**Основы статического временного анализа.**

**Часть 4: Source Synchronous** **Output Delay Constraint.**

**Введение.**

В статье представлен временной анализ передачи сигналов из FPGA во внешнее устройство. Рассмотрены теоретические основы анализа для двух вариантов защелкивания данных. Разобраны практические примеры создания временных ограничений. Показан способ решения проблем с ограничениями по *Setup* за счет инвертирования тактового сигнала и использования ODDR триггера.

1. **Передача данных для случая Source Synchronous.**

Данная статья частично опирается на материал, рассмотренный в предыдущих работах [1-2]. Предполагается, что читатель уже знаком с такими понятиями, как ограничение на максимальное (*Setup*) и минимальное (*Hold*) время распространения сигнала, запас (*Slack*) и т.д.

Ранее в [2] был представлен временной анализ передачи данных из FPGA во внешнее устройство в случае, когда тактовый сигнал формируется генератором, расположенным на плате (см. рисунке 1). Такой способ передачи называется System Synchronous.

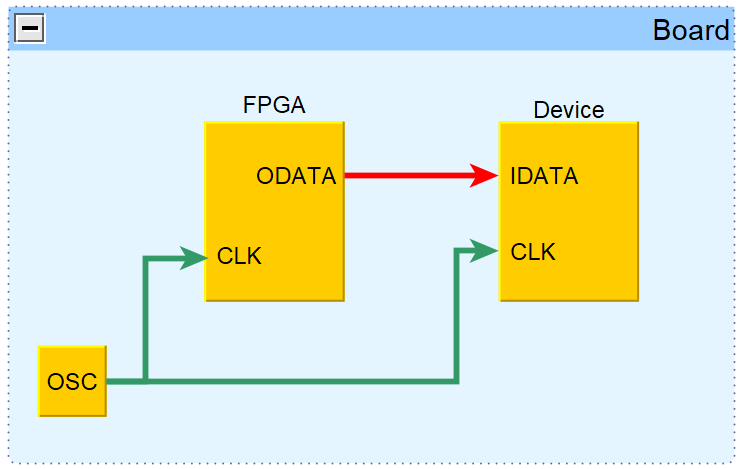


Рисунок 1. Соединение устройств на плате для случая System Synchronous.

В текущей статье будет рассмотрен другой способ, называемый Source Synchronous, при котором источник помимо данных также формирует тактовый сигнал (см. рисунке 2). В дальнейшем для краткости устройство, в которое из FPGA передаются данные и тактовый сигнал, будем иногда называть Device*.*

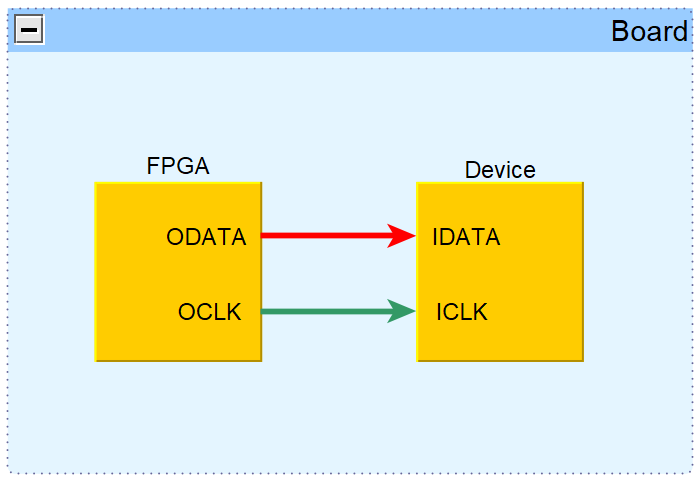


Рисунок 2. Соединение устройств на плате для случая Source Synchronous.

При передаче данных между FPGA и Device запускающий триггер располагается в FPGA, а защелкивающий – во внешнем устройстве. На рисунке 4 показан анализируемый путь, на который нанесены задержки сигналов. В случае Source Synchronous вместе с данными передает тактовый сигнал, которые может внутри FPGA может формироваться с помощью PLL или браться непосредственно с входной тактовой ножки.

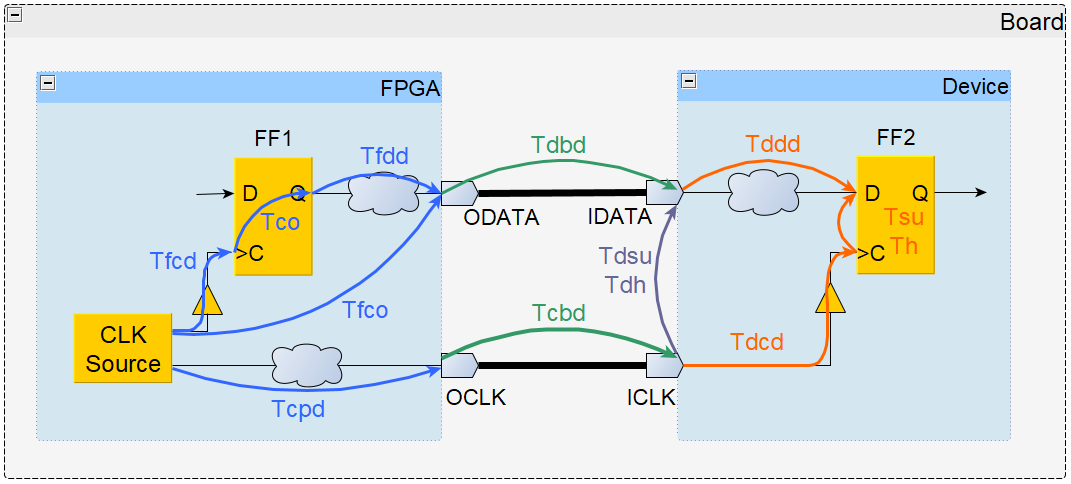


Рисунок 3. Путь с задержками для выходных данных и тактового сигнала.

