Constraining Input and Outputs – using SDC

- Mitu Raj, chip@chipmunklogic.com, Mar-2026

Prologue

Some of you may have already read my white paper on constraining clocks in RTL designs. In this white paper, we extend that discussion to constraining IO paths using SDC**.** We will discuss the IO constraints for the most commonly encountered interfaces, **system-synchronous** and **source-synchronous IO interfaces**, walking through a few practical design examples.

Timing paths summary

A typical clocked design module contains the following categories of timing paths:

- **reg2reg paths** – path between any 2 sequential elements. These were discussed in my previous white paper.
- **in2reg paths** - paths from input ports to internal registers.
- **reg2out paths** – paths from internal registers to output ports.
- **in2out paths** - combinatorial paths from inputs directly to outputs.

This white paper focuses on in2reg and reg2out paths.

Why constraining IO paths?

The goal of constraining in2reg and reg2out paths is to provide the STA tool with the **external timing information** that lies beyond the design boundary and is therefore unknown to the tool. This enables accurate setup and hold analysis at the endpoint registers. In the absence of these constraints, STA cannot determine when input data arrives at the design boundary or when output data is expected to be captured by external logic.

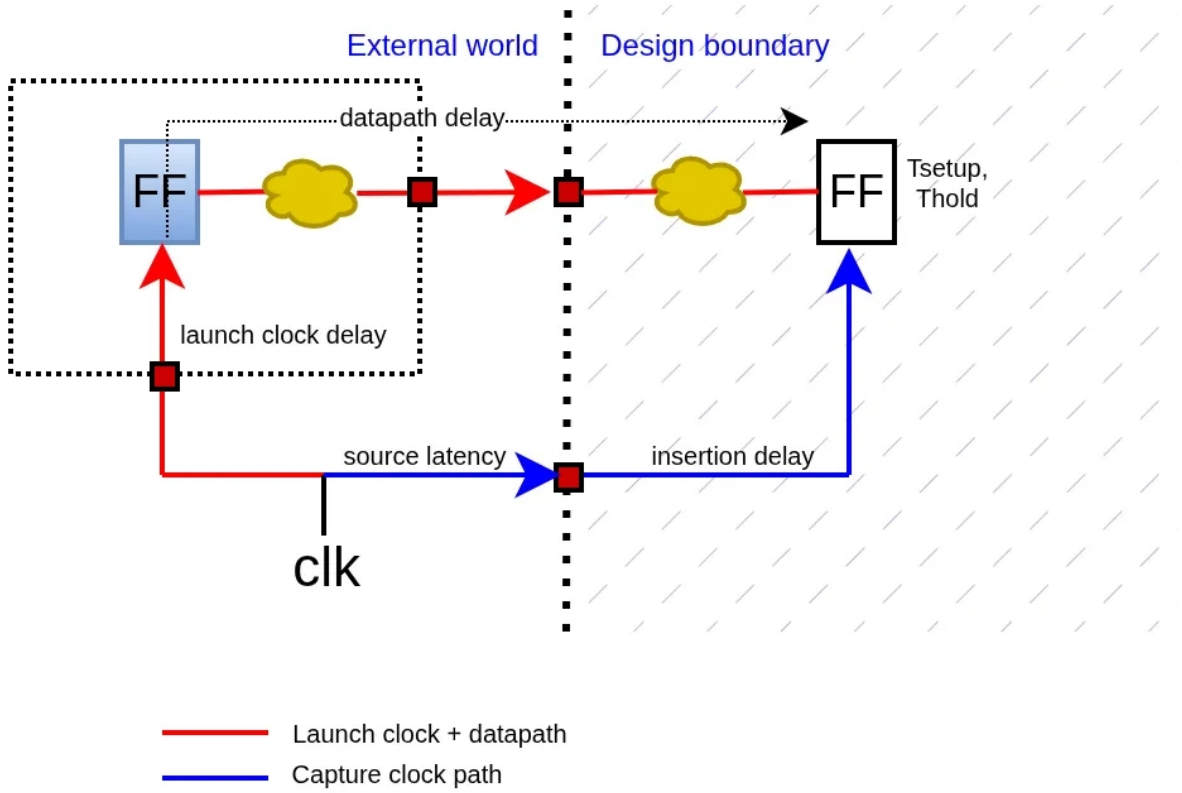
Designers usually use a **virtual clock**, a reference clock outside the design boundary to associate input/output paths with. This allows in2reg/out2reg paths to be analysed analogous to reg2reg paths. Virtual clocks allow designers to independently set specific interface delays and other constraints without factoring in the network latency and other constraints applied to the internal clock.

