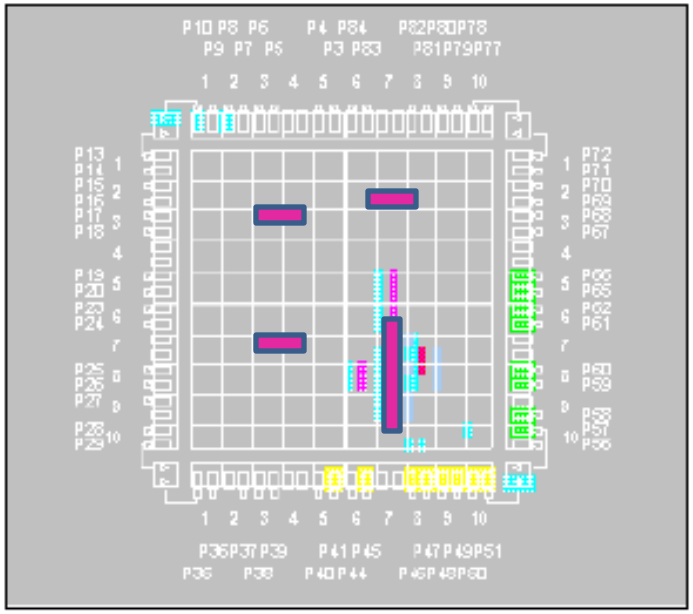
FPGAs-3

Design Without and With Constraints

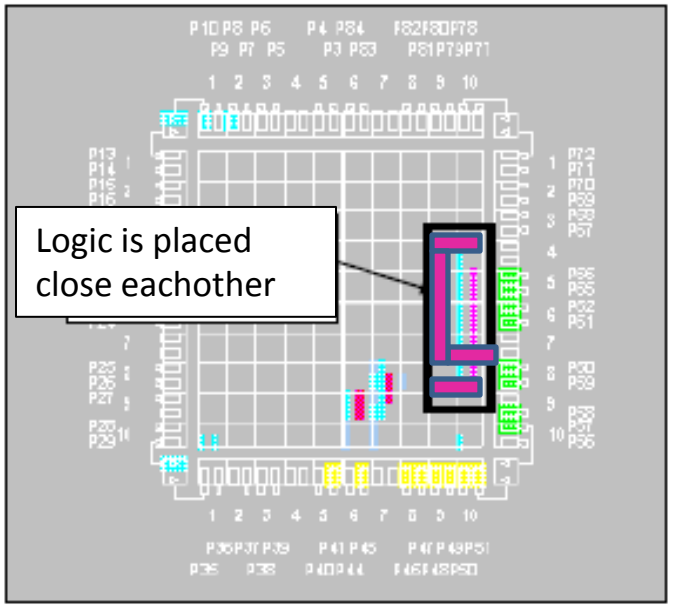
Logic Placement Can Be Very Different

Without global timing constraints



Logic is placed randomly

With global timing constraints



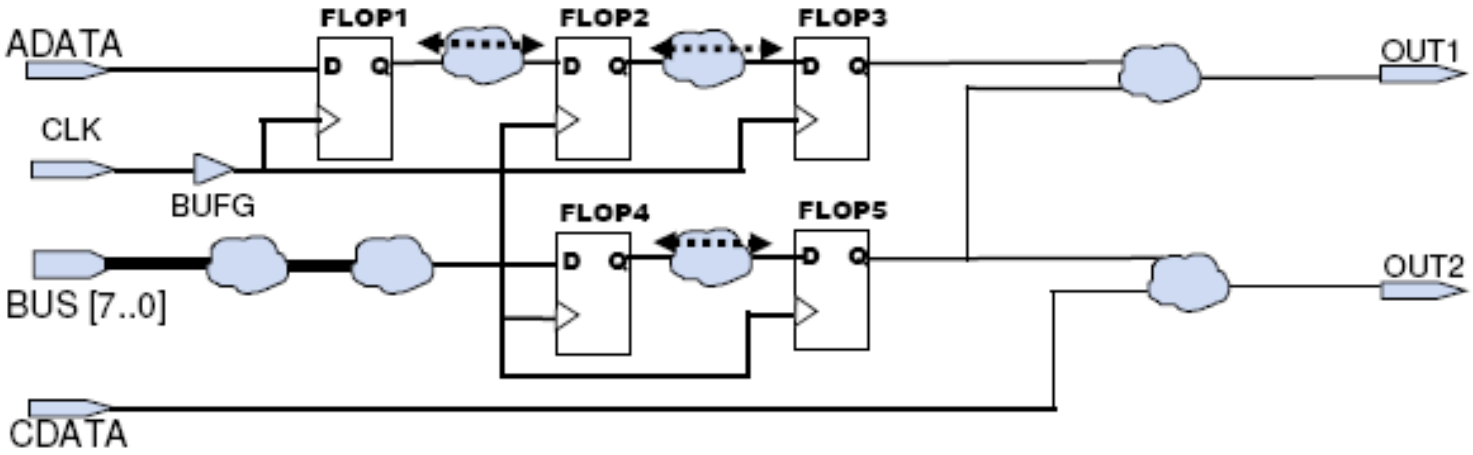
Logic is placed to result in a faster design

