**Основы статического временного анализа: period constraint.**

**Введение.**

Что будет в статье.

**Статический временной анализ.**

Во всех гайдах с рекомендациями написано избегать защелок (latch), поэтому правильно FPGA в который загружен правильно сделанный проект представляет из себя синхронное последовательное цифровое устройство. То есть схема проекта состоит из набора регистров разделенных комбинационной логики. В регистрах содержится состояние системы, изменяющееся только по фронту (спаду) тактового сигнала. Типичный путь сигнала внутри FPGA представлен на рис. 1.

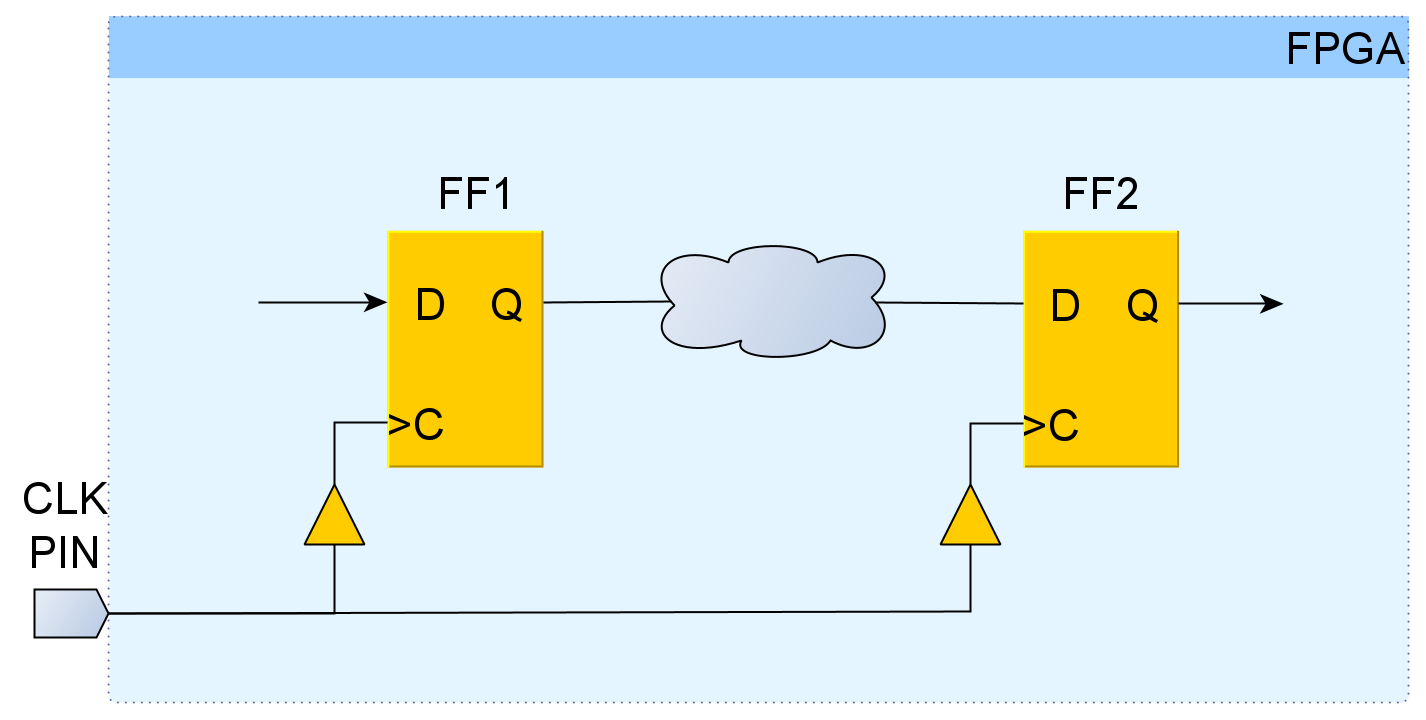


Рисунок 1. Типичный путь сигнала внутри FPGA.

По фронту тактового сигнала, который называют запускающий (start, source, launch), данные с D входа триггера FF1 переходят на выход Q, распространяются через комбинационную логику и попадают на D вход триггера FF2. Спустя период тактового сигнала приходит следующий фронт тактового сигнала, называемый защелкивающий (catch, destination, latch), триггер FF2 защелкивает данные на входе D и передает их на выход Q и далее по схеме.