Ниже даны определения задержек, представленных на рисунке 3.

* *Tdbd* (***D****ata* ***B****oard* ***D****elay*) – задержка распространения данных по дорожкам платы от FPGA до Device;
* *Tcbd* (***C****lock* ***B****oard* ***D****elay*) – задержка распространения тактового сигнала по дорожкам платы от FPGA до Device;
* *Tdcd* (***D****evice* ***C****lock* ***D****elay*) – задержка тактового сигнала от ножки ICLK Device до тактового входа защелкивающего триггера;
* *Tddd* (***D****evice* ***D****ata* ***D****elay*) – задержка распространения данных от ножки IDATA Device до защелкивающего триггера;
* *Tsu* (***S****et****U****p time*) – время установки защелкивающего триггера;
* *Th* (***H****old time*) – время удержания защелкивающего триггера;
* *Tcpd* (***C****lock to* ***P****in* ***D****elay*) – задержка тактового сигнала от места его формирования до ножки OCLK FPGA;
* *Tfcd* (***F****PGA* ***C****lock* ***D****elay*) – задержка тактового сигнала от места его формирования до тактового входа запускающего триггера;
* *Tco* (***C****lock to* ***O****utput*) – интервал времени между приходом фронта на тактовый вход триггера и появлением данных на его выходе Q;
* *Tfdd* (***F****PGA* ***D****ata* ***D****elay*) – задержка распространения данных от выходе Q запускающего триггера ножки ODATA FPGA.

Период тактового сигнала будем обозначать *Tclk*. Оранжевым и зеленым цветом на рисунке 4 представлены задержки для участков пути, которые располагаются вне FPGA. Эти 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.

Глядя на рисунок 3, можно определить фактическое время прибытия данных, как сумму задержек для запускающего фронта и задержек распространения данных

где введено обозначение (*Tfco* – ***F****PGA* ***C****lock to* ***O****utput time*):

В дальнейшем будет рассмотрено два варианта передачи данных между FPGA и Device. В первом случае данные будет запускаться из FPGA по одному фронту, а защелкиваться в Device - по следующему. При втором варианте данные будут запускаться и защелкиваться по одному и тому же тактовому фронту. Чтобы описать оба эти варианта введем дополнительную переменную *Tccd\_setup* (***C****apture* ***C****lock* ***D****elay*), которая описывает интервал времени между запускающим и защелкивающим фронтами.

Тогда из рисунка 3 можно получить требуемое время прибытия данных, как разницу между временем прибытия защелкивающего фронта и временем удержания триггера.

где *Tclk\_delay\_min* – задержка распространения тактового сигнала от его источника в FPGA до входа защелкивающего триггера в Device

Уравнение для *Slack* при анализе по *Setup* имеет вид [1]:

Подставим в него уравнения (1) и (2) и получим:

где *Tdsu (****D****evice* ***S****e****T****up)*– время установке для данных на входе IDATA Device относительно тактового входа ICLK (см. рисунок 3):

Объединим все слагаемые, описывающие задержки вне FPGA, в одну переменную и получим окончательное выражения для *Slack* при анализе по *Setup*:

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.

По аналогии с анализом по *Setup* уравнения для фактического времени прибытия данных имеют вид:

Найдем требуемое время прибытия данных. Для этого нужно рассчитать максимальную задержку распространения защелкивающего фронта и добавить к ней время удержания триггера [1]:

Переменная *Tccd\_hold* описывает интервал времени между текущим запускающим и предыдущим защелкивающим фронтами тактового сигнала.

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

С учетом полученных ранее результатов получим:

где *Tdh (****D****evice* ***H****old)* – время удержания для данных на входе IDATA Device относительно тактового входа ICLK (см. рисунок 3):

Объединим все задержки вне FPGA в одну переменную и получим окончательное выражения для анализа по *Hold*:

1. **Пример с защелкивание данных по следующему фронту.**

module top (

    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

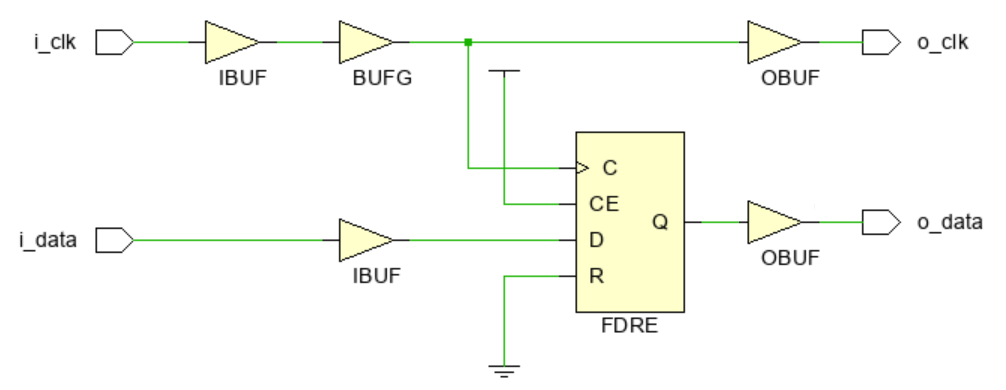
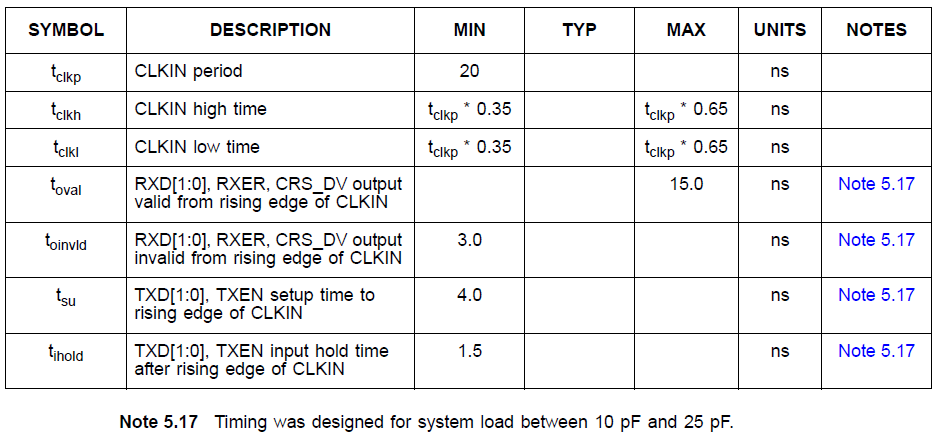