Setup and hold time

- ***Setup time:***The data must become valid at least a setup time before the arrival of the active clock edge at its pin.
- ***Hold Time*****:** The data must stay valid at least a hold time after the arrival of the active clock edge at its pin.

Period Constraints

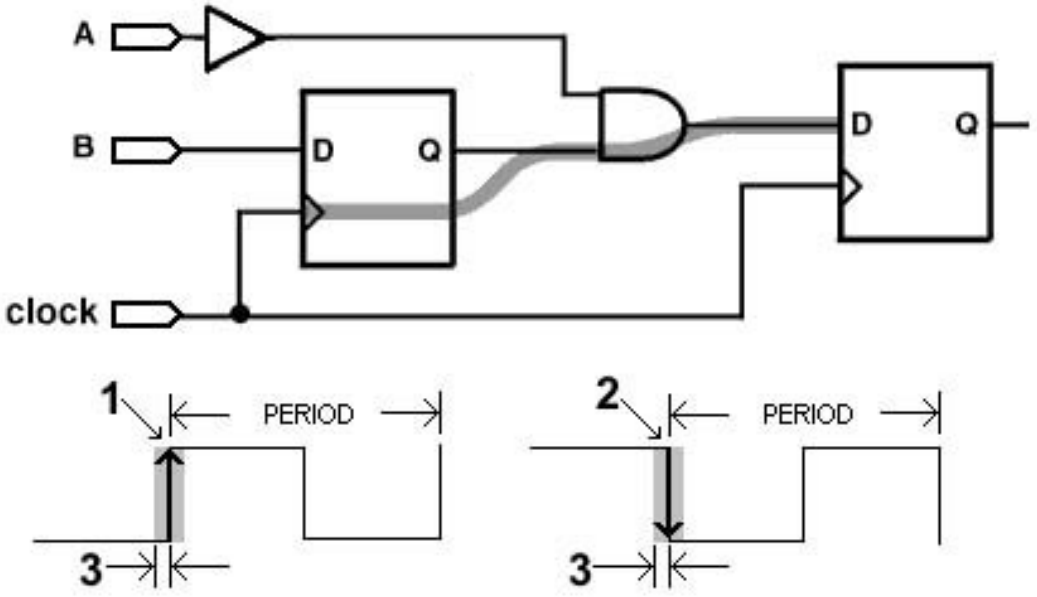
Cover Paths Between Synchronous Elements



PERIOD Constraint

Use the Most Accurate Timing Information

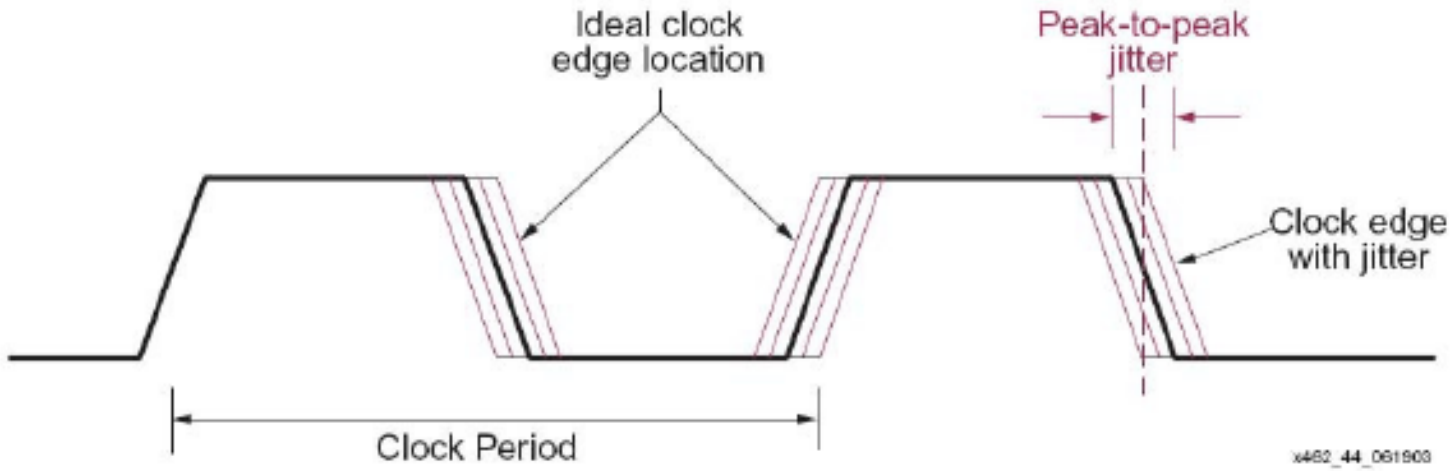
- ✓ Clock skew between the source and destination flip-flops
- ✓ Synchronous elements clocked on the negative edge
- ✓ Unequal clock duty cycles
- ✓ Clock input jitter