What constitute input and output delays?

As discussed earlier, input and output delays provide the missing external timing information required by the STA tool to accurately evaluate setup and hold relationships. Based on the input and output constraints provided, the STA tool analyzes and optimizes the internal clock and data paths to ensure that setup and hold requirements are met.

In2reg paths

In an in2reg path, the data is launched from an external device and the capturing flop is within the design.



Setup requirement

For an in2reg path-

$$\text{Launch clock path delay} + \text{Datapath delay} + T_{setup} \leq T_{clk} + \text{Capture clock path delay}$$

Clock path delay is the sum of the **source latency** and **network latency** (or **insertion delay**) of the clock.

Source latency is the latency of the clock from the source/origin to the clock definition point of the design. This is also an external timing information.

Insertion delay is the latency from the clock definition point to the clock pin of the sequential elements, like registers.

$$T_{\text{launch\_clk\_srclat}} + T_{\text{launch\_clk\_insdly}} + \text{Datapath delay} + T_{setup} \leq T_{clk} + T_{\text{capt\_clk\_srclat}} + T_{\text{capt\_clk\_insdly}} \text{---(1)}$$

Let us demarcate the clock and datapath delays to internal delays (within the design boundary and known to the STA tool) and external delays (unknown)-

$$T_{\text{launch\_clk\_srclat}} + T_{\text{launch\_clk\_insdly}} + T_{\text{data\_ext}} + T_{\text{data\_int}} + T_{setup} \leq T_{clk} + T_{\text{capt\_clk\_srclat}} + T_{\text{capt\_clk\_insdly}}$$

Therefore, the setup relation becomes-

$$\boxed{\text{Input delay} + T_{\text{data\_int}} + T_{setup} \leq T_{clk} + T_{\text{capt\_clk\_insdly}} \text{---(2)}}$$

where

Input delay = (Launch clock delay - Capture clock source latency) + External datapath delay

Max value of the above expression is the **max input delay constraint** which must be provided to the STA tool.

Hold requirement

For an in2reg path-

$$\text{Launch clock path delay} + \text{Datapath delay} \geq T_{hold} + \text{Capture clock path delay}$$

This can be expanded similar to Eq(1)-

$$T_{launch\_clk\_srclat} + T_{launch\_clk\_insdly} + \text{Datapath delay} \geq T_{hold} + T_{capt\_clk\_srclat} + T_{capt\_clk\_insdly} \text{---(3)}$$

Let us demarcate the clock and datapath delays to internal delays (within the design boundary and known to the STA tool) and external delays (unknown)-

$$T_{launch\_clk\_srclat} + T_{launch\_clk\_insdly} + T_{data\_ext} + T_{data\_int} \geq T_{hold} + T_{capt\_clk\_srclat} + T_{capt\_clk\_insdly}$$

Therefore, the hold relation becomes-

$$\text{Input delay} + T_{data\_int} \geq T_{hold} + T_{capt\_clk\_insdly} \text{---(4)}$$