Рисунок 3. Схема FPGA проекта.

Также данные, которые требуется принять, отмечены цветом.



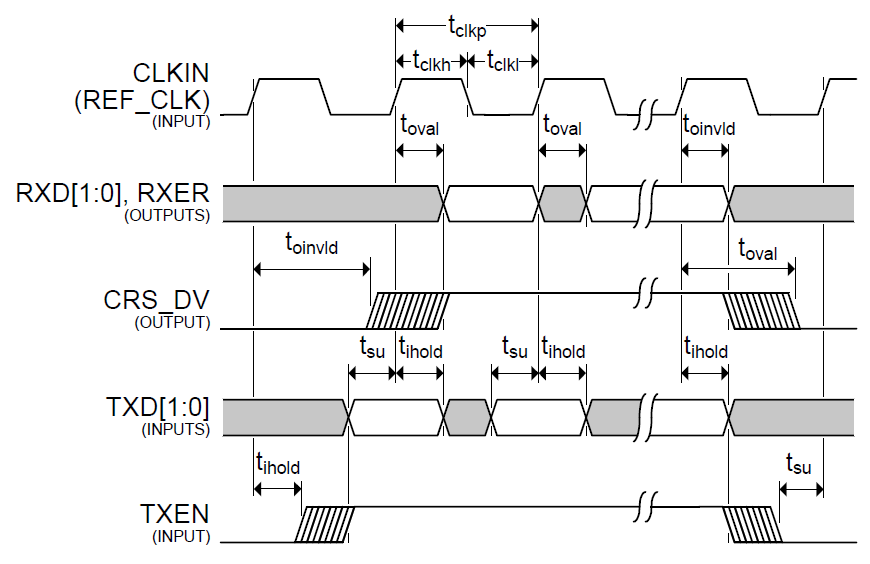


Рисунок 10. Задержки и временные диаграммы для LAN8740A.

Из таблицы задержек получаем, что старые данные удерживаются на шине RXD в течении *Тinvld =* 10 нс после фронта №1. Также видно, что после того же фронта нужные данные появляются на выходе LAN8740A спустя   
*Тval =* 28 нс. Будем считать, что LAN8740A работает режиме 100BASE-TX, и зададим те же, что и в предыдущем примере, значения задержек распространения сигналов по дорожкам платы. В этом случае в xdc-файл следует занести следующие команды:

# период тактового сигнала

set Tclk 40

# ограничение на период тактового сигнала

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

# 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

# время удержания старых данных после тактового сигнала на выходе LAN8740A

set Tinvld 10

# время между появлением тактового сигнала и данных на выходе LAN8740A

set Tval 28

Если перерисовать временные диаграммы, соблюдая масштабы задержек, то способ выравнивания тактового сигнала относительно данных станет более очевидным. Однако давайте просто последовательно разберем все три возможных шаблона создания временных ограничений.

Сначала попробуем защелкнуть интересующие нас данные по тактовому фронту №1, что соответствует случаю Edge Aligned. Если сопоставить временные диаграммы для LAN8740A с рисунком 6, то становится очевидно равенство *Tare\_dev = Тval*. На рисунке 6 предполагается, что старые данные пропадают с выхода Device до появления защелкивающего фронта, а на диаграммах для LAN8740A – после. Это расхождение можно учесть с помощью знака задержки, то есть *Tbre\_dev =* –*Тinvld.* Тогда, если вспомнить уравнения (11), то оставшаяся часть xdc-файла примет вид:

# исчезновения старых данных до тактового сигнала на выходе Device

set Tbre\_dev -$Tinvld

# появления новых данных после тактового сигнала на выходе Device

set Tare\_dev $Tval

# исчезновения старых данных до тактового сигнала на входе FPGA

set Tbre\_fpga [expr $Tbre\_dev + $Tcbd\_max - $Tdbd\_min]