Period Constraint

Clock uncertainty is automatically accounted for in global constraint calculations

Clock jitter is a form of clock uncertainty



Period Constraint

Timing Analyzer calculation accounts for most accurate timing information

Timing constraint: TS_inst_clockgen_CLK0_BUF_0 = PERIOD TIMEGRP "inst_clockgen_CLK0_BUF_0" TS_clk HIGH 50% INPUT_JITTER 0.001 ns;

10 paths analyzed, 4 endpoints analyzed, 0 failing endpoints

0 timing errors detected. (0 setup errors, 0 hold errors)

Minimum period is 2.016ns.

Slack: 7.984ns (**requirement - (data path - clock path skew + uncertainty)**)

Source: bincount/BU2/U0/q_i_0 (FF)	**clk:** clkgen_out rising at 0.000ns
Destination: bincount/BU2/U0/q_i_3 (FF)	**clk:** clkgen_out rising at 10.000ns

Requirement	**Data Path Delay**	**Clock Path Skew:**	**Clock Uncertainty**
10.000ns	2.015ns (Levels of Logic = 2)	0.000ns	0.001ns

Clock Uncertainty: 0.001ns ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE	
Total System Jitter (TSJ):	0.000ns
Total Input Jitter (TIJ):	0.001ns
Discrete Jitter (DJ):	0.000ns
Phase Error (PE):	0.000ns

Maximum Data Path: bincount/BU2/U0/q_i_0 to bincount/BU2/U0/q_i_3

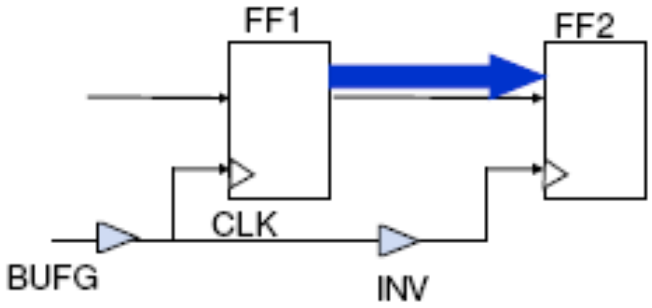
Delay type	**Delay(ns)**	**Logical Resource**
Tcko	0.370	bincount/BU2/U0/q_i_0
net (fanout=2)	0.246	dout_0_OBUF

