\_ports clk\_2]

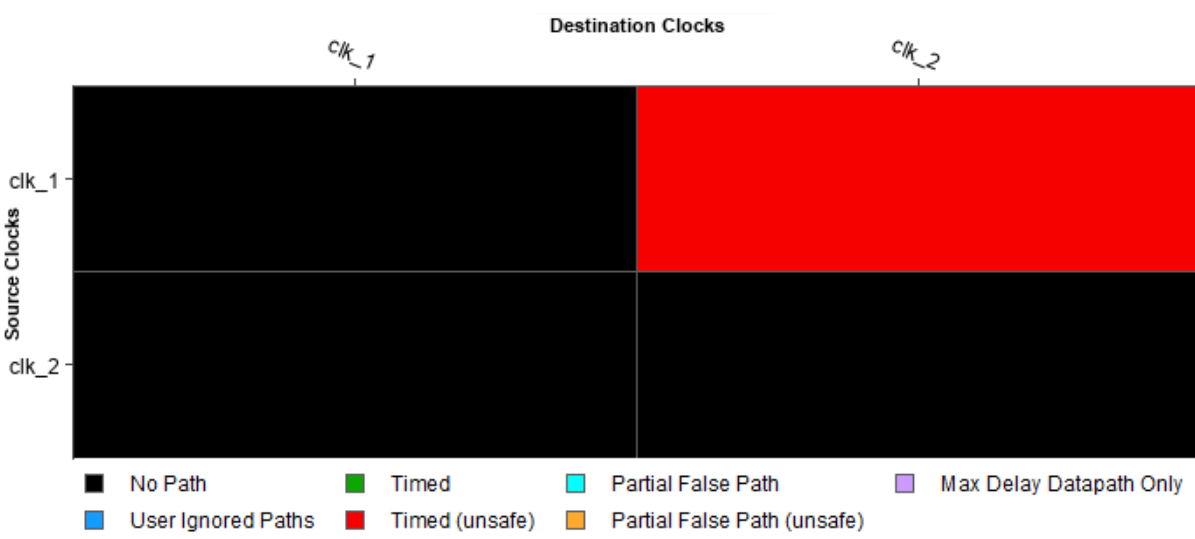


Рисунок 5. Таблица взаимодействия тактовых сигналов.

Clock Interaction Table

-----------------------

Clock-Pair Inter-Clock

From Clock To Clock Classification Constraints

------------ ------------ ------------------- --------------

clk\_1 clk\_2 No Common Clock Timed (unsafe)

module top\_2 (

    input  logic i\_clk,

    input  logic i\_data,

    output logic o\_data

);

    logic clk\_1;

    logic clk\_2;

    logic cdc\_data;

    clk\_wiz\_0 pll

    (

        .i\_clk (i\_clk),

        .clk\_1 (clk\_1),

        .clk\_2 (clk\_2)

    );

    // передающий тактовый домен

    always\_ff @(posedge clk\_1)

        cdc\_data <= i\_data;

    // принимающий тактовый домен

    always\_ff @(posedge clk\_2)

        o\_data <= cdc\_data;

endmodule

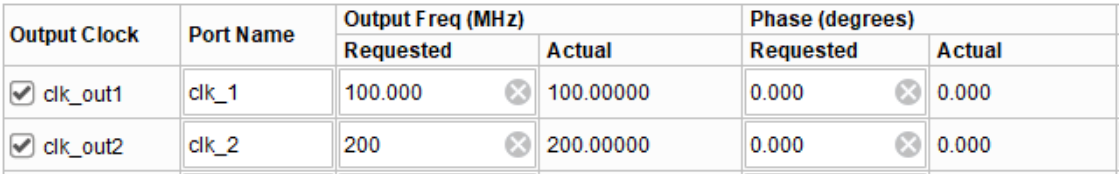


Рисунок 6. Настройки PLL.