# появления новых данных после тактового сигнала на входе FPGA

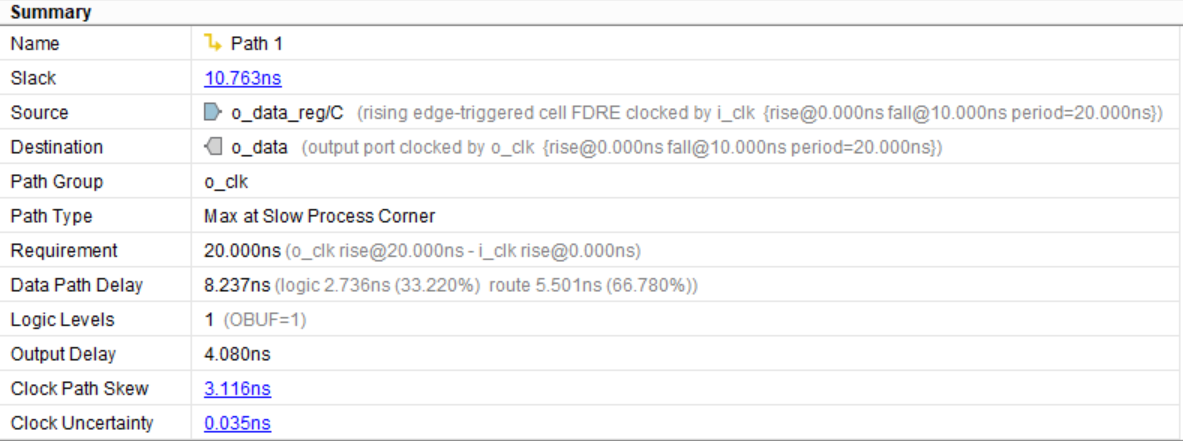
set Tare\_fpga [expr $Tare\_dev + $Tdbd\_max - $Tcbd\_min]

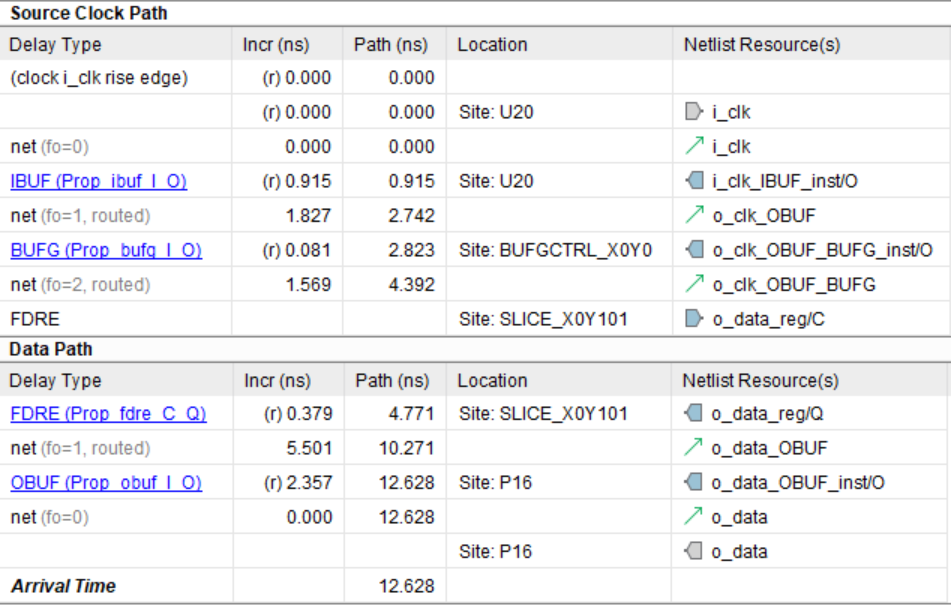
# временные ограничение для входного сигнала IDATA

set\_input\_delay -clock ICLK -max [expr $Tclk + $Tare\_fpga] [get\_ports IDATA]

set\_input\_delay -clock ICLK -min [expr $Tclk - $Tbre\_fpga] [get\_ports IDATA]

В дальнейшем для краткости будем рассматривать только анализ по *Setup*. На рисунке 11 показан раздел Summary временного отчета, в котором отрицательное значение *Slack* указывает на нарушение временных ограничений. Если считать, что фронт №1 формируется на выходе генератора в нулевой момент времени, то на входе FPGA он появится спустя   
*Tcbd\_min* = 0.07 нс. Передаваемые данные дойдут до FPGA в момент времени *Тval + Tdbd\_max* = 28.15 нс. Такой большой разброс между временем прихода защелкивающего фронта и данных является причиной отрицательного значения *Slack =* – 27.199 нс.





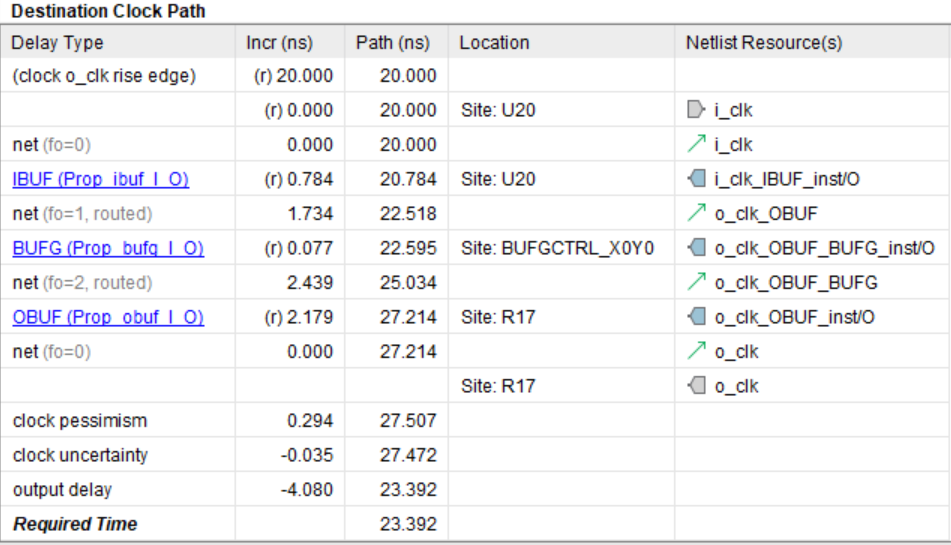


Рисунок 11. Результаты временного анализа (Edge Aligned).