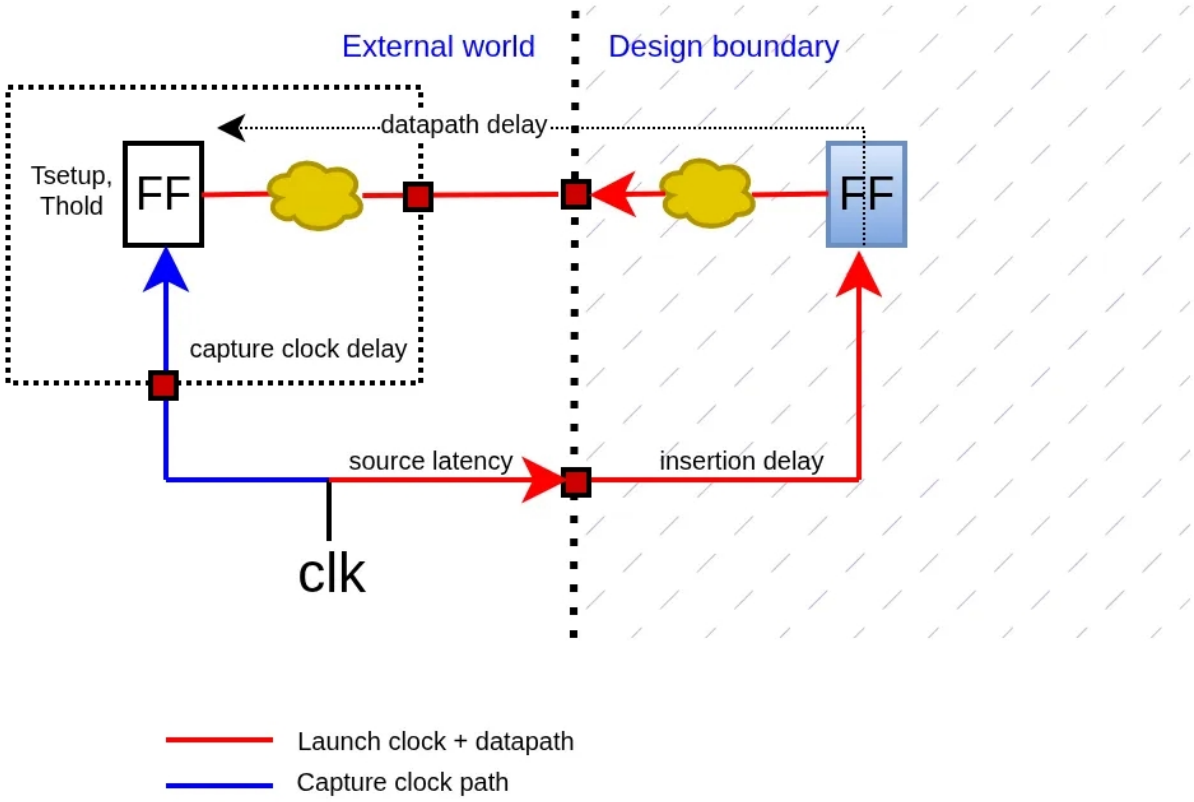
where

Input delay = (Launch clock delay - Capture clock source latency) + External datapath delay

Min value of the above expression is the **min input delay constraint** which must be provided to the STA tool.

reg2out paths

In a reg2out path, the data is launched from a flop in the design and it is captured by an external device.



Setup requirement

The requirement is the same as Eq(1). Let us demarcate the clock and datapath delays to internal delays (within design boundary and known to the STA tool) and external delays (unknown)-

$$T_{\text{launch\_clk\_srclat}} + T_{\text{launch\_clk\_insdly}} + T_{\text{data\_int}} + T_{\text{data\_ext}} + T_{setup} \leq T_{clk} + T_{\text{capt\_clk\_srclat}} + T_{\text{capt\_clk\_insdly}}$$

Therefore, the setup relation becomes-

$$T_{\text{launch\_clk\_insdly}} + T_{\text{data\_int}} + \text{Output delay} \leq T_{clk} \text{ ---(5)}$$

where

Output delay = (Launch clock source latency - Capture clock source latency) + External setup time

where

$$\text{External setup time} = T_{setup} + \left(T_{\text{data\_ext}} - T_{\text{capt\_clk\_insdly}}\right)$$

Max value of the above expression is the **max output delay constraint** which must be provided to the STA tool.