Чтобы данные корректно распространялись описанным выше образом от триггера к триггеру при заданной частоте тактового сигнала задержки должны быть выполнены два условия:

* данные должны распространяться достаточно быстро, чтобы успеть дойти до триггера до прихода защелкивающего фронта (максимальное время распространения);
* данные должны распространяться достаточно медленно, чтобы предыдущие данные успели быть получены защелкивающим фронтом (минимальное время распространения).

Задача статический временного анализа заключается в том, чтобы для каждого пути между двумя триггерам рассчитать задержки распространения данных и тактового сигнала и установить удовлетворяются ли два приведенных выше ограничения.

Рассмотрим каким образом временной анализатор проводит расчет задержек. На рис. 2 представлен путь сигнала, на который нанесены задержки для данных и тактового сигнала.

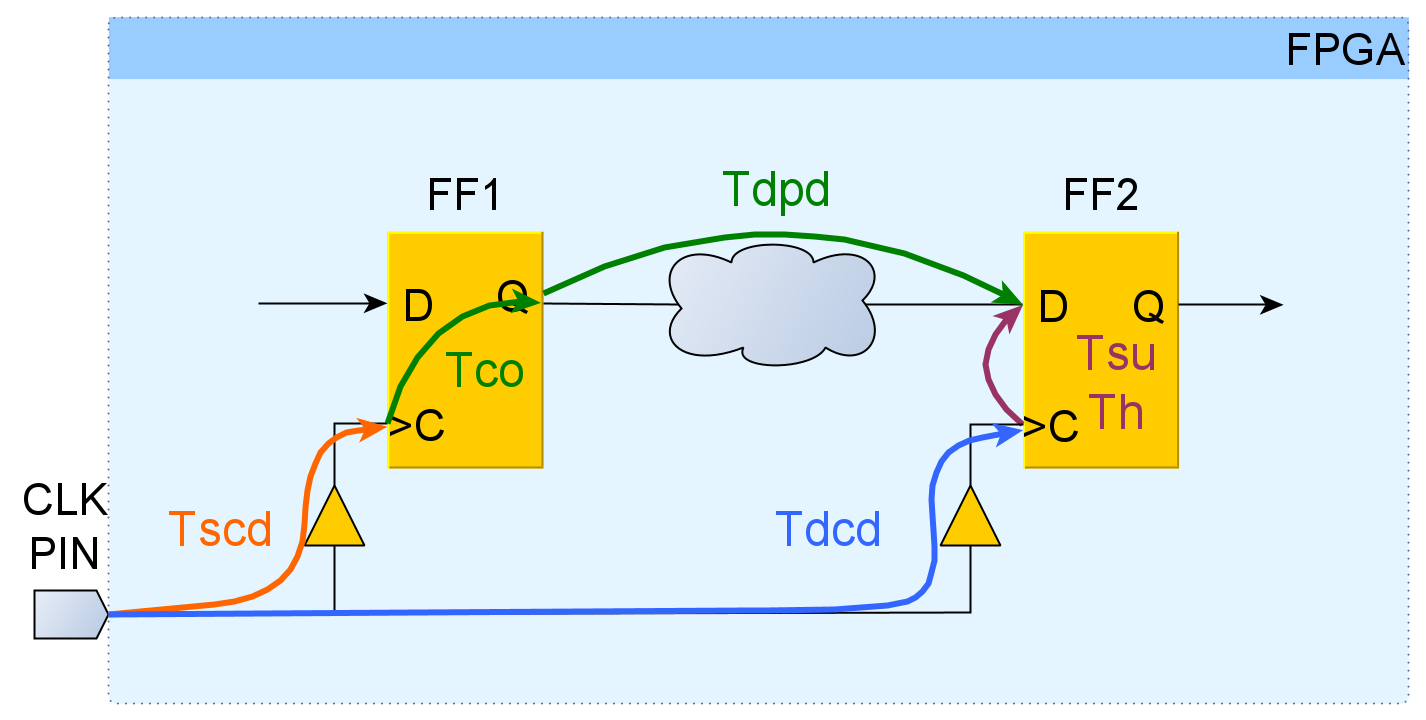


Рисунок 2. Задержки при распространении данных и тактового сигнала.

Ниже представлены определения задержек, представленных на рис.2.

* (source clock delay) – задержка тактового сигнала от тактового пина до тактового входа триггера FF1;
* (destination clock delay) – задержка тактового сигнала от тактового пина до тактового входа триггера FF2;
* (clock to output) – задержка от прихода фронта на тактовый вход триггера до появления данных c входа D на выходе Q;
* (data propagation delay) – задержка распространения данных между триггерами по соединениям и через логику;
* (setup time) – время установления, то есть промежуток времени до фронта тактового сигнала, в течении которого данные должны быть стабильны на входе D, чтобы надежно защелкнуться;
* (hold time) – время удержания, то есть промежуток времени после фронта тактового сигнала, в течении которого данные должны быть стабильны на входе D, чтобы надежно защелкнуться.

