How-To Configure and Implement Altera_PLL Cascading Feature

Introduction: This document describes the PLL cascading feature support using the Altera_PLL megafunction and implementation. Compared to the conventional PLL Cascading; Altera_PLL Cascading feature uses a dedicated cascading clock path between a pair of fPLLs to achieve better jitter performance and save global clock resources.

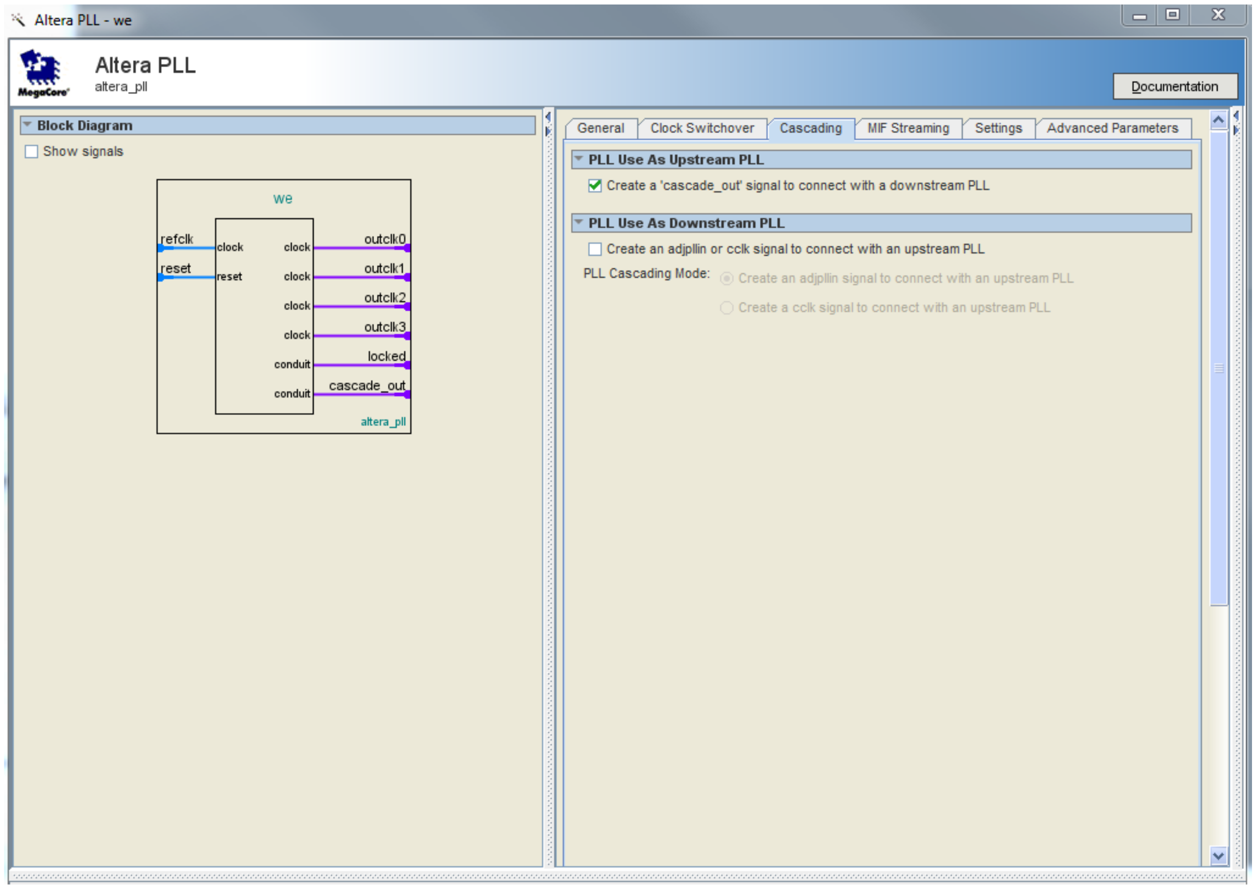
1. **Set Up**

The Quartus® II software version 12.1 and the Altera_PLL megafunction version 12.1 are used throughout this How-To document to describe the required connections and usage for the cascading feature.

