Applying clock uncertainty for cascaded PLL and non-dedicated clock path for Arria 10® device family

If you are connecting PLL reference clock from a PLL output or a non-dedicated clock pin in your Arria® 10 design. This jitter can be compensated by adding 100ps clock uncertainty constraint at the **output clocks of the downstream PLL** in the design.

Case#1: Cascaded input from IOPLL or XCVR PLL (FPLL/ATXPLL/CDRPLL)

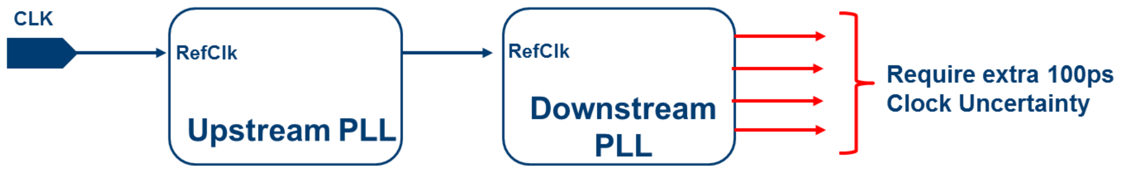


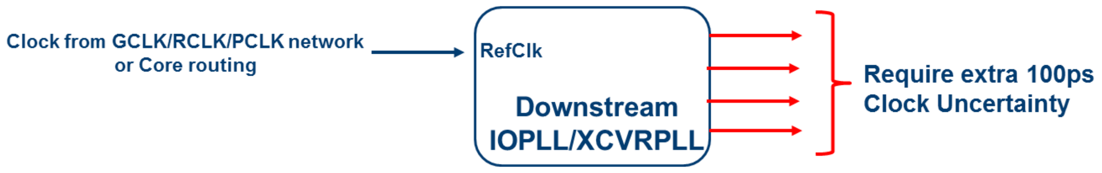
Table1: Supported and unsupported PLL cascading configuration

		Upstream PLL				
		FPLL	ATX	CDR	CMU	IOPLL
Downstream PLL	FPLL	🟥	🟥	🟥	▨	🟥
	ATX	🟥	▨	▨	▨	▨
	CDR	🟥	▨	▨	▨	🟥
	CMU	🟥	▨	▨	▨	🟥
	IOPLL	🟥	▨	🟥	▨	🟥