С этой проблемой можно справиться, если защелкивать данный по фронту №2, что соответствует случаю Edge Aligned MMCM. Для создания ограничений теперь необходимо использовать уравнения (12). Содержимое xdc-файла почти полностью совпадает с ограничениями для Edge Aligned и представлено ниже:

# исчезновения старых данных до тактового сигнала на выходе Device

set Tbre\_dev -$Tinvld

# появления новых данных после тактового сигнала на выходе Device

set Tare\_dev $Tval

# исчезновения старых данных до тактового сигнала на входе FPGA

set Tbre\_fpga [expr $Tbre\_dev + $Tcbd\_max - $Tdbd\_min]

# появления новых данных после тактового сигнала на входе FPGA

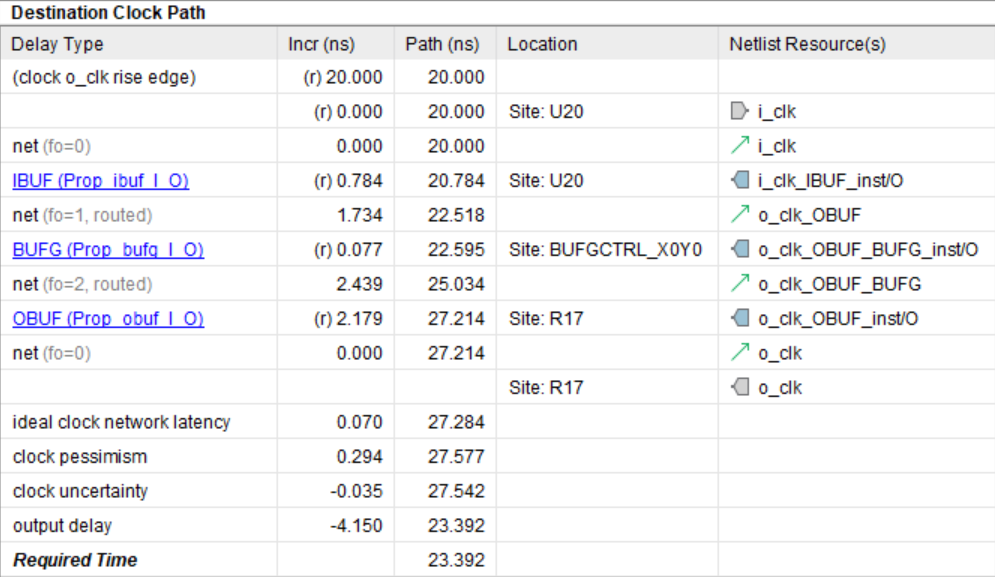
set Tare\_fpga [expr $Tare\_dev + $Tdbd\_max - $Tcbd\_min]

# временные ограничение для входного сигнала IDATA

set\_input\_delay -clock ICLK -max $Tare\_fpga [get\_ports IDATA]

