Using Global Resources in Actel's Axcelerator Family

Introduction

Actel's Axcelerator FPGA family offers the most flexible global network scheme of any antifuse-based FPGA to date. This architecture provides eight segmentable chip-wide global networks, eight embedded PLLs, and available dedicated power-on reset/set signals. This application note describes these global networks, discusses how to assign these global networks in a design flow, and reviews design rules for input clock buffer assignment. Details concerning Axcelerator PLLs will not be discussed in this application note, as they are described in Actel's *Axcelerator Family PLLs and Clock Management* application note.

Architecture

Global Network Architecture Overview

All eight global networks in the Axcelerator family can be accessed by external signals, internal signals, or by the PLLs. Each family member has four types of global signals: HCLK, CLK (RCLK), GCLR, GPSET. There are four hardwired clock drivers (HCLK) per device, all located on the north end, which can directly drive the clock input of each R-cell. The hardwired HCLK networks have no antifuse connections between the source signal and register clock pins. This allows for minimal clock skew and propagation delays and makes the HCLK network the fastest network on the device. Each of the four routed clock drivers (CLK) are located on the south end of the device and can drive the clock, preset, or enable pin of an R-cell or any input of a C-cell. Global clear (GCLR) and global preset (GPSET) can drive the clear and preset inputs of each R-cell as well as each I/O register on a chip-wide basis at power-up.

Each of the HCLK and CLK networks is associated with a PLL. The PLL can be used to condition the incoming clock signal, or it can be bypassed completely. Once a signal is brought onto one of the global networks, it is sent to multiplexors (HCLKMux and RCLKMux), which are centered on each core tile in the device. From there, signals on HCLK are distributed horizontally throughout the tile to columns that connect directly to the CLK pin of every flip-flop in the tile. Signals on CLK are distributed vertically throughout the tile to rows that can connect to flip-flops and C-cells (Figure 1).
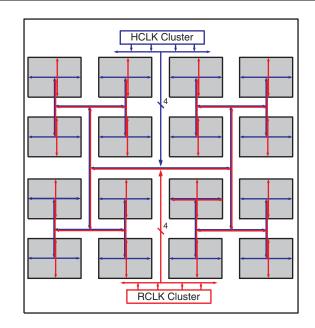

Figure 1 • Global Network Distribution in Axcelerator Devices

CLK and HCLK Global Network Drivers

The global networks offer the flexibility of being driven by different types of sources such as external pins, internal nets, and PLLs. This is made possible by the multiplexor architecture shown in Figure 2. In order for an external signal to directly drive a global network, it will first pass through an I/O buffer macro, which defines the I/O standard and voltage for the clock signal and then onto the clock network driver through a series of MUXes. An internal signal will automatically connect from the internal routing by an internal buffer called "CLKINT_W," before it passes to the MUXes and the clock driver. Clock signals driven by a PLL will only pass to the clock driver through the MUXes.
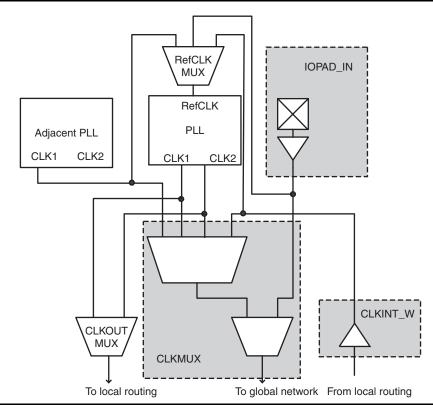

Figure 2 • *Clock Network Access Architecture*

Global Network Connections

The unique feature of the HCLK network is that it is hardwired to the clock pin of all registers in the device. A MUX in front of each register cell determines if the register will be driven by HCLK, CLK, or some other signal. The HCLK network is hardwired to the input of the MUX, eliminating the need for any antifuse connections in the HCLK network. The HCLK network is also directly connected to I/O registers, PLLs, and embedded RAM and FIFOs in the device. HCLK cannot drive any other pins on any other module.

The CLK network offers the advantage of flexibility in that it can connect to a wide variety of module pins on Axcelerator devices. It can connect to CLK, PRE, CLR, EN pins on flip-flops, and any input on C-cells. It can also directly drive output ports and PLLs. It cannot connect to data input pins on R-cells such as D, A, B, S. If you attempt to connect CLK to one of these pins, Designer automatically inserts a buffer between the CLK and the pin you are trying to drive and issues a warning during compile to notify you of the buffer insertions.