Notes

1. ***External setup time*** *is the effective setup requirement at the receiver, accounting for external clock and data path delays. Imagine the setup time of a flop is 5 ns. The reference points to this setup time are the D and the CLK pins of the flop. Suppose, a delay of 2 ns is added on the D-input data path and 1 ns on clock path of the flop and*

*the flop is blackboxed as a receiver with external input and clock pins. The effective setup time as referenced from the Input pin to Clock pin of the receiver, becomes 5+(2-1) = 6 ns. This means that, at the receiver boundary, the input data must arrive **6 ns before the clock edge** to satisfy the internal setup requirement (5 ns) of the flop, because the data is slower than the clock by 1 ns inside the receiver.*

Hold requirement

The requirement is the same as Eq(3). Let us demarcate the clock and datapath delays to internal delays (within design boundary and known to the STA tool) and external delays (unknown)-

$$T_{\text{launch\_clk\_srclat}} + T_{\text{launch\_clk\_insdly}} + T_{\text{data\_int}} + T_{\text{data\_ext}} \geq T_{hold} + T_{\text{capt\_clk\_srclat}} + T_{\text{capt\_clk\_insdly}}$$

Therefore, the hold relation becomes-

$$T_{\text{launch\_clk\_insdly}} + T_{\text{data\_int}} + \text{Output delay} \geq 0 \quad \text{---(6)}$$

where

$$\text{Output delay} = (\text{Launch clock source latency} - \text{Capture clock source latency}) - \text{External hold time}$$

where

$$\text{External hold time} = T_{hold} + \left(T_{\text{capt\_clk\_insdly}} - T_{\text{data\_ext}}\right)$$

Min value of the above expression is the **min output delay constraint** which must be provided to the STA tool.

Notes

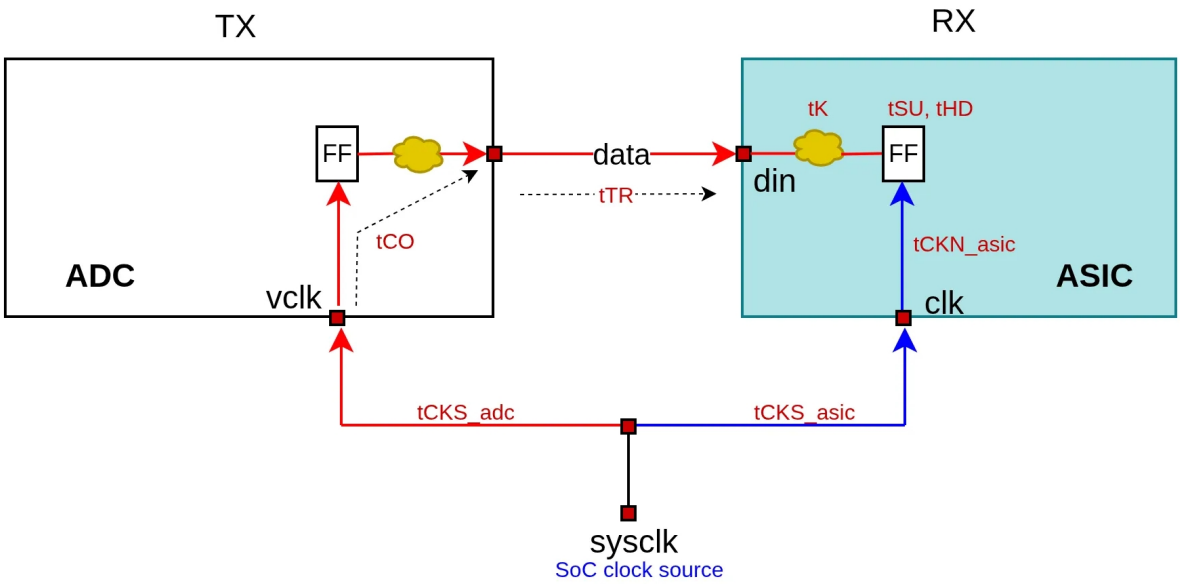
1. ***External hold time** is the effective hold requirement at the receiver, accounting for external clock and data path delays. Imagine the hold time of a flop is 5 ns. The reference points to this hold time are the D and the CLK pins of the flop. Suppose, a delay of 1 ns is added on the D-input data path and 2 ns on clock path of the flop and the flop is blackboxed as a receiver with external input and clock pins. The effective hold time as referenced from the Input pin to Clock pin of the receiver, becomes 5+(2-1) = 6 ns. This means that, at the receiver boundary, the input data must be stable for **6 ns after the clock edge** to satisfy the internal hold requirement (5 ns) of the flop, because the clock is slower than the data by 1 ns inside the receiver.*