Equation takes into account data path delay, clock skew and clock uncertainty

PERIOD Constraint

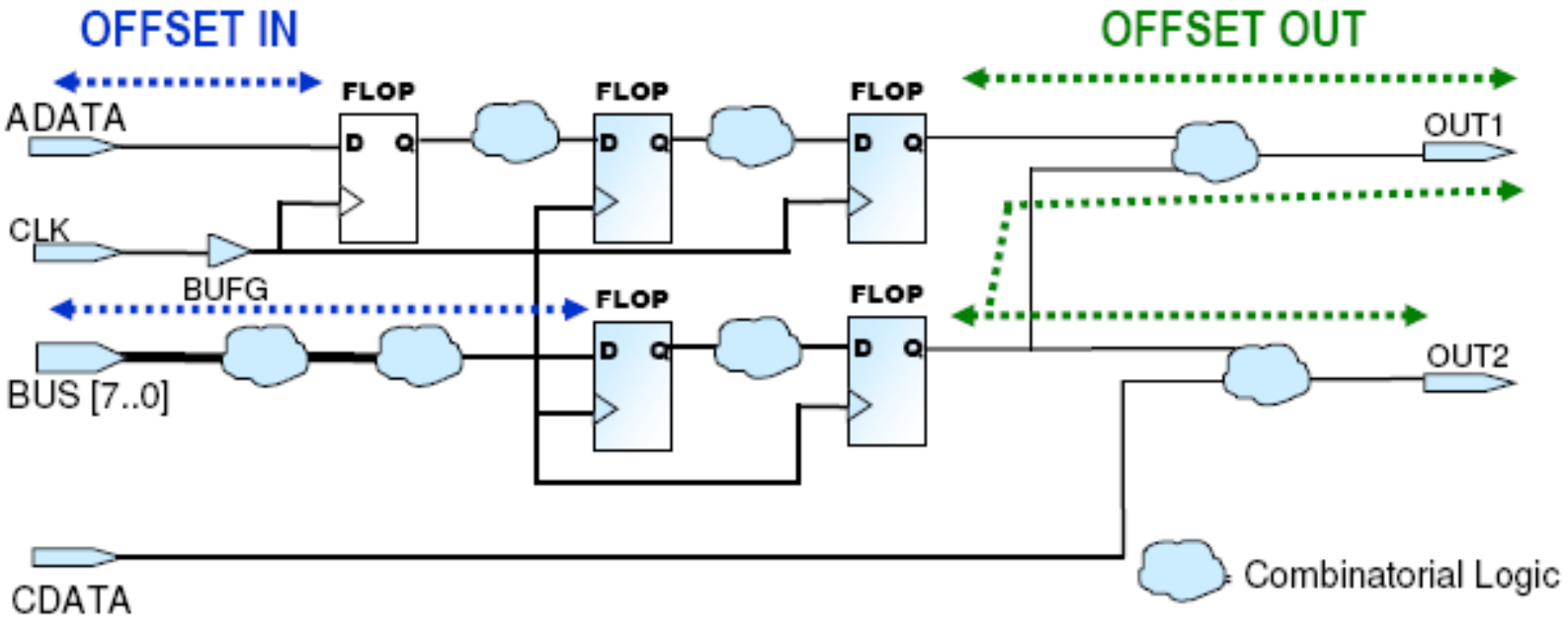
Calculation takes into account inverted clock edges

- Assume:
  - 50 percent duty cycle on CLK
  - PERIOD constraint of 10 ns
  - Because FF2 will be clocked on the falling edge of CLK, the path between the two flip-flops will be constrained to 50 percent of 10 ns = 5 ns



Offset Constraints

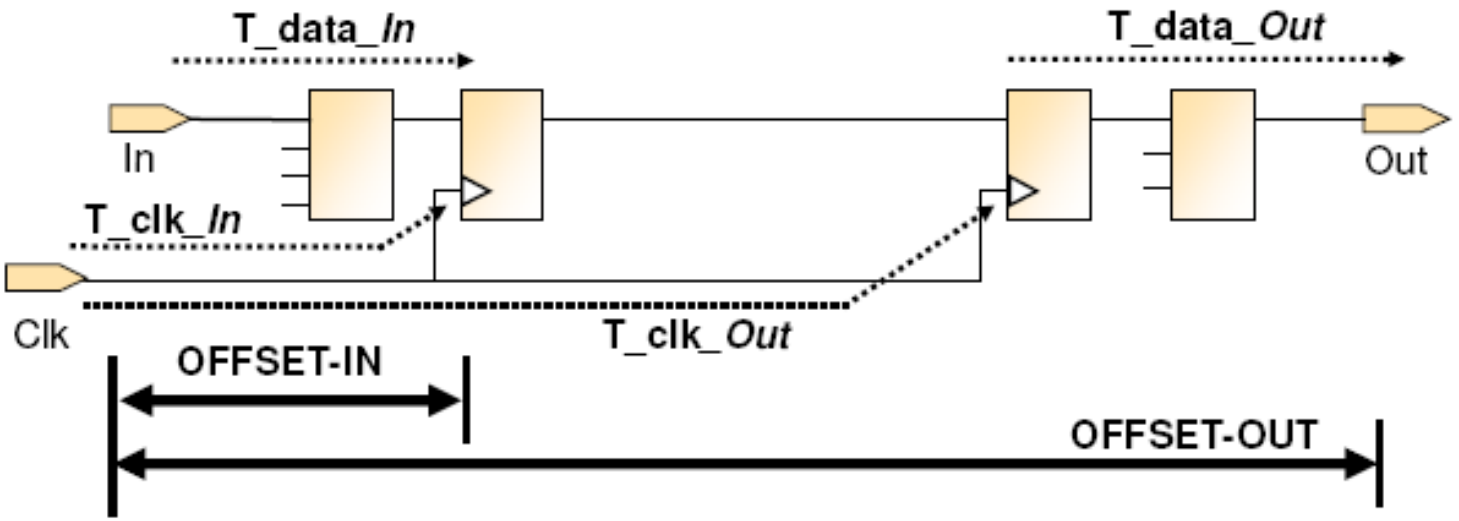
Constrains I/O Pads To/From Synchronous Elements relative to associated clock signal



