## Анализ разработанного модуля

## Анализ критических путей в дизайне

Данный модуль является комбинационным, поэтому для анализа критических путей необходимо было добавить в проект файл, который будет содержать дополнительные параметры для его временного анализа. Был добавлен файл timing.xdc который содержал следующие строки:

- create\_clock -name VCLK -period 8.0 -waveform {0 4.0} создание виртуальных часов для подсчета временных задержек блока, с именем VCLK, пириодом в 8 нс, переход с единицы на ноль происходит в середине периода.
- set\_input\_delay 1.0 -clock [get\_clocks VCLK] [all\_inputs] для всех входных портов новый сигнал поступает через 1нс после VCLK нарастания.
- set\_output\_delay 1.0 -clock [get\_clocks VCLK] [all\_outputs] все выходные сигналы должны быть установлены за 1нс до VCLK нарастания.

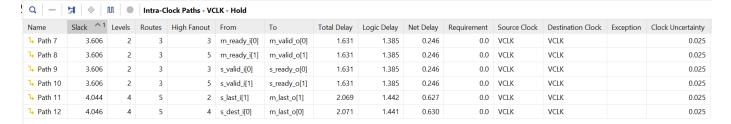
При применении данных параметров я получил такой результат



## Setup



Hold



Стоит упомянуть, что данный синтез, был произведен для кроссбара из примера, а именно с двумя master, slave и разрядностью данных в 8 бит, соответственно для других реализация время будет различаться.

Получается при данной реализации тактовая частота может быть около 125 Мгц, что не сильно быстроё. Как видно по табличке первые два пути в setup являются критическими и сильно отличаются от следующих. Данные пути — это выходные сигналы m\_last\_o, они информируют приемник о том, что данный пакет является последним. Изменив путь для этих сигналов, мы сможем заметно ускорить модуль.

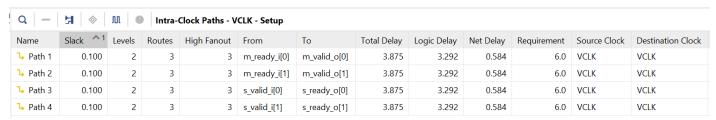
По логике работы данные сигналы формируется в модуле round-robin. После полного опроса арбитражем всех запросов, модуль начинает вычислять значение сигнала m\_last\_o, хотя это совсем не обязательно. Данный сигнал может быть вычислен сразу после нахождения следующего отправителя. Так, пришлось изменить модуль:

```
(m_last_o) begin
if(m_last_o) begin
                                                                                 m_last_o = 0;
    if(state == S DATA COUNT - 1) begin
                                                                                 if(state == S_DATA_COUNT - 1) begin
       state = 0;
                                                                                     state = 0;
    end else begin
                                                                                  nd else begir
       state = state + 1;
                                                                                     state = state + 1;
for (i = 0; i < S_DATA_COUNT; i = i + 1) begin
    if (request_mask_i[(state + i) % S_DATA_COUNT] && !first_one) begin
                                                                             for (i = 0; i < S_DATA_COUNT; i = i + 1) begin
       grant_o[(state + i) % S_DATA_COUNT] = 1'b1;
                                                                                 if (request_mask_i[(state + i) % S_DATA_COUNT] && !first_one) begin
       first_one = 1;
       state = (state + i) % S_DATA_COUNT;
                                                                                     grant_o[(state + i) % S_DATA_COUNT] = 1'b1;
                                                                                     first_one = 1;
       grant_o[(state + i) % S_DATA_COUNT] = 1'b0;
                                                                                     if (s_last_i[(state + i) % S_DATA_COUNT]) m_last_o = 1;
                                                                                     state = (state + i) % S_DATA_COUNT;
                                                                                 end else begir
                                                                                     grant_o[(state + i) % S_DATA_COUNT] = 1'b0;
if (s_last_i[state]) begin
   m_last_o = 1;
   else m_last_o = 0;
```

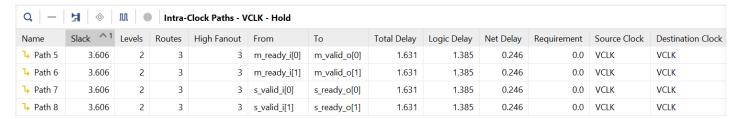
Изменив путь для этого сигнала, я смог повысить тактовую частоту до 166 Мгц, при учете input, output delay = 1нс. Если считать частоту без учета данных задержек, то финальная частота модуля равна 250 Мгц.

Setup		Hold	
Worst Negative Slack (WNS):	0,100 ns	Worst Hold Slack (WHS):	3,606 ns
Total Negative Slack (TNS):	0,000 ns	Total Hold Slack (THS):	0,000 ns

## Setup



Hold



Рекомендации по изменению дизайна, которые могут повысить частоту и/или пропускную способность

Кроме уже предложенного ранее решения для повышения частоты данного модуля можно применить технологию конвейера.

Благодаря конвейеризации (временной параллелизм) можно увеличивает скорость работы схемы без сильного увеличения аппаратных затрат. Разделив данную схему на n участков, при условии, что временная задержка на всех участках одинаковая, мы сможем сохранить прежнюю латентность системы при серьезном повышении пропускной способности.