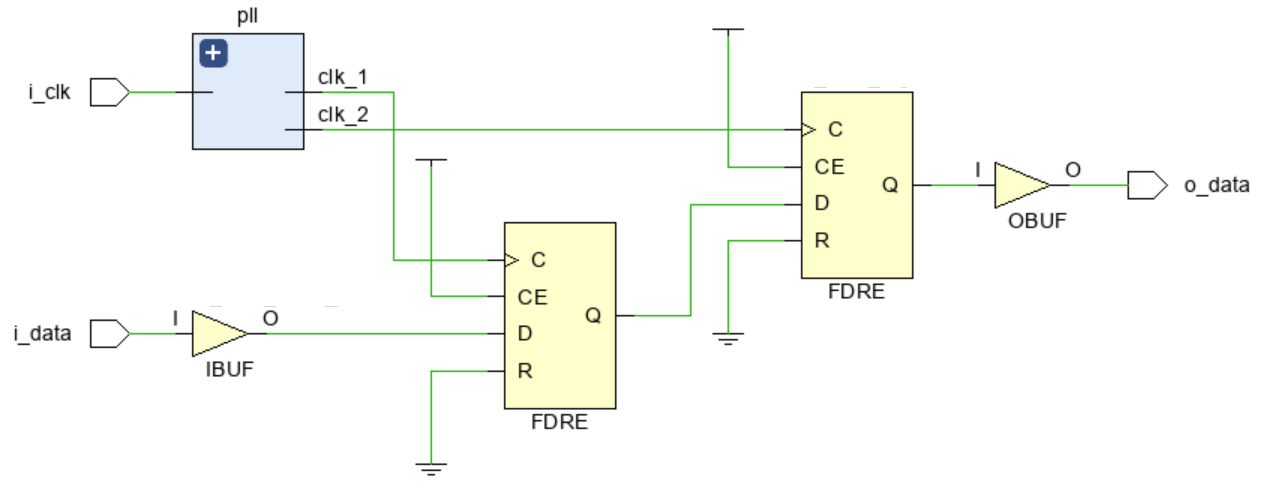


Рисунок 7. Схема проекта.

# 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

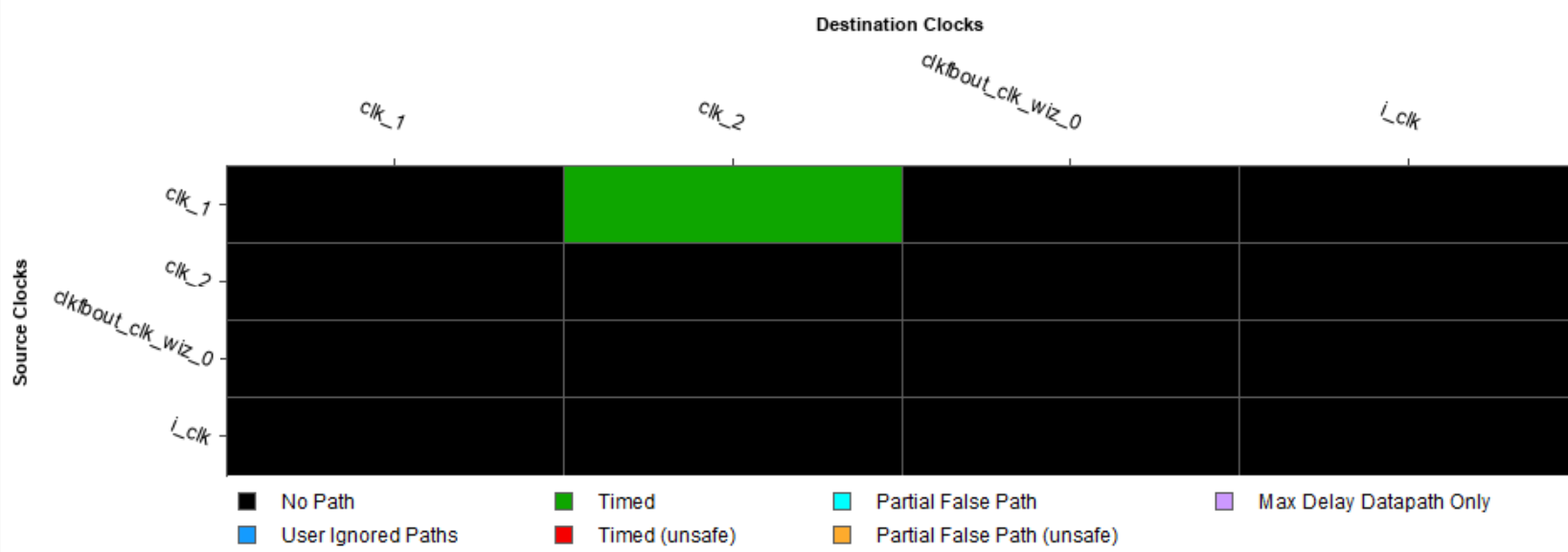


Рисунок 8. Таблица взаимодействия тактовых сигналов.