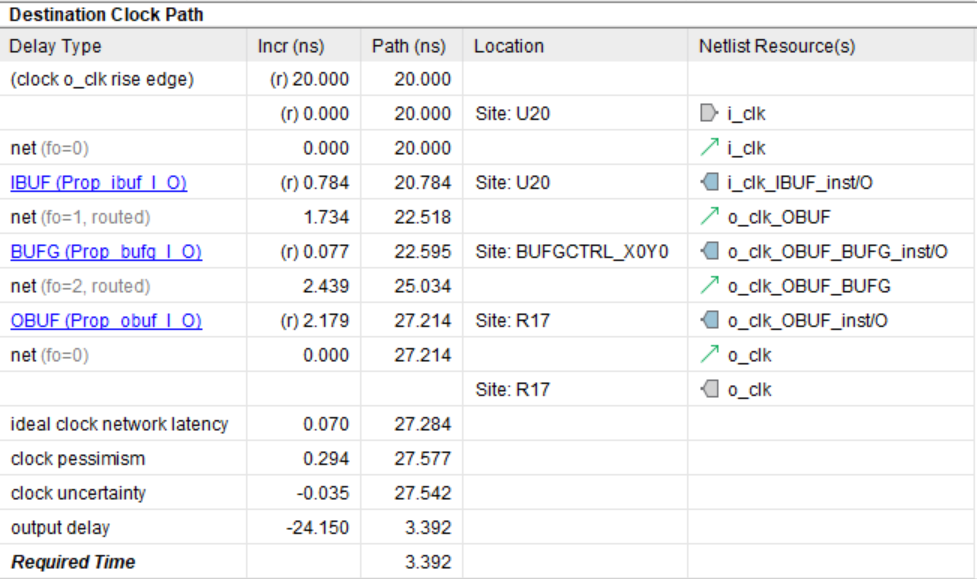
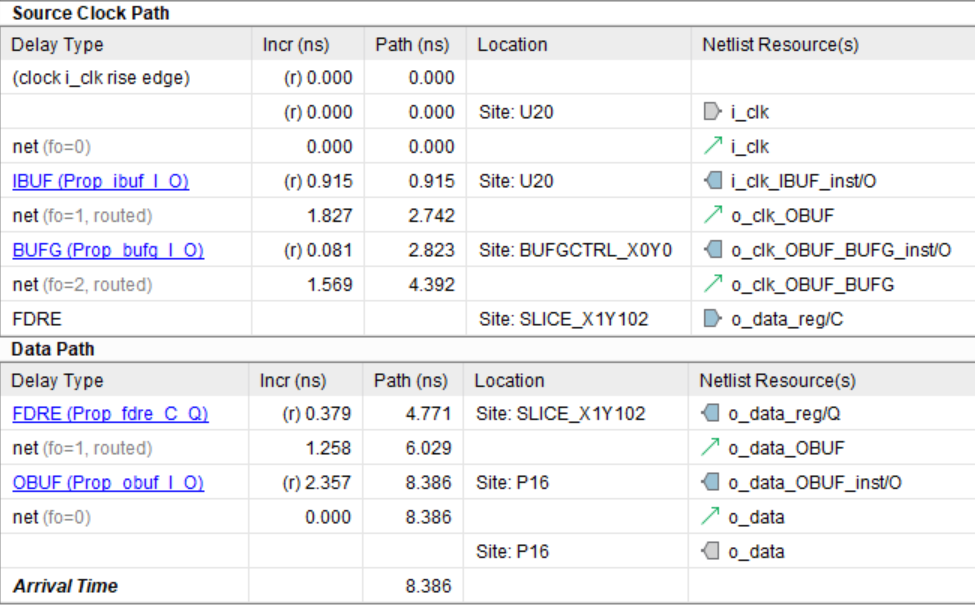
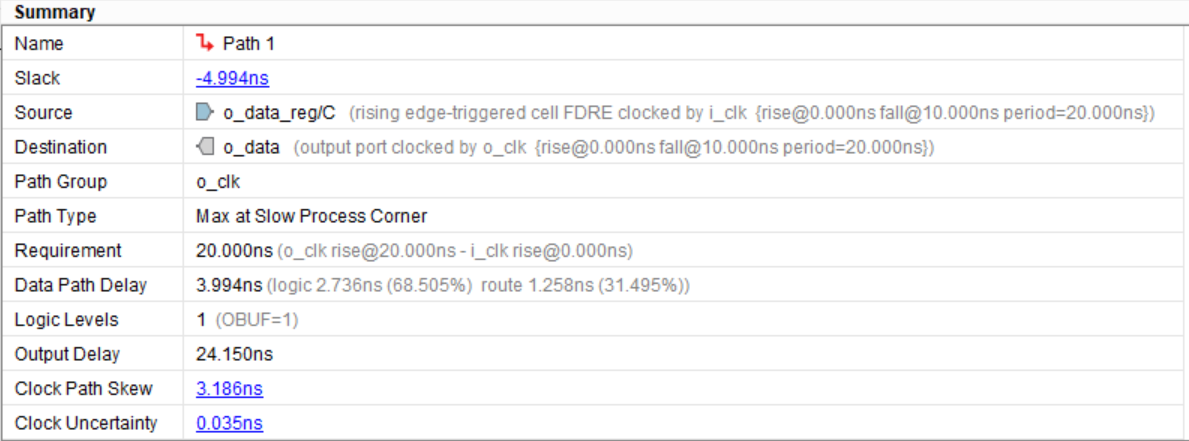
set\_input\_delay -clock ICLK -min -$Tbre\_fpga [get\_ports IDATA]