Offset Constraints

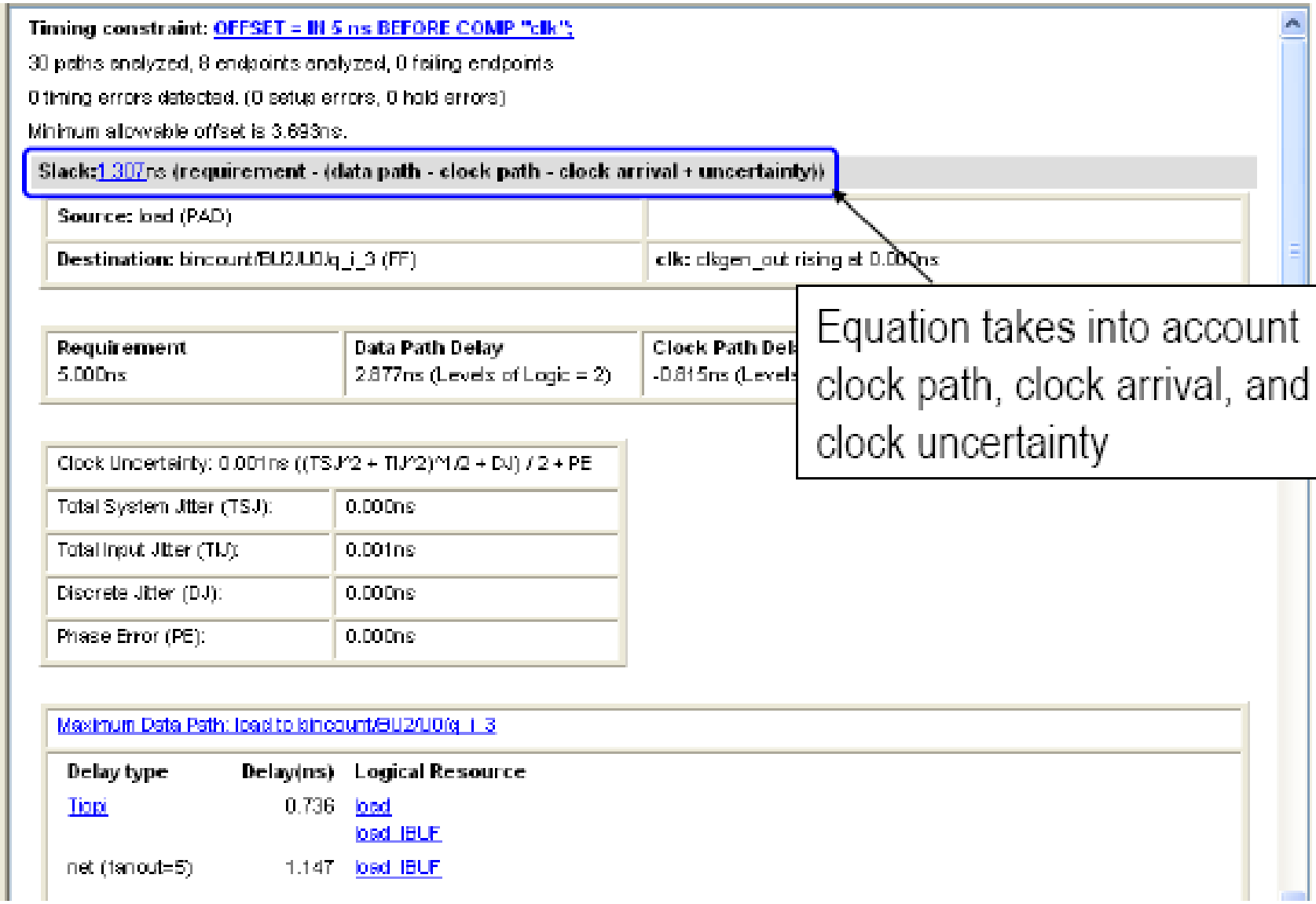
Accounts for Clock Delay

- OFFSET IN = T_data_In - T_clk_In
- OFFSET OUT = T_data_Out + T_clk_Out



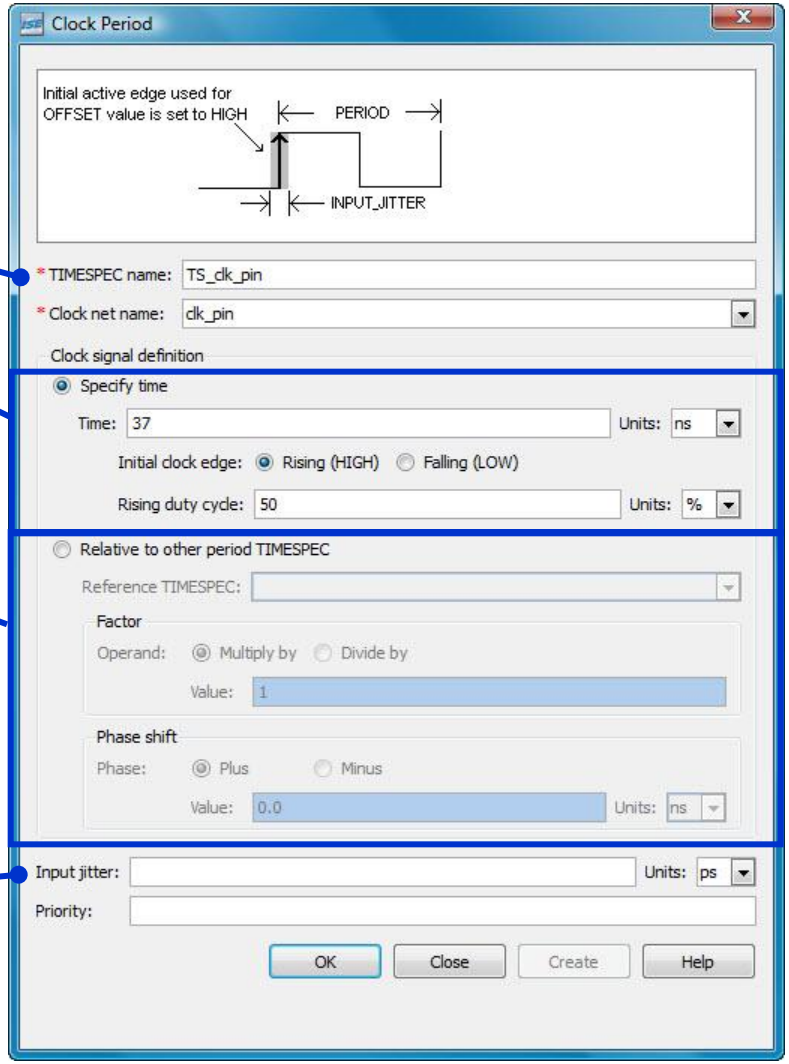
Offset Constraints

Timing Analyzer calculation accounts for most accurate timing information



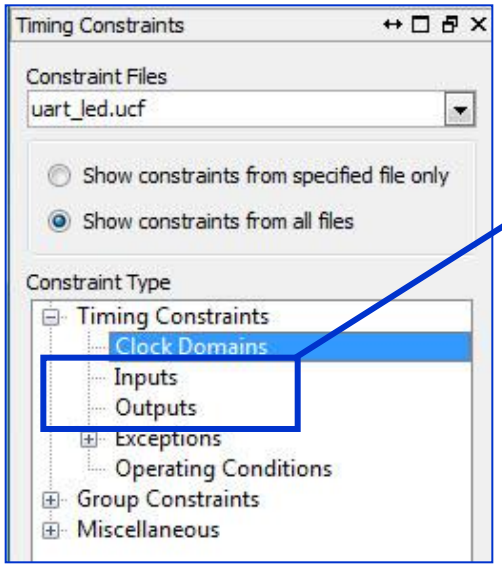