Обычно все события отсчитываются от некоторого нулевого момента времени. В качестве нулевого момента возьмем запускающий фронт. Период тактового сигнала обозначим как .

**Максимальное время распространения**

Для начала рассмотрим ограничение на максимальное время распространения:

Временной анализ проводится всегда для самого пессимистичного случая. Этому случаю соответствует максимально задержаний запускающий фронт и данные и максимально быстро пришедший защелкивающий фронт.

* Время прибытия запускающего фронта (Source clock arrive time):
* Задержка распространения данных (data delay):
* Время прибытия данных на вход защелкивающего триггера (data arrive time):
* Время прибытия защелкивающего фронта (Destination clock arrive time):

Защелкивающий является следующим после запускающего, поэтому к задержке распространения добавлен один период тактового сигнала.

* Требуемое время прибытия данных (data require time):

Так как данные должны быть стабильны в течении времени установки до защелкивающего фронта, данные должны прийти на время раньше защелкивающего фронта.

Чтобы было удовлетворено требование на максимальное время распространения, данные должны попасть на D вход защелкивающего триггера не позже, чем требуется в соответствии с .

Статический анализатор рассчитывает запас () в задержке данных

Если положительный, то это значит, что данные приходят раньше, чем требуется. Отрицательный указывает на нарушение ограничения на максимальное время распространения.

Подставив рассчитанные значения задержек получим еще ряд выражений для

Величина равная разности времени распространения тактового сигнал до запускающего и защелкивающего триггеров называется расфазировкой тактового сигнала (clock skew)

.

Можно увидеть, что положительное значение увеличивает .

**Минимальное время распространения**

Теперь рассмотрим ограничение на минимальное время распространения:

Временной анализ также проводится для самого пессимистичного случая, которому соответствует наиболее быстро распространяющийся запускающий фронт и данные и максимально задержанный защелкивающий фронт.

* Время прибытия запускающего фронта (Source clock arrive time):
* Задержка распространения данных (data delay):
* Время прибытия данных на вход защелкивающего триггера (data arrive time):
* Время прибытия защелкивающего фронта (Destination clock arrive time):

Защелкивающий фронт должен успеть защелкнуть предыдущие данные, то есть он появляется в то же время что и запускающий фронт, что поэтому к задержке распространения период тактового сигнала не добавляется.

* Требуемое время прибытия данных (data require time):

Так как предыдущие данные должны быть стабильны в течении времени удержания после защелкивающего фронта, новые данные должны прийти на время позже защелкивающего фронта.

Чтобы было удовлетворено требование на минимальное время распространения, данные должны попасть на D вход защелкивающего триггера не раньше, чем требуется в соответствии с .

Статический анализатор рассчитывает в задержке данных

Если положительный, то это значит, что данные приходят позже, чем требуется. Отрицательный указывает на нарушение ограничения на минимальное время распространения.

Подставив рассчитанные значения задержек получим еще ряд выражений для

Можно увидеть, что в выражение для расфазировкой тактового сигнала входит с знаком противоположным, тому который был при расчете запаса на максимальное время распространения. То есть положительное значение уменьшает .

**Задание ограничений и разбор отчетов в Vivado.**

Рассмотрим задание ограничений в Vivado. После синтеза для каждого пути Vivado известно через какие логические элементы он проходит (LUT, MUX, CARRY CHAIN). Типовые задержки указываются в Datasheet для конкретного кристалла. Например для в Datasheet для Artix 7 (ds 181) в таблице 27 указано время распространения через LUT (input, logic, output), а также время установления и удержания для триггеров и . В таблицах 32 – 35 указываются задержки распространения тактового сигнала через различные виды буферов.

После размещения и разводки проекта Vivado также получает информацию о задержках сигналов при распространении через линии соединения. После имплементации Vivado знает значения все переменных, которые входят в уравнения (1) и (2), кроме периода тактового сигнала . В нашем случае тактовый сигнал поступает от внешнего генератора, частоту которого Vivado, естественно, не знает.