Clock Interaction Table

-----------------------

Clock-Pair Inter-Clock

From Clock To Clock Classification Constraints

------------ ------------ ------------------- -----------

clk\_1 clk\_2 Clean Timed

1. **False Path Constrains.**

Примеры нарушения временных ограничений из-за асинхронности. Показать False path.

# период запускающего тактового сигнала

create\_clock -period 10.0 -name clk\_1 [get\_ports clk\_1]

# период защелкивающего тактового сигнала

create\_clock -period 5.0 -name clk\_2 [get\_ports clk\_2]

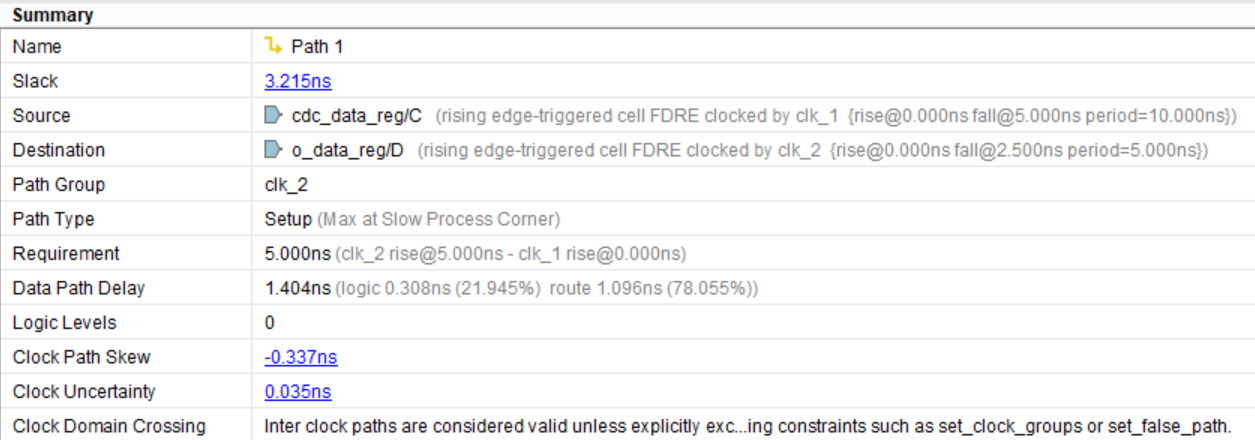


Рисунок 9. Раздел *Summary* временного отчета.

# период запускающего тактового сигнала

create\_clock -period 10.0 -name clk\_1 [get\_ports clk\_1]

# период защелкивающего тактового сигнала

create\_clock -period 33.3 -name clk\_2 [get\_ports clk\_2]

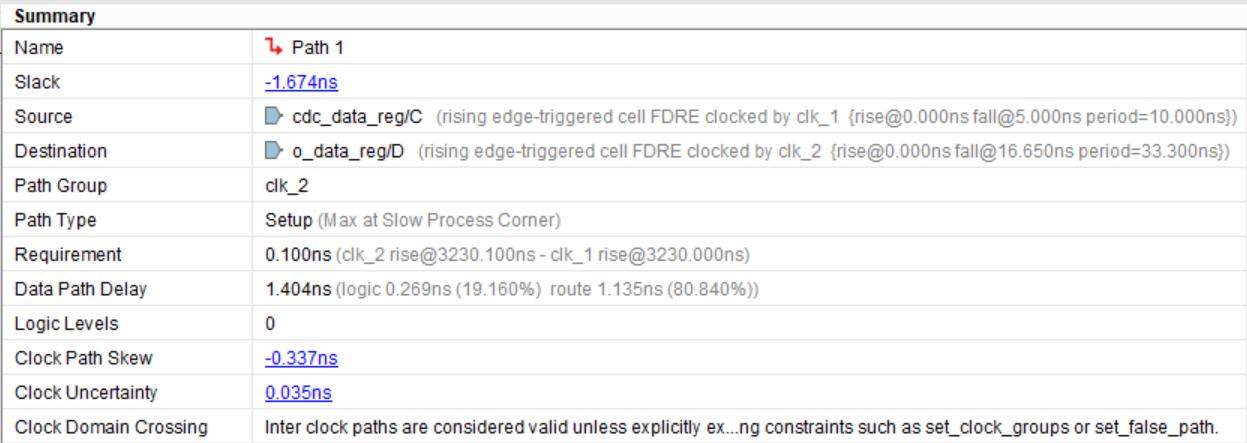


Рисунок 10. Раздел *Summary* временного отчета.