PERIOD Constraint Options

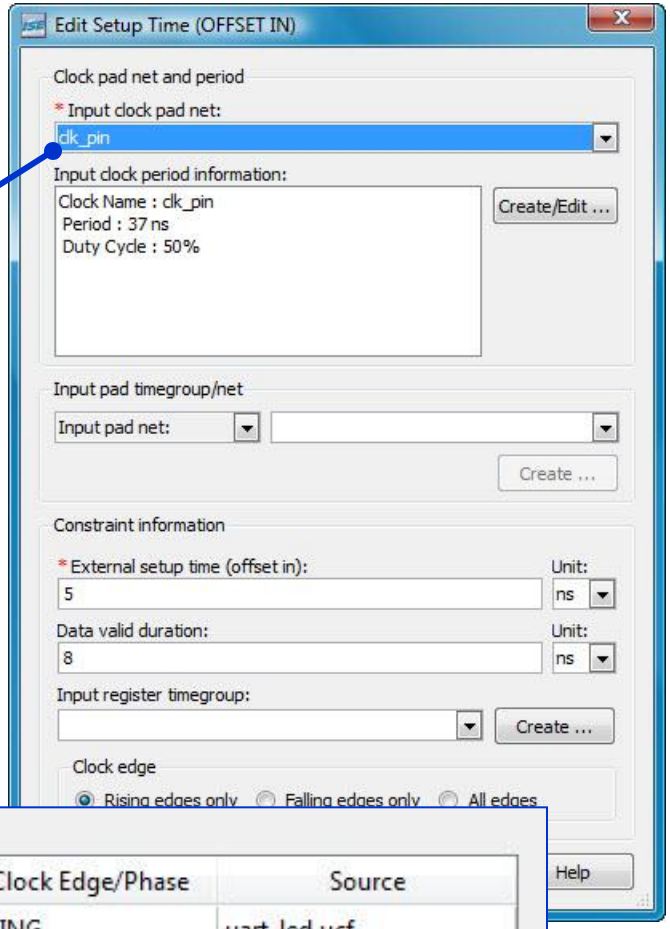
- TIMESPEC name
- Specific constraint value
  - Active clock edge
  - Duty cycle
- Relative to other PERIOD TIMESPEC
  - Useful for designs with multiple clock signals
  - Can define both frequency and phase relationships
- Input jitter