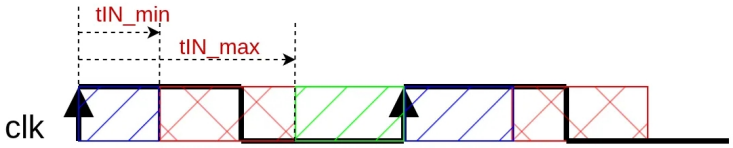
System-synchronous interfaces

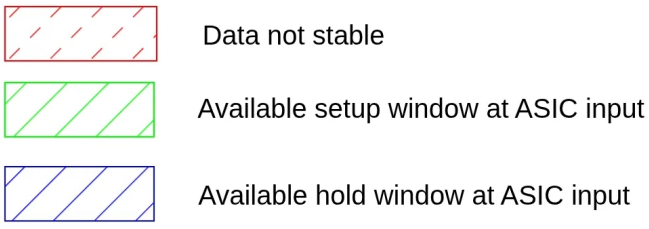
A system-synchronous interface is one in which both the transmitting and receiving devices are driven by a common reference clock.

Design example 1

In an SoC, an ASIC receives input data from an ADC. Both the ASIC and ADC are driven by the same external clock source. The propagation delay at the ADC output and routing estimates are known. Constrain the ASIC input port accordingly.







Constraints

Let-

Clock period	*tCLK*
Clock-to-output propagation delay at the ADC output from the datasheet**	*tCO*

Pin-to-Pin trace delay	tTR
Combinatorial delay at the ASIC input	tK
Setup, Hold times of the register at the ASIC input	tSU , tHD
Source latency of the clock to the ADC clock pin	$tCKS_{adc}$
Source latency of the clock to the ASIC clock pin	$tCKS_{asic}$
Network latency (insertion delay) of the clock from the ASIC clock pin	$tCKN_{asic}$

*** tCO includes the launch clock insertion delay, because the clock-to-output delay is referenced from clock pin to output pin*

Setup time requirement

$$tCKS_{adc} + tCO + tTR + tK + tSU \leq tCLK + tCKS_{asic} + tCKN_{asic}$$

Group the external timing parameters, similar to Eq(2)-

$$(tCKS_{adc} - tCKS_{asic} + tCO + tTR) + tK + tSU \leq tCLK + tCKN_{asic}$$
$$(\text{Input Delay}) + tK + tSU \leq tCLK + tCKN_{asic} \quad \text{---(7)}$$

Hold time requirement

$$tCKS_{adc} + tCO + tTR + tK \geq tHD + tCKS_{asic} + tCKN_{asic}$$

Group the external timing parameters, similar to Eq(4)-

$$(tCKS_{adc} - tCKS_{asic} + tCO + tTR) + tK \geq tHD + tCKN_{asic}$$
$$(\text{Input Delay}) + tK \geq tHD + tCKN_{asic} \quad \text{---(8)}$$