# объявление false path через начало и конец пути

set\_false\_path -from [get\_pins cdc\_data\_reg/C]

-to [get\_pins o\_data\_reg/D]

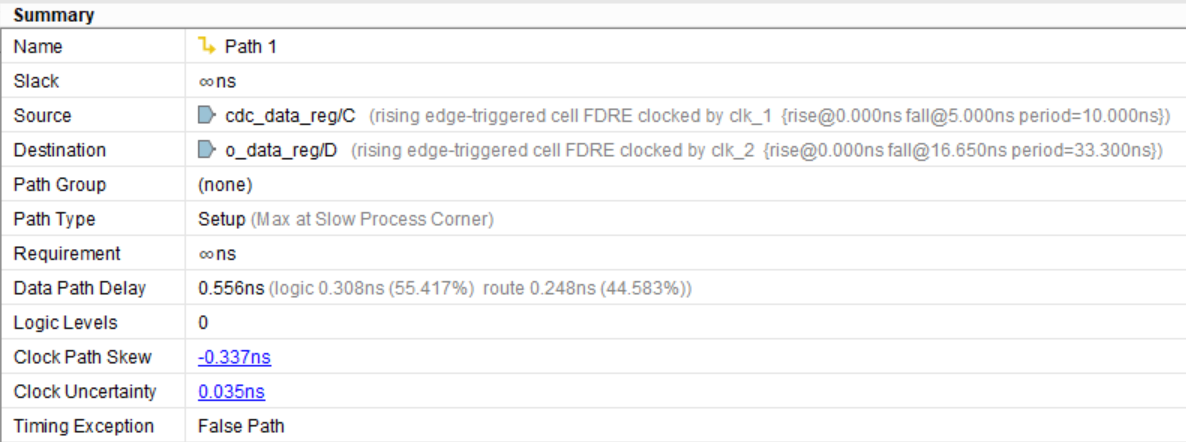


Рисунок 11. Раздел *Summary* временного отчета.

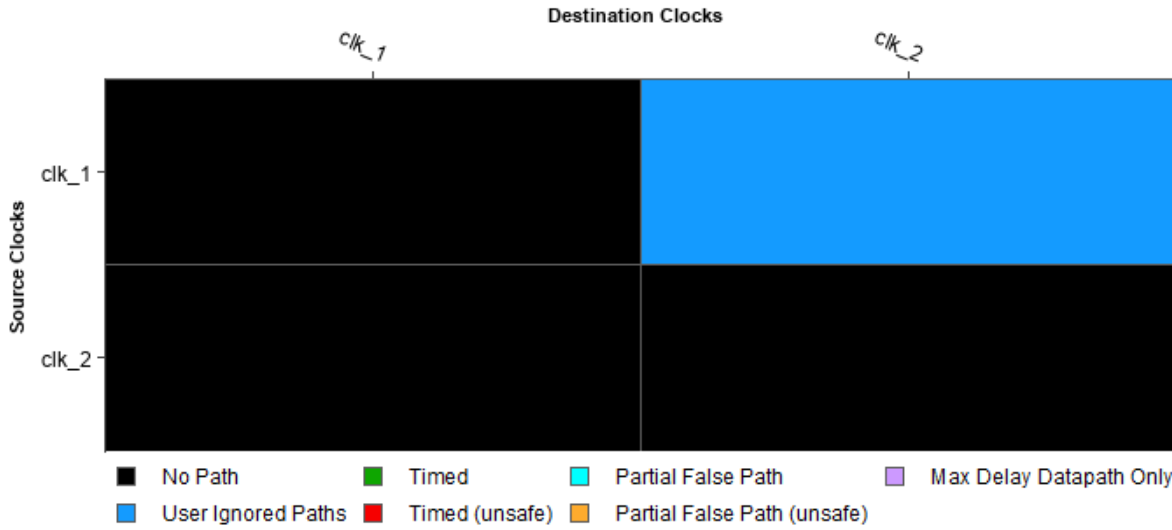


Рисунок 12. Таблица взаимодействия тактовых сигналов.