2. **Upstream and Downstream PLL configurations**

The Upstream PLL configuration uses the following ports:

- refclk input
- clock output(s)
- cascade out bus (bundles all the output clocks)
  Below figure 2-1 shows the typical upstream configuration.



PLL Use As Upstream PLL

[✓] Create a 'cascade_out' signal to connect with a downstream PLL

PLL Use As Downstream PLL

[ ] Create an adjpllin or cclk signal to connect with an upstream PLL

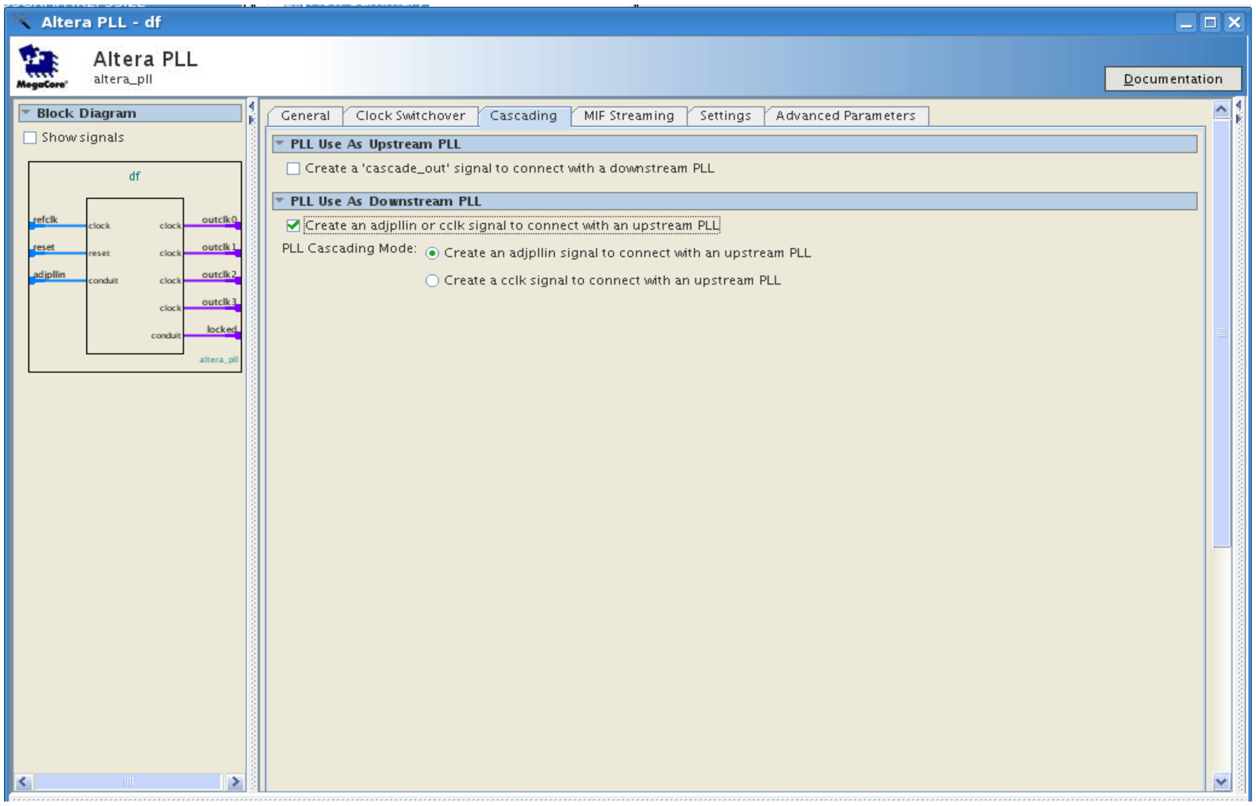
PLL Cascading Mode: ● Create an adjpllin signal to connect with an upstream PLL ○ Create a cclk signal to connect with an upstream PLL

Figure 2-1: Altera_PLL megafunction Upstream Configuration

The Downstream PLL configuration uses the following ports:

- refclk
- output clocks
- adjpllin or cclk (to be used as a cascade input reference clock)

Below figure 2-2 shows the typical downstream configuration.