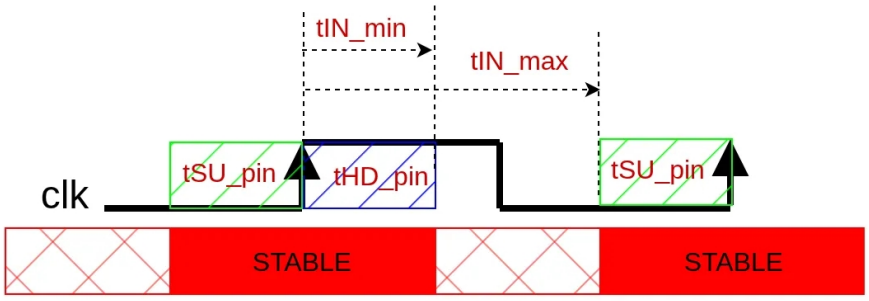
SDC

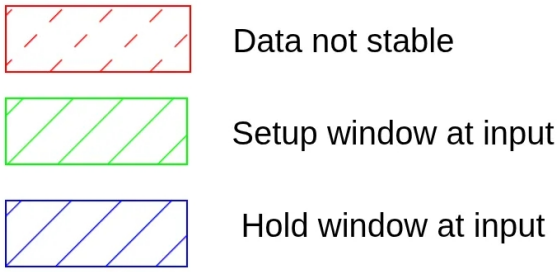
```
# Timing parameters
set tCLK <>
set tCO_min <>
set tCO_max <>
set tTR_min <>
set tTR_max <>
set tCKS_adc_min <>
set tCKS_adc_max <>
set tCKS_asic_min <>
set tCKS_asic_max <>

# Min and Max input delays
set tIN_min [expr $tCKS_adc_min - $tCKS_asic_max + $tCO_min + $tTR_min]
set tIN_max [expr $tCKS_adc_max – $tCKS_asic_min + $tCO_max + $tTR_max]

# Clock
create_clock -name clk -period $tCLK [get_ports clk]

# Virtual clock to constraint inputs
create_clock -name vclk -period $tCLK

# Setup constraint on the input
set_input_delay -clock vclk -max $tIN_max [get_ports din]
```

```
# Hold constraint on the input
set_input_delay -clock vclk -min $tIN_min [get_ports din]
```

Design example 2

The timing specifications required at the input port of an ASIC with respect to the reference clock is given in the figure below. The external timing parameters are unknown. Therefore, any external device interfaced to the ASIC is expected to meet these timing requirements. Constraint the input port at the ASIC to meet the specifications.





Constraints

Let-

Clock period	$tCLK$
Setup, Hold times of the register at the ASIC input	tSU, tHD
Setup, Hold times at the ASIC input pin**	tSU_{pin}, tHD_{pin}

*** This is similar to external setup and hold times that we discussed earlier*

Setup time requirement

From Eq(7)-

$$\text{Input Delay} + \left(tK - tCKN_{asic} + tSU\right) \leq tCLK$$
$$\text{Input Delay} + \left(tSU_{pin}\right) \leq tCLK$$
$$\text{Input Delay} \leq tCLK - tSU_{pin}$$