Для примера рассмотрим самый простой пример. Пусть проект состоит из двух триггеров между которыми расположен LUT выполняющий логическое отрицание. Пример описания на System Verilog

module top (

    input  logic clk\_pin,

    input  logic in1,

    output logic out1

);

    logic ff1, ff2;

    always\_ff @(posedge clk\_pin)

        ff1 <= in1;

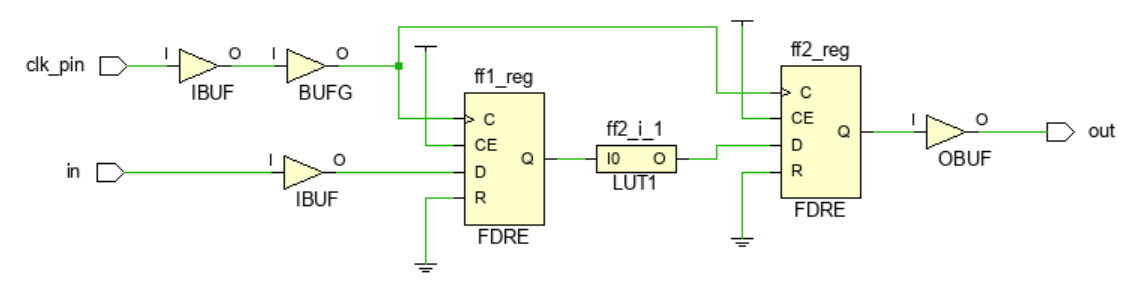
    always\_ff @(posedge clk\_pin)

        ff2 <= ~ff1;

    assign out1 = ff2;

endmodule

Схема проекта, полученная после имплементации представлена рисунке 3.

Рисунок 3. Схема проекта

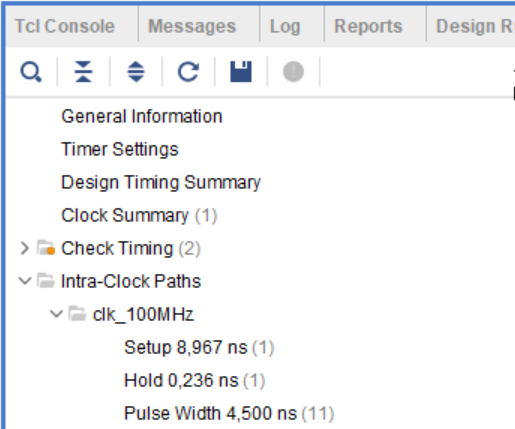
Чтобы Vivado мог провести временной анализ и проверить проект на удовлетворение максимальное и минимальное время распространения ему необходимо указать период тактового сигнала. Это делается в файле ограничений формата XDC. Будем считать, что частота тактового сигнала равна 100 МГц.

В xdc-файле необходимо указать следующую команду

create\_clock -period 10.000 -name clk\_100MHz [get\_ports clk\_pin]

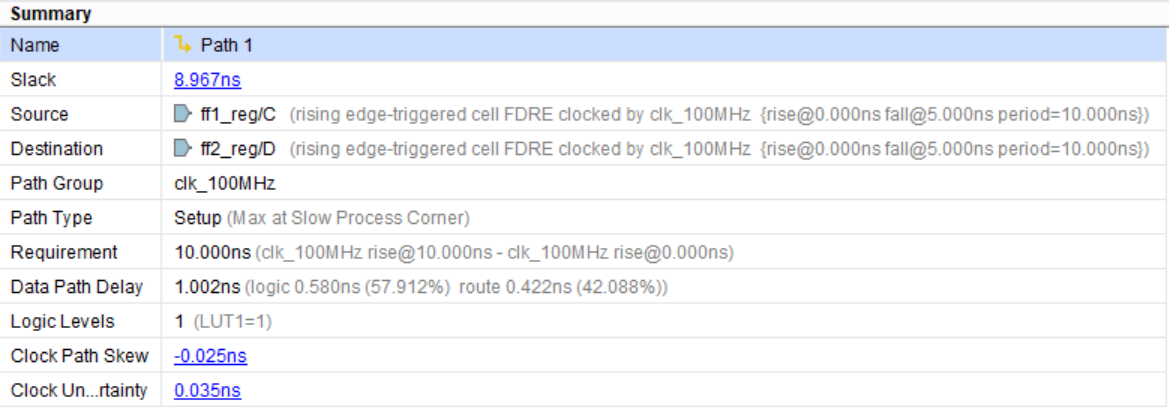
Опция –period указывает период тактового сигнала в наносекундах. Конструкция [get\_ports clk\_pin] возвращает порт проекта с именем clk\_pin, указывает откуда приходит тактовый сигнал. С помощью опции –name можно задать имя тактового сигнала. Если этого не сделать имя тактового сигнала будет совпадать с именем порта.