PLL Use As Upstream PLL

☐ Create a 'cascade_out' signal to connect with a downstream PLL

PLL Use As Downstream PLL

[✓] Create an adjpllin or cclk signal to connect with an upstream PLL

PLL Cascading Mode: [✓ Create an adjpllin signal to connect with an upstream PLL] Create a cclk signal to connect with an upstream PLL

Figure 2-2: Altera_PLL megafunction Downstream Configuration

3. Altera Cascade PLL Implementation:

- The Downstream refclk input should remain unconnected and the internal mux will select the adjpllin or cclk as the cascade input clock depending on the selection you make in the megafunction.
- To ensure the Downstream fPLL is reset and synchronized properly; Altera recommends using the Upstream "locked" output port to connect to the downstream fPLL reset port as shown in figure 3-1.
- The Upstream cascade_out is a bus that bundles all of the output clocks. The Downstream PLL needs to be connected to one of these output clocks at the adjpllin or cclk input port. Below figure 3-1 shows the Cascade fPLL design implementation.

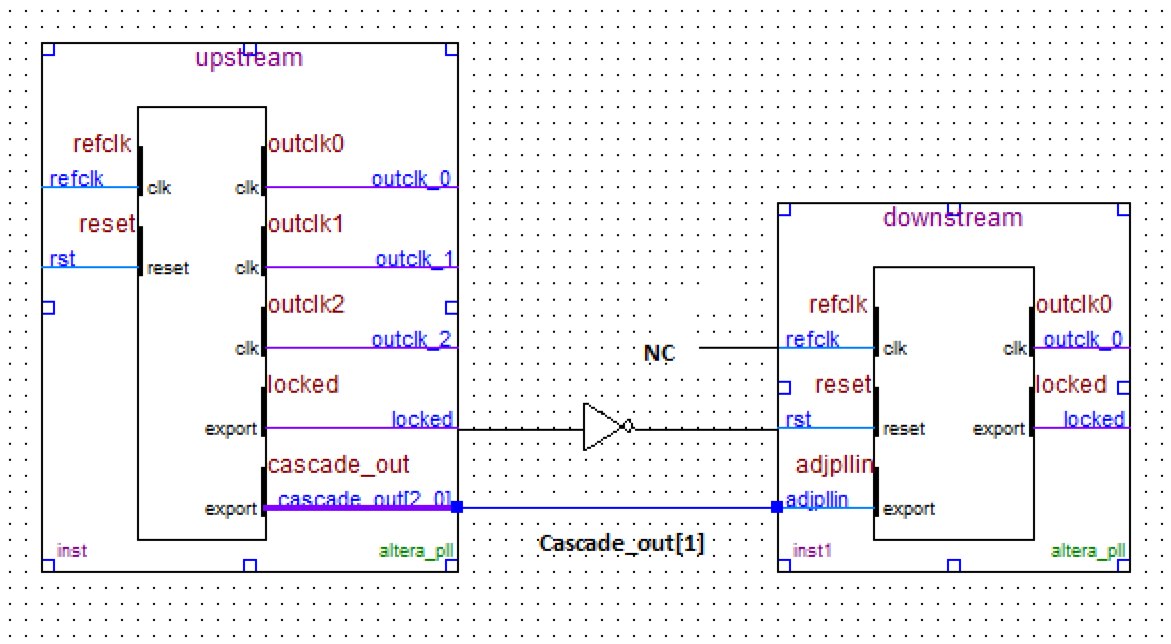


Figure 3-1: Cascade fPLL Implementation

- PLL location assignments are required in the Assignment Editor when using the Casacde PLL feature in Quartus II software version 12.1. Figure 3-2 shows a possible f PLL pair for Casade fPLL implementation (highlighted in yellow) for Stratix® V GX B5 and B6 devices. Below are fPLL location assignment examples:

```
-  set_location assignment FRACTIONALPLL_X90_Y2_N0 -to
“pll_u1|altera_pll:altera_pll_i|altera_stratixv_pll:stratixv_base:fp
ll_0|fpll”
-  set_location assignment FRACTIONALPLL_X90_Y11_N0 -to
“pll_u2|altera_pll:altera_pll_i|altera_stratixv_pll:stratixv_base:fp
ll_0|fpll”
```