Clock Interaction Table

-----------------------

Clock-Pair Inter-Clock

From Clock To Clock Classification Constraints

------------ ------------ ------------------- -----------

clk\_1 clk\_2 Ignored False Path

# объявление false path через запускающий и защелкивающий триггеры

set\_false\_path -from [get\_cells cdc\_data\_reg]

-to [get\_cells o\_data\_reg]

# объявление false path через имена тактовых доменов

set\_false\_path -from [get\_clocks clk\_1] -to [get\_clocks clk\_2]

# объявление тактовых доменов асинхронными

set\_clock\_groups -name cdc\_async -group [get\_clocks clk\_1]

-group [get\_clocks clk\_2] -asynchronous

Clock Interaction Table

-----------------------

Clock-Pair Inter-Clock

From Clock To Clock Classification Constraints

------------ ------------ ------------------- -------------------

clk\_1 clk\_2 Ignored Asynchronous Groups

1. **Report CDC.**

Показать report\_cdc. Рассказать о работе синхронизатора.

CDC Report

Severity Source Clock Destination Clock Exceptions

-------- ------------ ----------------- -------------------

Critical clk\_1 clk\_2 Asynch Clock Groups

Endpoints Safe Unsafe Unknown No ASYNC\_REG

--------- ---- ------ ------- ------------

1 0 0 1 0

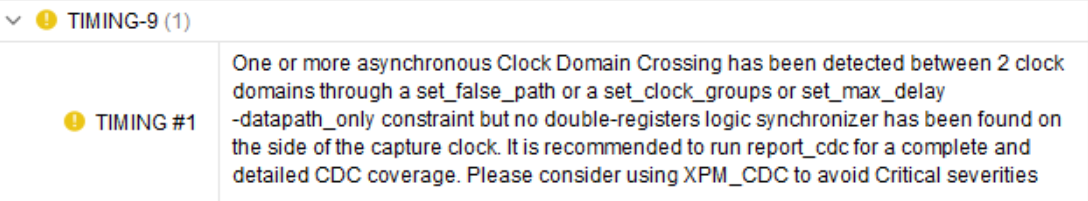


Рисунок 13. Отчет о проверки методологии проектирования.

module top\_3 (

    input  logic clk\_1,

    input  logic clk\_2,

    input  logic i\_data,

    output logic o\_data

);

    logic cdc\_data, sync\_data;

    // передающий тактовый домен

    always\_ff @(posedge clk\_1)

        cdc\_data <= i\_data;

    // принимающий тактовый домен

    always\_ff @(posedge clk\_2) begin

        sync\_data <= cdc\_data;

        o\_data <= sync\_data;

    end

endmodule

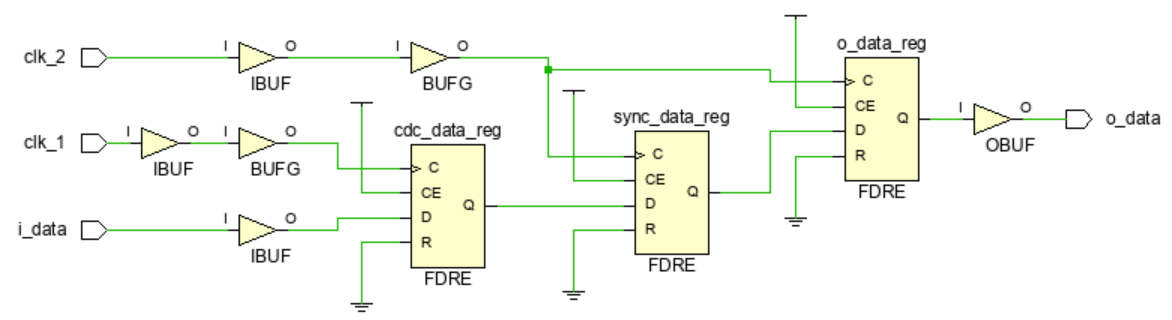


Рисунок 14. Схема проекта.