Hold time requirement

From Eq(8)-

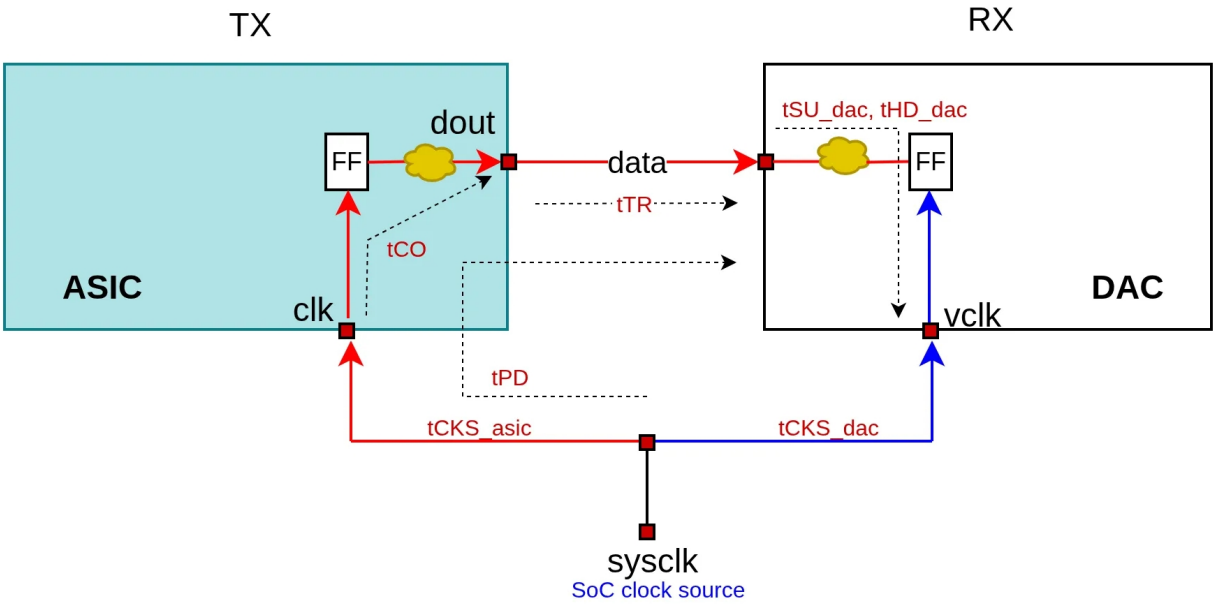
$$\text{Input Delay} \geq \left(tHD + tCKN_{asic} - tK\right)$$
$$\text{Input Delay} \geq tHD_{pin}$$

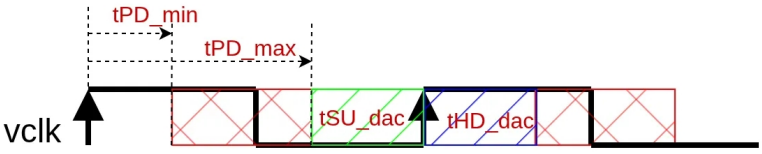
SDC

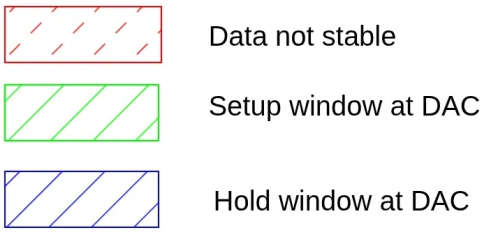
```
# Timing parameters
set tCLK <>
set tSU_pin <>
set tHD_pin <>

# Min and Max input delays
set tIN_min [expr $tHD_pin]
set tIN_max [expr $tCLK - $tSU_pin]

# Clock
create_clock -name clk -period $tCLK [get_ports clk]

# Virtual clock to constraint inputs
create_clock -name vclk -period $tCLK

# Setup constraint on the input
set_input_delay -clock vclk -max $tIN_max [get_ports din]

# Hold constraint on the input
set_input_delay -clock vclk -min $tIN_min [get_ports din]
```

Design example 3

In an SoC, an ASIC sends output data to a DAC. Both the ASIC and DAC are driven by the same external clock source. The timing requirements (setup/hold) at the DAC input port and routing estimates are known. Constrain the ASIC output port accordingly.







Constraints

Let-

Clock period	$tCLK$
Clock-to-output propagation delay at the ASIC output	tCO
Pin-to-Pin trace delay	tTR
Setup, Hold times at the DAC input from the datasheet**	tSU_{dac}, tHD_{dac}

Source latency of the clock to the DAC clock pin	$tCKS_{dac}$
Source latency of the clock to the ASIC clock pin	$tCKS_{asic}$

*** This is similar to external setup and hold times that we discussed earlier*

Setup time requirement

$$tCKS_{asic} + tCO + tTR + tSU_{dac} \leq tCLK + tCKS_{dac}$$

Group the external timing parameters, similar to Eq(5)-

$$tCO + \left(tCKS_{asic} - tCKS_{dac} + tTR + tSU_{dac}\right) \leq tCLK$$
$$tCO + \left(\text{Output Delay}\right) \leq tCLK \quad \text{---(9)}$$