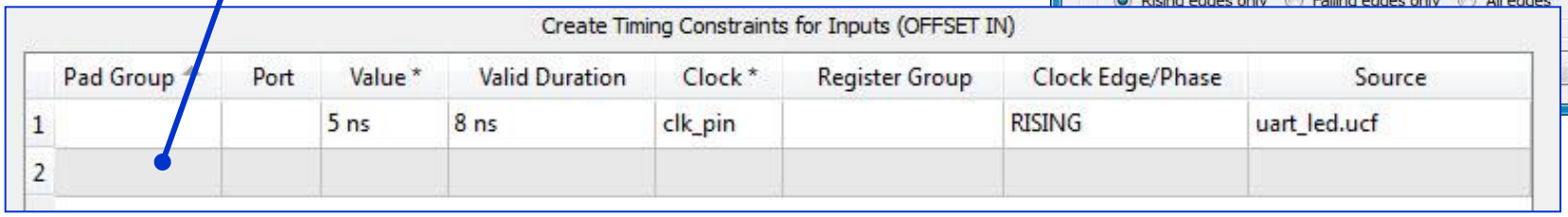


Entering OFFSET Constraints

- Global OFFSET IN and OFFSET OUT constraints can be made from Inputs or Outputs
- Right-click here and select **Create Constraint** to make an OFFSET constraint