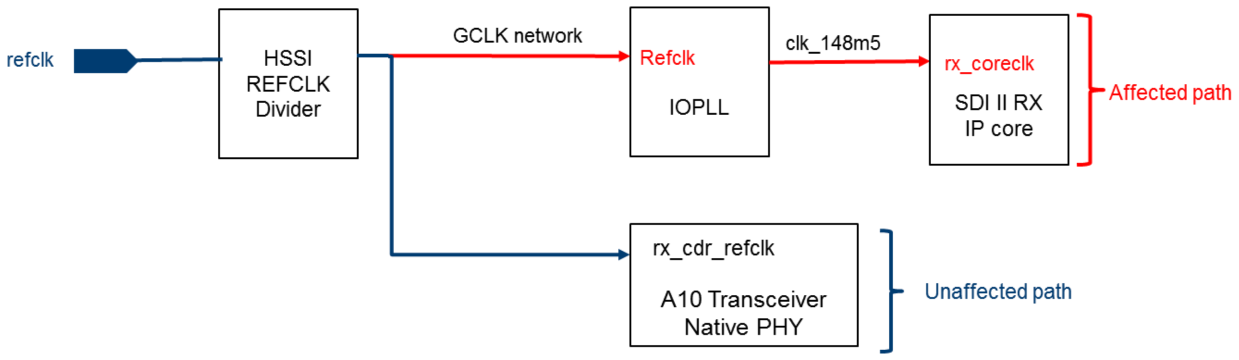
🟥 Affected cases

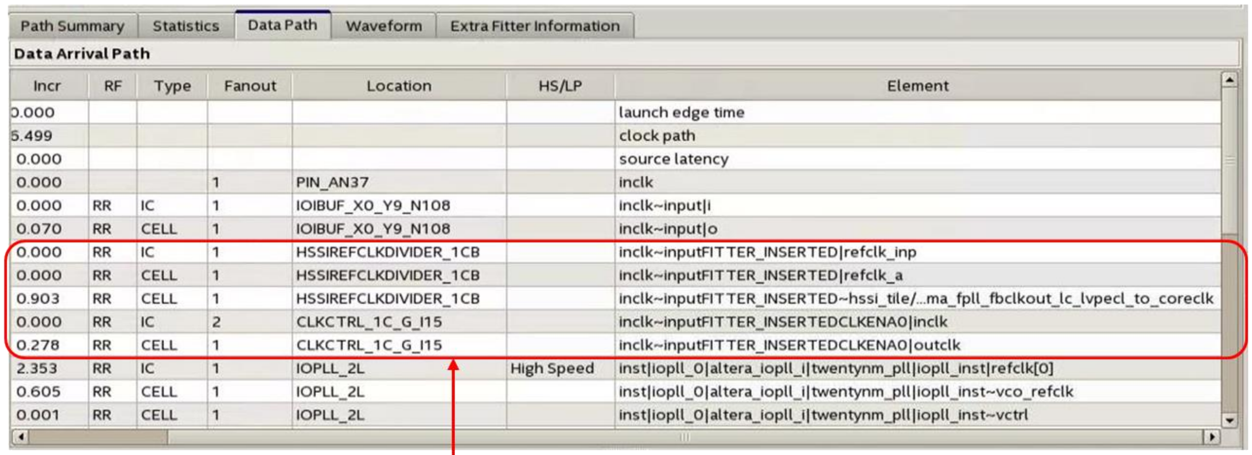
▨ Unsupported PLL cascading

Case#2: Core Reference Clock


Clock from GCLK/RCLK/PCLK network or Core routing
RefClk
Downstream IOPLL/XCVRPLL
Require extra 100ps Clock Uncertainty

- Example: Reference clock from XCVR is routed on GCLK network and fed to reference clock of downstream IOPLL





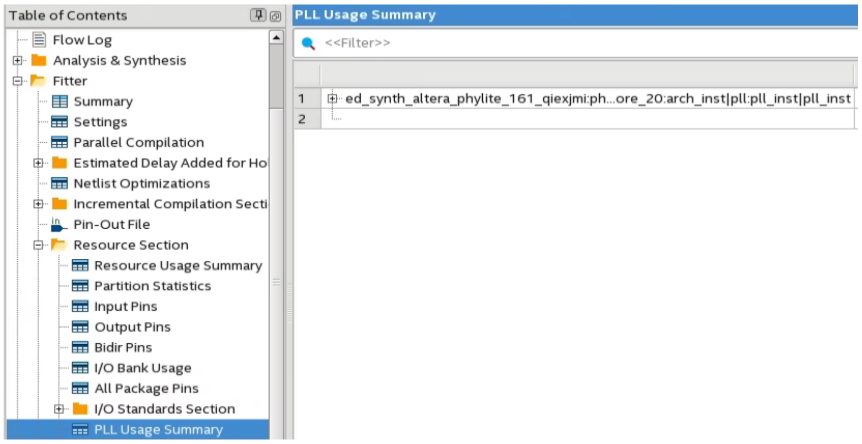
Data Arrival Path

Incr	RF	Type	Fanout	Location	HS/LP	Element
0.000						launch edge time
5.499						clock path
0.000						source latency
0.000			1	PIN_AN37		inclk
0.000	RR	IC	1	IOIBUF_X0_Y9_N108		inclk~input|i
0.070	RR	CELL	1	IOIBUF_X0_Y9_N108		inclk~input|o
0.000	RR	IC	1	HSSIREFCLKDIVIDER_1CB		inclk~inputFITTER_INSERTED|refclk_inp
0.000	RR	CELL	1	HSSIREFCLKDIVIDER_1CB		inclk~inputFITTER_INSERTED|refclk_a
0.903	RR	CELL	1	HSSIREFCLKDIVIDER_1CB		inclk~inputFITTER_INSERTED~hssi_tile/...ma_fpll_fbclkout_lc_lvpecl_to_coreclk
0.000	RR	IC	2	CLKCTRL_1C_G_I15		inclk~inputFITTER_INSERTEDCLKENA0|inclk
0.278	RR	CELL	1	CLKCTRL_1C_G_I15		inclk~inputFITTER_INSERTEDCLKENA0|outclk
2.353	RR	IC	1	IOPLL_2L	High Speed	inst|iopll_0|altera_iopll_i|twentynm_pll|iopll_inst|refclk[0]
0.605	RR	CELL	1	IOPLL_2L		inst|iopll_0|altera_iopll_i|twentynm_pll|iopll_inst~vco_refclk
0.001	RR	CELL	1	IOPLL_2L		inst|iopll_0|altera_iopll_i|twentynm_pll|iopll_inst~vctrl