CDC Report

Severity Source Clock Destination Clock Exceptions

-------- ------------ ----------------- -------------------

Warning clk\_1 clk\_2 Asynch Clock Groups

Endpoints Safe Unsafe Unknown No ASYNC\_REG

--------- ---- ------ ------- ------------

1 1 0 0 1

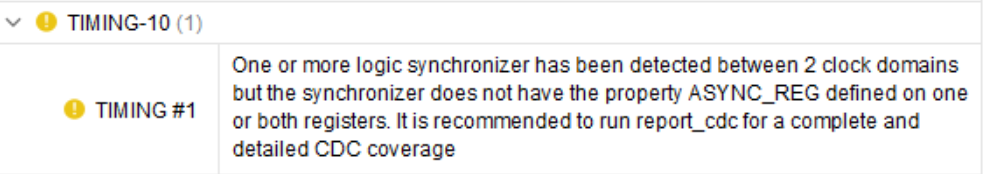


Рисунок 15. Отчет о проверки методологии проектирования.

1. **Свойство ASYNC.**

Рассказать про свойство async и про оптимизации. Рассказать про готовые синхронизаторы от Xilinx.

# период запускающего тактового сигнала

create\_clock -period 10.0 -name clk\_1 [get\_ports clk\_1]

# период защелкивающего тактового сигнала

create\_clock -period 33.3 -name clk\_2 [get\_ports clk\_2]

# объявление тактовых доменов асинхронными

set\_clock\_groups -name cdc\_async -group [get\_clocks clk\_1]

-group [get\_clocks clk\_2] -asynchronous

# объявление синхронизирующих триггеров

set\_property ASYNC\_REG true [get\_cells {sync\_data\_reg o\_data\_reg}]

CDC Report

Severity Source Clock Destination Clock Exceptions

-------- ------------ ----------------- -------------------

Info clk\_1 clk\_2 Asynch Clock Groups

Endpoints Safe Unsafe Unknown No ASYNC\_REG

--------- ---- ------ ------- ------------

1 1 0 0 0

module top\_4 (

    input  logic clk\_1,

    input  logic clk\_2,

    input  logic i\_data,

    output logic o\_data

);

    logic cdc\_data, sync\_data;

    // передающий тактовый домен

    always\_ff @(posedge clk\_1)

        cdc\_data <= i\_data;

    // XPM синхронизатор

    xpm\_cdc\_single #(

        .DEST\_SYNC\_FF(2),

        .SRC\_INPUT\_REG(0)

    )

    xpm\_synchronizer (

        .src\_clk(clk\_1),

        .src\_in(cdc\_data),

        .dest\_clk(clk\_2),

        .dest\_out(sync\_data)

    );

    // принимающий тактовый домен

    always\_ff @(posedge clk\_2)

        o\_data <= sync\_data;

endmodule

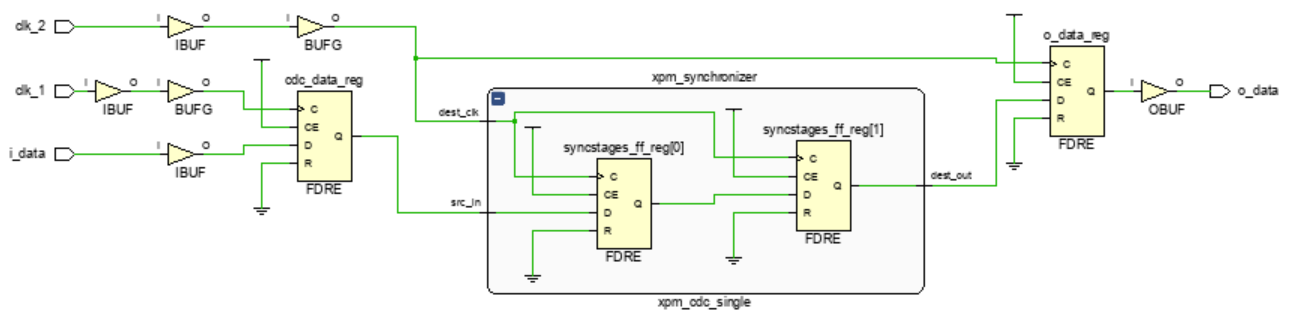


Рисунок 16. Схема проекта.

# период запускающего тактового сигнала

create\_clock -period 10.0 -name clk\_1 [get\_ports clk\_1]

# период защелкивающего тактового сигнала

create\_clock -period 33.3 -name clk\_2 [get\_ports clk\_2]

CDC Report

Severity Source Clock Destination Clock Exceptions

-------- ------------ ----------------- -------------------

Info clk\_1 clk\_2 False Path

Endpoints Safe Unsafe Unknown No ASYNC\_REG

--------- ---- ------ ------- ------------

1 1 0 0 0

**Заключение.**

**Ссылки.**