На рисунке 12 показан раздел S



1. **Пример с защелкивание данных по текущему фронту.**

Ремен



1. **Инвертирование тактового сигнала.**

Реме

module top\_2 (

    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

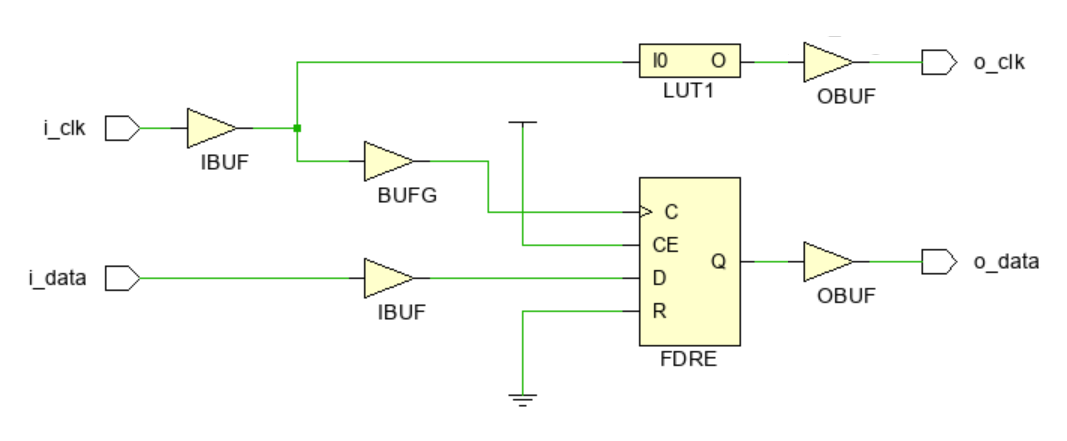
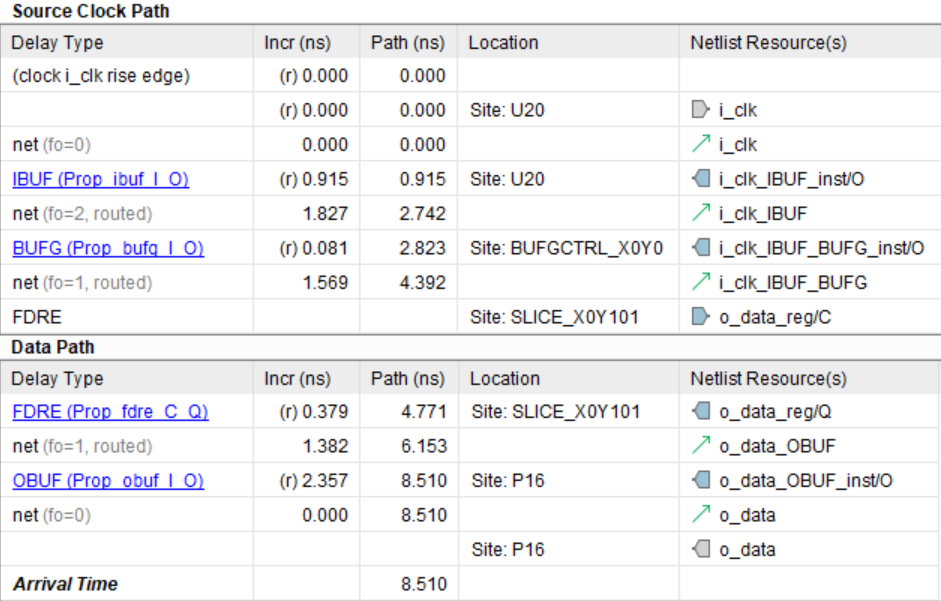
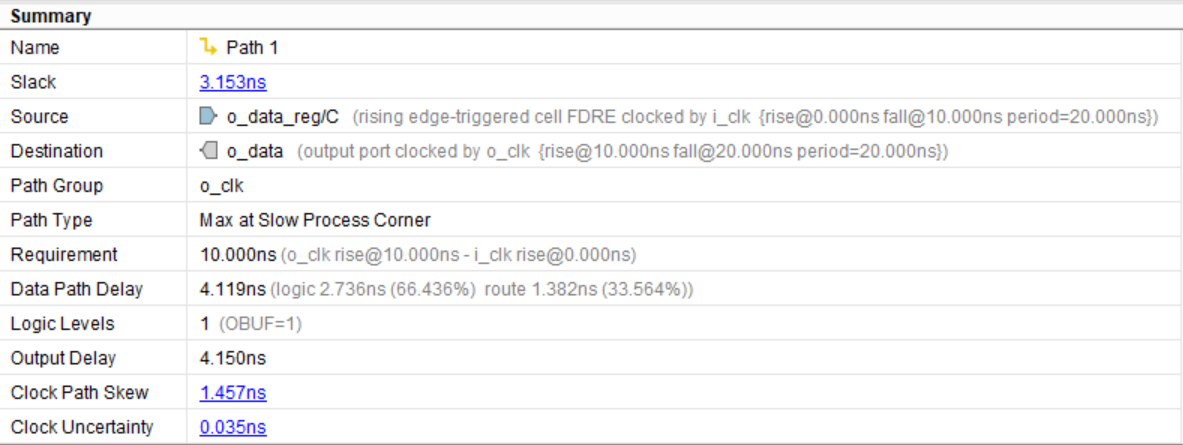
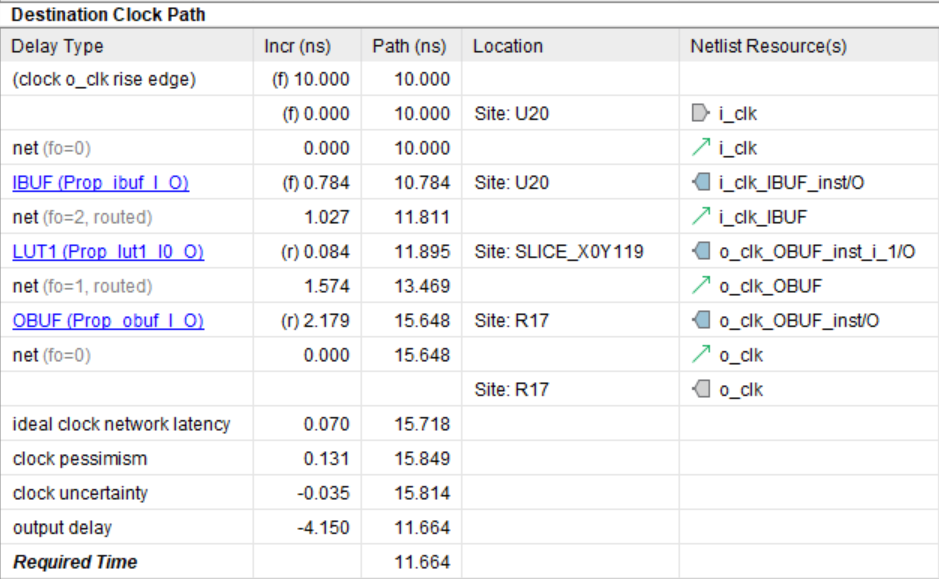


Рисунок 3. Схема FPGA проекта.



нному



1. **Использование ODDR триггера.**

Реме

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;

    ODDR ODDR\_reg (

        .C(i\_clk),

        .D1(1'b0),

        .D2(1'b1),

        .Q(o\_clk),

        .CE(1'b1),

        .R((1'b0),

        .S((1'b0)