The HSSI refclk goes through HSSI refclk divder and core clock network via CLKCTRL

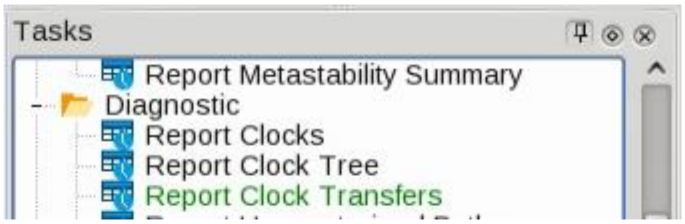
Following are the steps to apply the clock uncertainty to the design:

1. Identify all the PLLs in the design. Go to Fitter → Resource Section → PLL Usage Summary

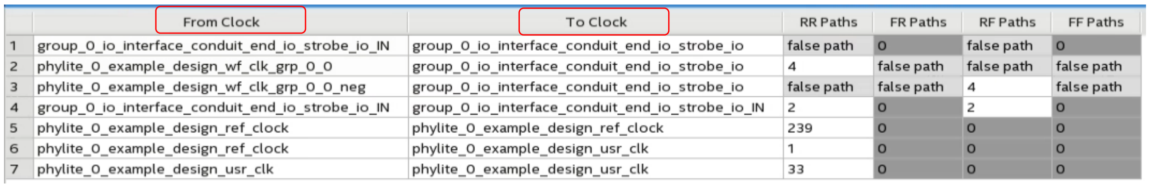


1	ed_synth_altera_phylite_161_qiexjmi:ph...ore_20:arch_inst|pll:pll_inst|pll_inst
2	