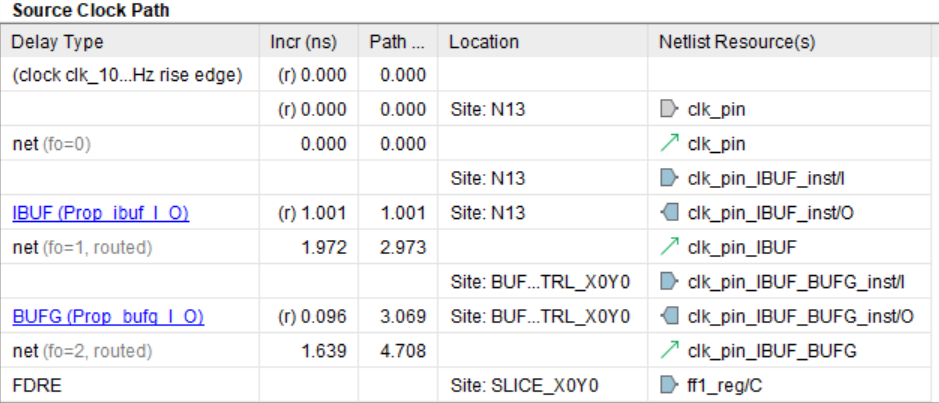
После имаплементации открыть Timing Summury, во вкладке Intra-Clock Path clk\_100MHz будет представлены некоторые пути, для которых проводился временной анализ.

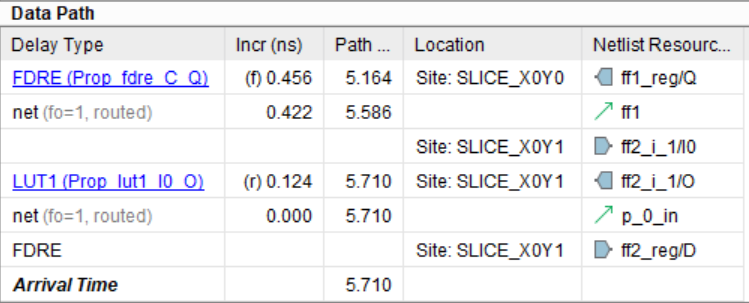


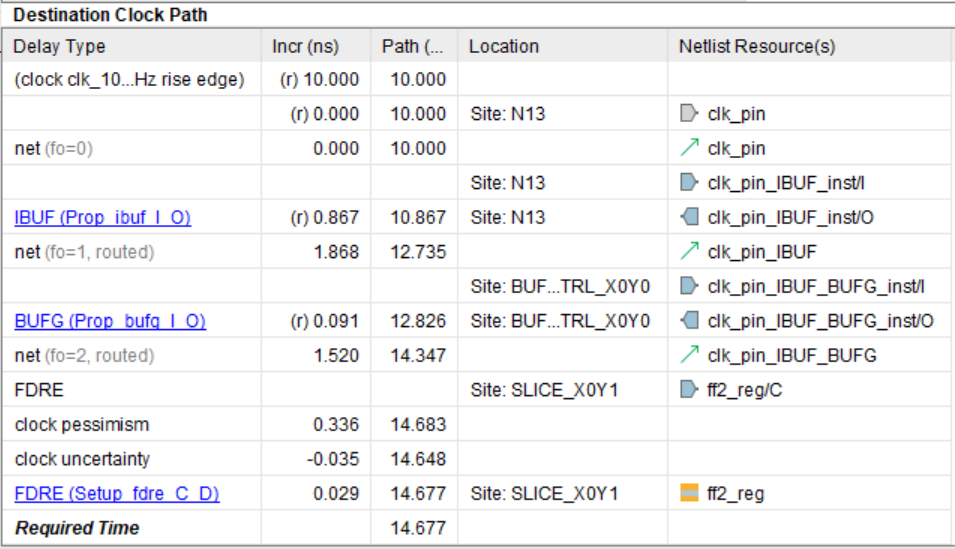
Раздел Setup соответствует проверке на максимальное время распространения, а раздел Hold на минимальное время распространения.

Для начала рассмотри отчет максимальное время распространения. В разделе Setup дважды нажав на показанный путь можно открыть расширенный отчет path report. Данный отчет состоит из четырех разделов. Рассмотрим их по порядку.









**Неопределенность и устранение пессимизма задержки.**

set\_system\_jitter, set\_input\_jitter, CPPR

**Заключение.**

Что было в статье. Что будет дальше.