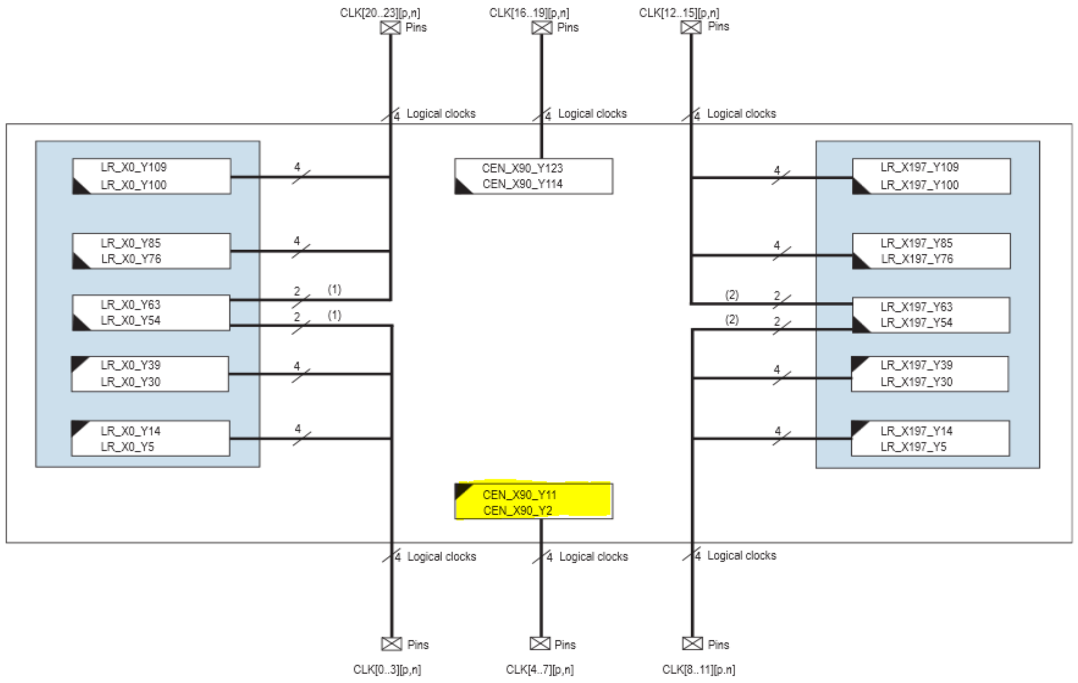


Figure 3- 2 Stratix V GX B5 & B6 fPLL location

- **TimeQuest Requirements** : You will encounter a warning message in Quartus II software version12.1 when cascading fPLLs during the design compilation when using the global clock constraints command in the project.sdc file shows below:
  - Global Clock Constraints Command:
    - `Derive_pll_clocks – create_base_clocks`
  - Warning Message:
    - The master clock for this clock assignment could not be derived. Clock:u_clks_resets|u_pll_c|ddf2_pll_c_inst|altera_pll_i|stratixv_pll|fpll_0|fpll|vcoph[1] was not created"
  - Work around:
    - You must create constraints for all the fPLL output clocks and below is one of the fPLL clock output constraint example:

      -
        ```
        create_clock -name clock_a -period 4.082
        [get_pins
        {pll_a|altera_pll_i|stratixv_pll|counter[0].out
        put_counter|divclk}]
        ```

- **Adjpllin vs cclk cascading implementation**: Adjpllin is used for inter-cascading between fPLLs that are not in the same fPLL pair and cclk is used for intra-cascading within the same pair of fPLLs. Routing will be different as adjpllin crosses fPLL boundaries. Figure 3 -3 shows the physical adjpllin and cclk cascading implementation from the Chip Planner. The C-Counter Clock source from output counter 4 or output counter 13 is used for cclk connection. When using the adjpllin input, it will be connected to the nearest output counter from the adjacent fPLL.