2. Report clock transfers in TimeQuest

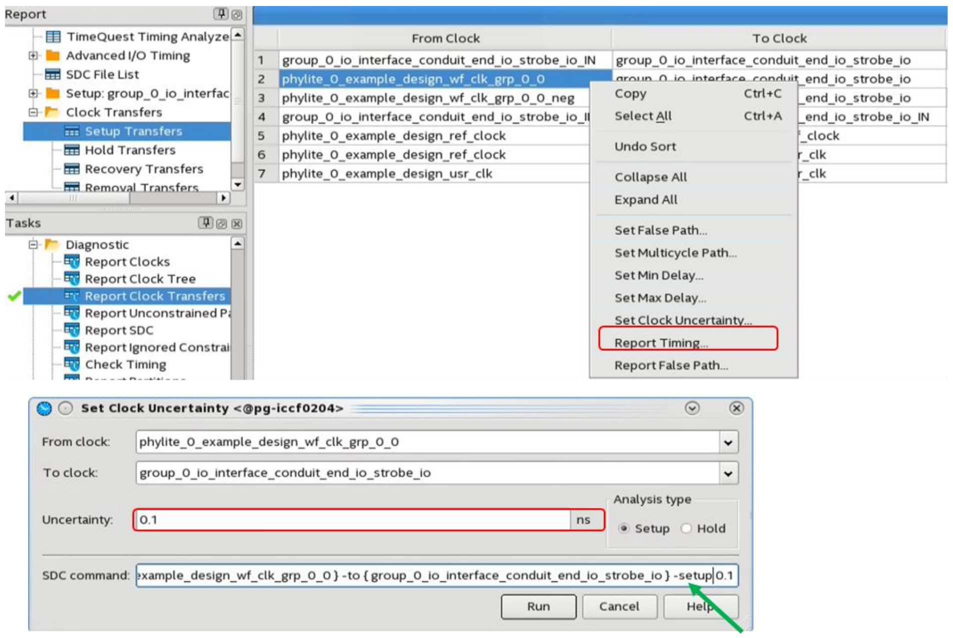


3. Find out clock transfer/s that has/have output of the downstream PLL: From/To/Both



	From Clock	To Clock	RR Paths	FR Paths	RF Paths	FF Paths
1	group_0_io_interface_conduit_end_io_strobe_io_IN	group_0_io_interface_conduit_end_io_strobe_io	false path	0	false path	0
2	phylite_0_example_design_wf_clk_grp_0_0	group_0_io_interface_conduit_end_io_strobe_io	4	false path	false path	false path
3	phylite_0_example_design_wf_clk_grp_0_0_neg	group_0_io_interface_conduit_end_io_strobe_io	false path	false path	4	false path
4	group_0_io_interface_conduit_end_io_strobe_io_IN	group_0_io_interface_conduit_end_io_strobe_io_IN	2	0	2	0
5	phylite_0_example_design_ref_clock	phylite_0_example_design_ref_clock	239	0	0	0
6	phylite_0_example_design_ref_clock	phylite_0_example_design_usr_clk	1	0	0	0
7	phylite_0_example_design_usr_clk	phylite_0_example_design_usr_clk	33	0	0	0

4. Create the set_clock_uncertainty using the macro to add extra 100ps clock uncertainty



	From Clock	To Clock
1	group_0_io_interface_conduit_end_io_strobe_io_IN	group_0_io_interface_conduit_end_io_strobe_io
2	phylite_0_example_design_wf_clk_grp_0_0	group_0_io_interface_conduit_end_io_strobe_io
3	phylite_0_example_design_wf_clk_grp_0_0_neg	_end_io_strobe_io
4	group_0_io_interface_conduit_end_io_strobe_io_I	_end_io_strobe_io_IN
5	phylite_0_example_design_ref_clock	_clock
6	phylite_0_example_design_ref_clock	r_clk
7	phylite_0_example_design_usr_clk	r_clk

From clock: phylite_0_example_design_wf_clk_grp_0_0

To clock: group_0_io_interface_conduit_end_io_strobe_io

Uncertainty: 0.1 ns

Analysis type: [✓ Setup] Hold

SDC command: `xample_design_wf_clk_grp_0_0 } -to { group_0_io_interface_conduit_end_io_strobe_io } -setup 0.1`

Remove "-setup" and put "-add",
please refer to next step for details

5. Add the constraint into the sdc file based on the following conditions:
   a) if the existing clock uncertainty only comes from derive_clock_uncertainty
      set_clock_uncertainty -add -to <clock x> -from <clock y> 0.1

   b) if the existing clock uncertainty comes from derive_clock_uncertainty + additional "set_clock_uncertainty -add"
      set_clock_uncertainty -add -to <clock x> -from <clock y> [expr <existing_value> +0.1]

   c) if derive_clock_uncertainty is overridden by "set_clock_uncertainty" **(no –add)**
      set_clock_uncertainty -to <clock x> -from <clock y> [expr <existing_value> +0.1]

Note:

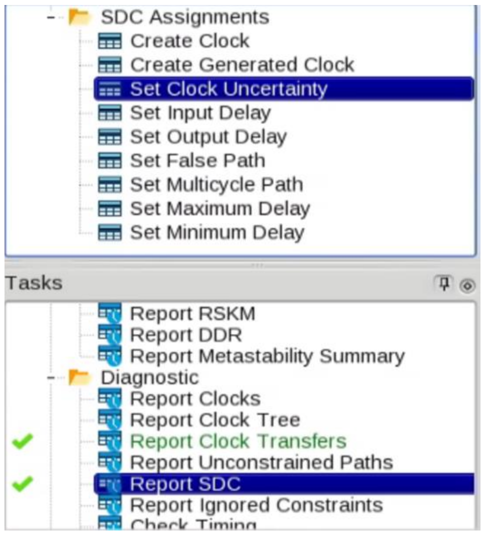
"set_clock_uncertainty -add"

- The clock uncertainty value will be added to the value from "derive_clock_uncertainty"
- If multiple "-add" constraint present, only the value from the last constraint will be added

"set_clock_uncertainty" (no -add)

- overrides any previous clock uncertainty value including "derive_clock_uncertainty"

6. Report SDC and check in the SDC Assignments>Set Clock Uncertainty report, make sure the extra 100ps is added into the affected clock transfer paths



7. Retime or recompile the design and ensure timing closure

8. Perform rigorous hardware testing to ensure design is working properly before going into production