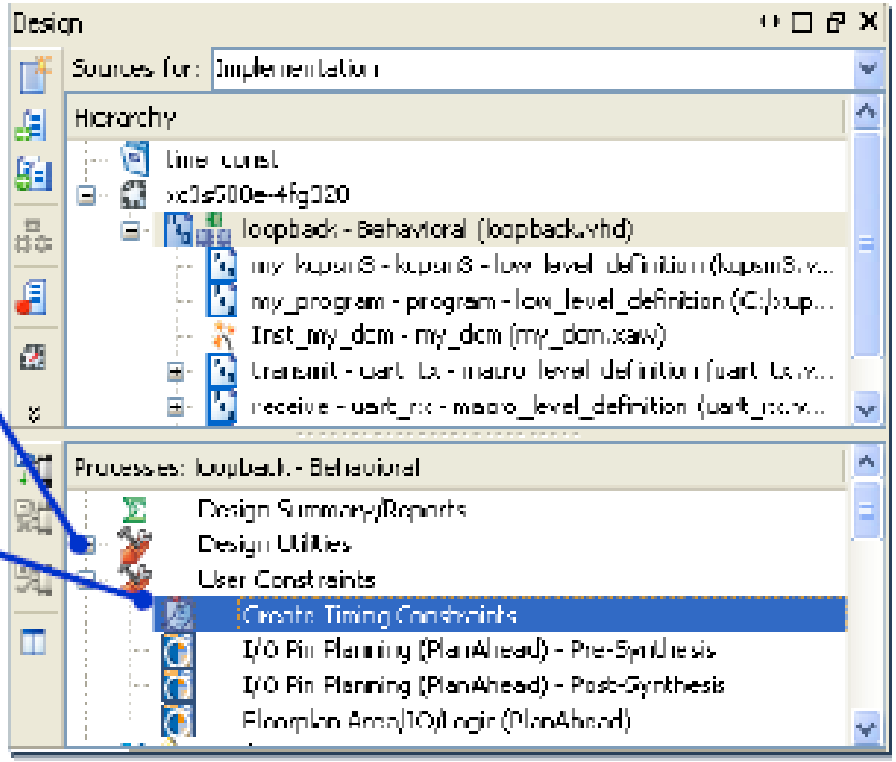




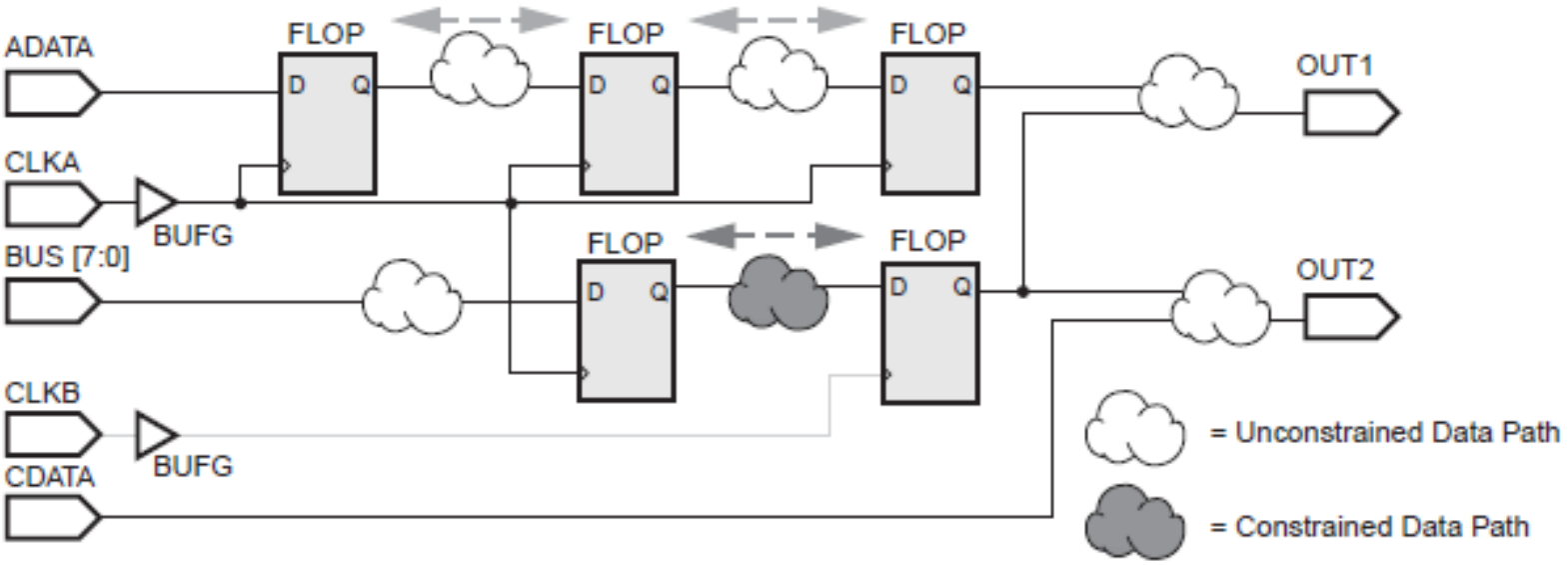
Access the Constraints Editor

Enter constraints in the Constraints Editor GUI

- Expand **User Constraints** in the Processes for Source window
- Double-click **Create Timing Constraints**



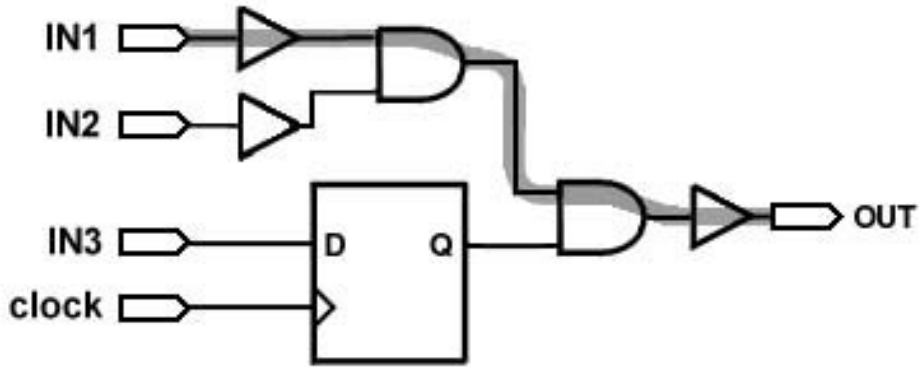
```
NET "clk" TNM_NET = clk;
TIMESPEC TS_clk = PERIOD "clk" 20 ns HIGH 50%;
OFFSET = IN 7 ns VALID 20 ns BEFORE "clk" RISING;
OFFSET = OUT 7.5 ns AFTER "clk";
```



TIMESPEC TS_clk1_to_clk2 = FROM clk1 TO clk2 8 ns;
Constrain from time group **clkA** to time group **clkB** to be 8 ns.

Pad to Pad Constraints

Covers Purely Combinatorial Paths that start and end at I/O pads



TIMESPEC TS_Pad2Pad = FROM PADS TO PADS 14.4 ns;

Latches

- Level-sensitive storage
  - Data transmitted while enable is '1'
    - *transparent* latch
  - Data stored while enable is '0'