Hold time requirement

$$tCKS_{asic} + tCO + tTR \geq tHD_{dac} + tCKS_{dac}$$

Group the external timing parameters, similar to Eq(6)-

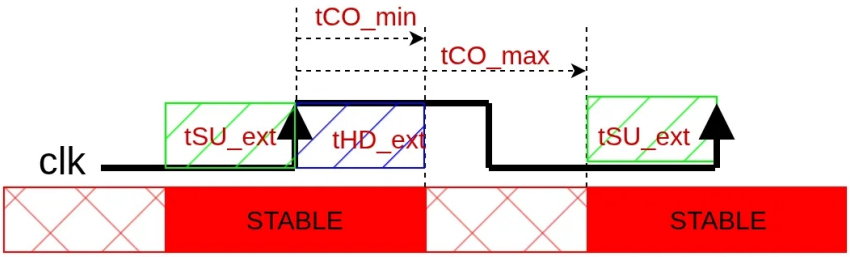
$$tCO + \left(tCKS_{asic} - tCKS_{dac} + tTR - tHD_{dac}\right) \geq 0$$
$$tCO + \left(\text{Output Delay}\right) \geq 0 \quad \text{---(10)}$$

SDC

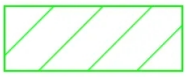
```
set tCLK <>
set tSU_dac <>
set tHD_dac <>
set tTR_min <>
set tTR_max <>
set tCKS_dac_min <>
set tCKS_dac_max <>
set tCKS_asic_min <>
set tCKS_asic_max <>

# Min and Max output delays
set tOUT_min [expr $tCKS_asic_min - $tCKS_dac_max + $tTR_min - $tHD_dac]
set tOUT_max [expr $tCKS_asic_max - $tCKS_dac_min + $tTR_max + $tSU_dac]

# Clock
create_clock -name clk -period $tCLK [get_ports clk]

# Virtual clock to constraint outputs
create_clock -name vclk -period $tCLK

# Setup constraint on the output
set_output_delay -clock vclk -max $tOUT_max [get_ports dout]

# Hold constraint on the output
set_output_delay -clock vclk -min $tOUT_min [get_ports dout]
```

Design example 4

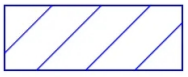
The timing specifications required at the output port of an ASIC with respect to the reference clock is given in the figure below. The external timing parameters are unknown. Any external device interfaced to the ASIC is expected to meet setup and hold with respect to these timing specifications. Constraint the output port at the ASIC to meet the specifications.



Data not stable



Available setup window at output



Available hold window at output

Constraints

Let-

Clock period	$tCLK$
Setup, Hold times available at the ASIC output	tSU_{ext}, tHD_{ext}

We can relate the external setup and hold times to tCO-

$$tSU_{ext} = tCLK - tCO_{max}$$
$$tHD_{ext} = tCO_{min}$$

Setup time requirement

From Eq(9)-

$$\text{Output Delay} \leq tCLK - tCO_{max}$$
$$\text{Output Delay} \leq tSU_{ext}$$

Hold time requirement

From Eq(10)-

$$\text{Output delay} \geq -tCO_{min}$$
$$\text{Output delay} \geq -tHD_{ext}$$

SDC

```
set tCLK <>
set tSU_ext <>
set tHD_ext <>

# Min and Max output delays
set tOUT_min [expr -1 * $tHD]
set tOUT_max [expr $tSU]

# Clock
create_clock -name clk -period $tCLK [get_ports clk]

# Virtual clock to constraint inputs
create_clock -name vclk -period $tCLK

# Setup constraint on the output
set_output_delay -clock vclk -max $tOUT_max [get_ports dout]

# Hold constraint on the output
set_output_delay -clock vclk -min $tOUT_min [get_ports dout]
```

Source-synchronous interfaces

This is quite tricky! To be updated soon … :) Stay tuned at chipmunklogic.com