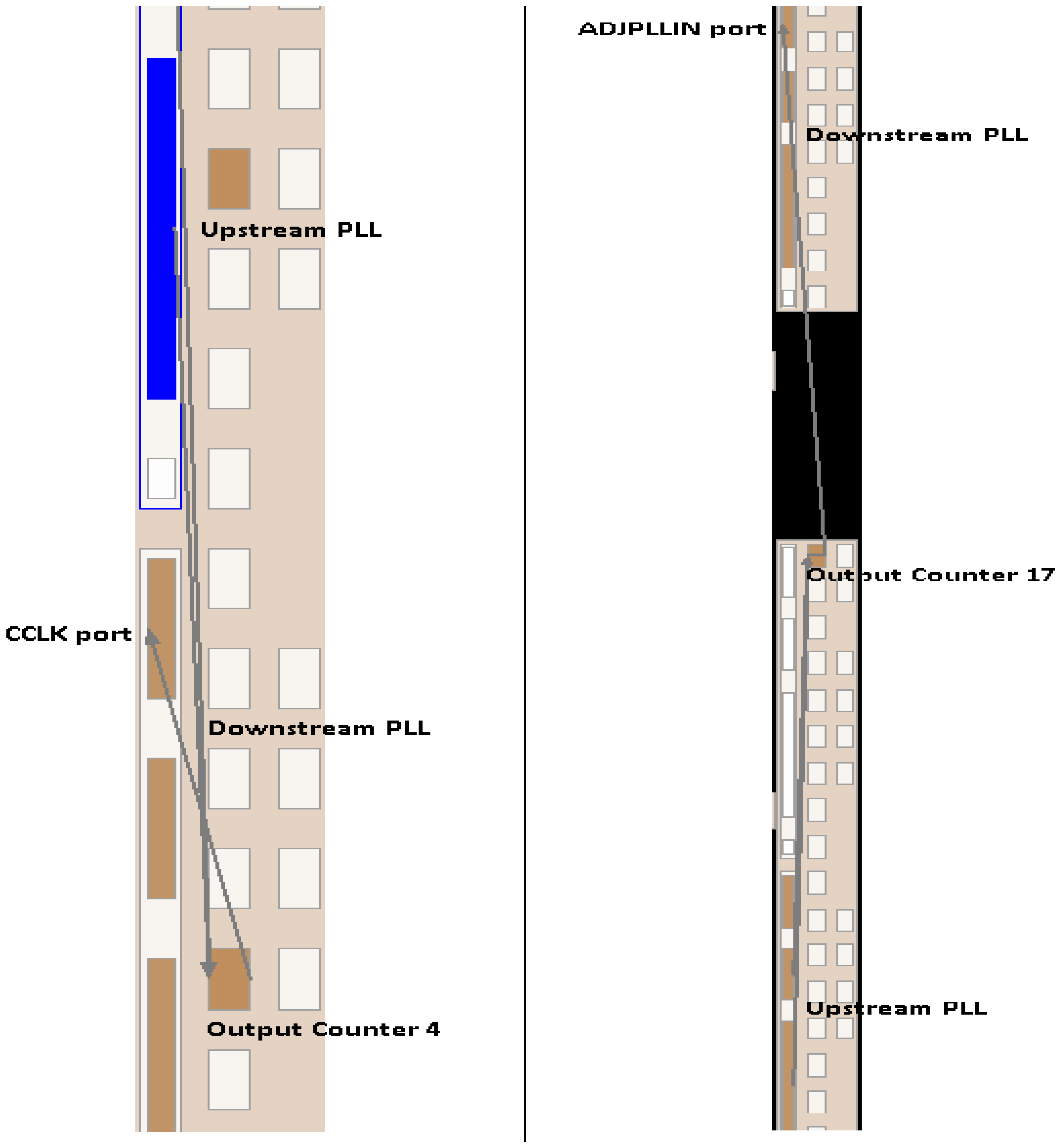


Figure 3- 3 adjpllin vs cclk cascading implementation

4. Conclusion

This document demonstrated how to use the Altera_PLL megafunction to configure the PLL as an upstream or downstream PLL using a dedicated cascading PLL clock path, as well as highlighted the assignments and .sdc requirements when using this feature.

2. Revision History

Revision	**Changes Made**	**Date**
V1.0	Initial release	March 2013