    );

endmodule

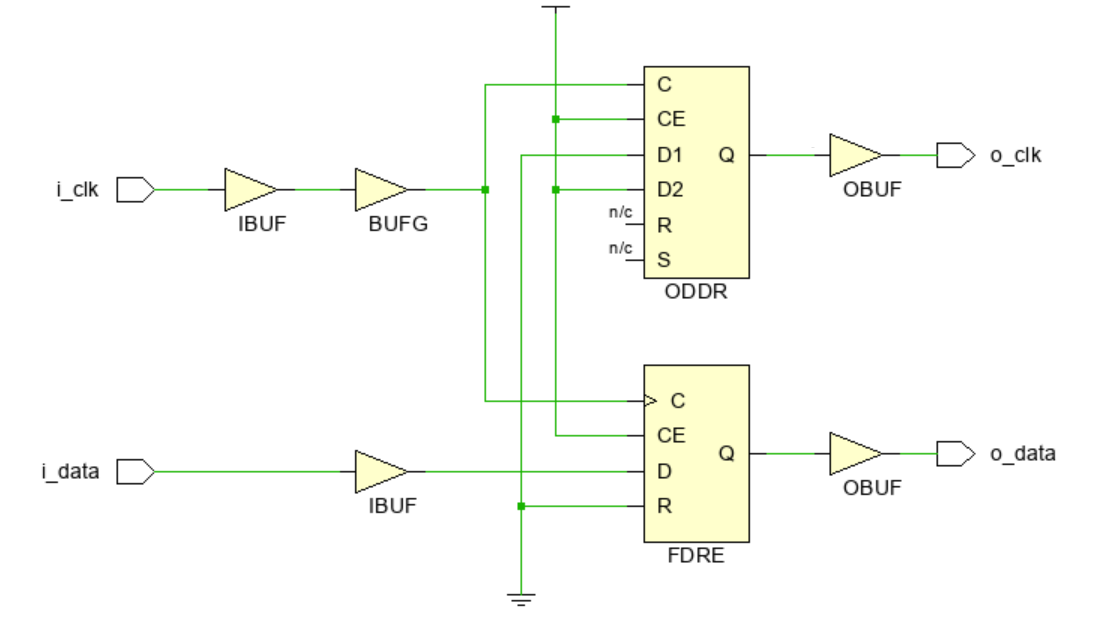
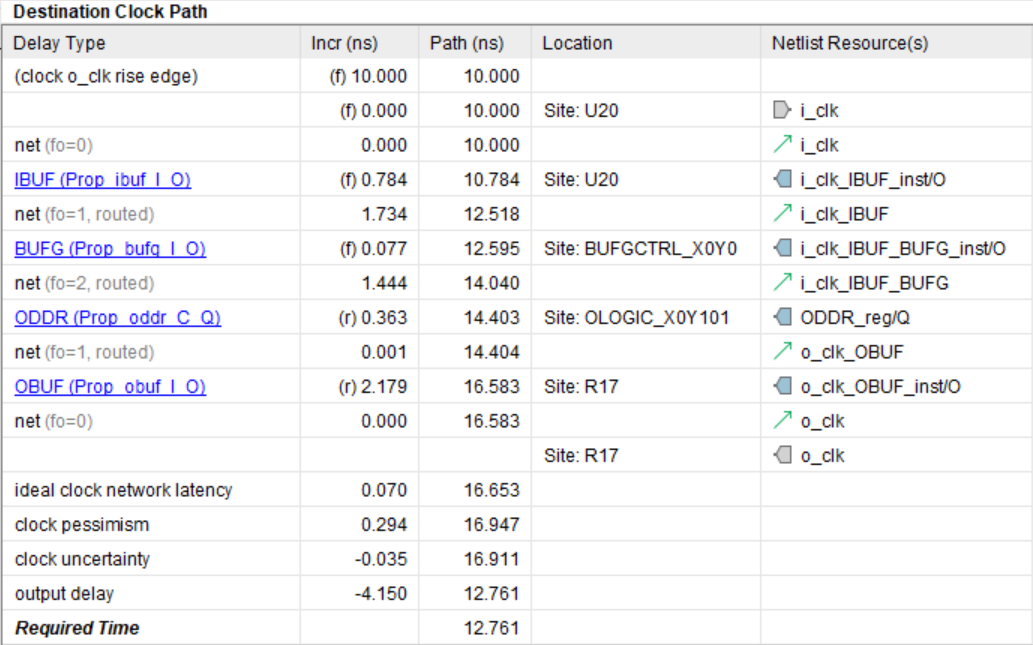
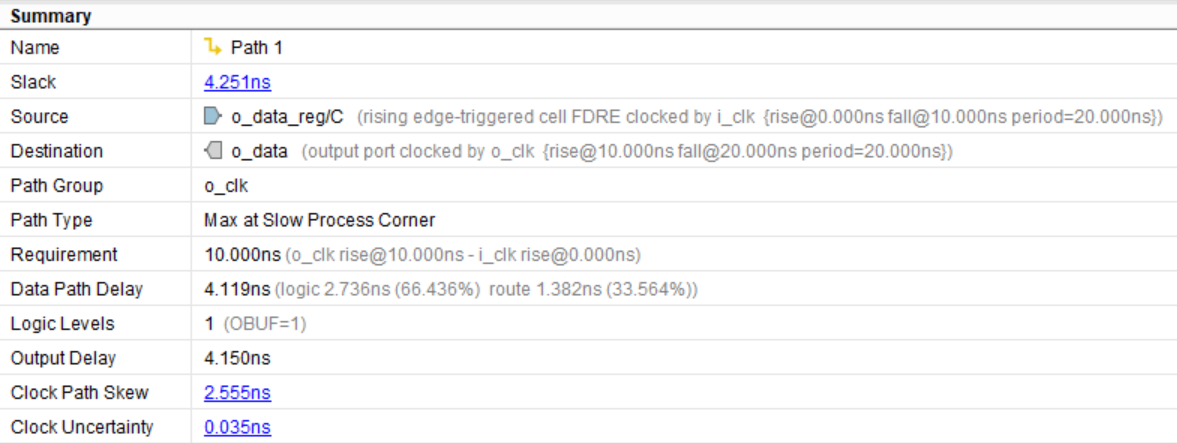


Рисунок 3. Схема FPGA проекта.



нному

**Заключение.**

ременному

**Ссылки.**

ременному