Design Flow Considerations

Assigning Global Networks to I/O Banks

Global signal assignment to I/O banks is no different from regular I/O assignment to I/O banks (see Actel's application note, *Axcelerator I/O Selection Guide*) with the exception that you are limited to the pin placement location and number of clock signals. Determining how to place globals can be simplified if you follow these guidelines:

1. Only global signals compatible with both the V_{CCI} and V_{REF} standards can be assigned to the same bank. Table 1 lists global macros that are compatible with each other.

2. All HCLK drivers are located on the north side of the device; all CLK drivers are located on the south side of

Table 1 • *Compatible Global Input Macros for Legal I/O Placement*

	V_{REF}	H/CLKBUF/I (LVTTL)	H/CLKBUF_GTLP25	H/CLKBUF_GTLP33	H/CLKBUF_HSTL_I	H/CLKBUF_LVCMOS18	H/CLKBUF_LVCMOS25	H/CLKBUF_LVDS (all V_{REF})	H/CLKBUF_LVPECL (all V_{REF})	H/CLKBUF_PCI/X	H/CLKBUF_SSTL2_I/II	H/CLKBUF_SSTL3_I/II
H/CLKBUF/I (LVTTL)	1.0	Y		Y					Y	Y		
H/CLKBUF/I (LVTTL)	1.5	Y							Y	Y		Y
H/CLKBUF_GTLP25	1.0		Y				Y					
H/CLKBUF_GTLP33	1.0	Y		Y					Y	Y		
H/CLKBUF_HSTL_I	0.75				Y							
H/CLKBUF_LVCMOS18	N/A					Y						
H/CLKBUF_LVCMOS25	N/A						Y					
H/CLKBUF_LVDS	1.0		Y					Y	Y			
H/CLKBUF_LVDS	1.25							Y	Y		Y	
H/CLKBUF_LVPECL	1.0	Y		Y					Y	Y		
H/CLKBUF_LVPECL	1.5	Y							Y	Y		Y
H/CLKBUF_PCI/X	N/A									Y		
H/CLKBUF_SSTL2_I/II	1.25						Y				Y	
H/CLKBUF_SSTL3_I/II	1.5	Y							Y	Y		Y

the device. You must use macros that start with HCLK* to assign global signals to the north side of the device and use macros that start with CLK* to assign global signals to the south side of the device.

3. A maximum of four networks of each type (HCLK, CLK) are available. This includes internally driven clock networks and PLL driven clock networks. For example, if you use one internally driven HCLK (HCLKINT macro), and one HCLK network driven by a PLL (PLLHCLK macro), only two more HCLK networks are available for use.

4. There are two pins associated with each global network ("P" and "N") to allow for differential signal inputs. For nondifferential inputs, the signal must be assigned to the "P" pad. In this case, the "N" pad will be available for user I/O. LVDS and LVPECL require both the "P" and "N" pads. PinEditor will force you to do this. If you are setting this constraint in PDC, you must ensure that both the "P" and "N" inputs are placed on the correct pads for the same global network.

Global Network Access Macros

Table 1 lists all of the available macros for I/O global network assignment. In addition to this, the global networks can be accessed by internal signals using the HCLKINT or CLKINT macros and can be driven by PLLs with the PLLRCLK or PLLHCLK macros. Refer to Actel's *Antifuse Macro Library Guide* for a detailed explanation of each macro.

Implementing Global Macros in Schematic Designs

Adding Global network buffers in schematics for Axcelerator is no different from any other device family. Please refer to the *Actel Quick Start Guide* and the interface guide for the tool you are using for detailed information on available netlist attributes. A general recommendation for schematic designs is related to naming clock networks for timing analysis purposes in Actel's Timer. Internally generated clocks are referred to by the instance name of the FF output driving the clock signal, so

we recommend giving the FF an instance name that will help you uniquely identify the clock network it is driving. Externally driven signals will be referred to by their top-level port name.

Implementing Global Macros in VHDL/Verilog Designs

The current synthesis tool libraries will only infer the default global network macros such as CLKBUF, CLKINT, and HCLKBUF. All other global macros will have to be instantiated manually into your HDL code. The following are a few examples of global macro instantiations that you can copy and paste into your code:

CLKBUF_LVCMOS25 Driver

VHDL:

```
component clkbuf_lvcmos25
port (pad : in std_logic;
   y : out std_logic);
end component
begin
-- concurrent statements
u2 : clkbuf_lvcmos25 port map (pad =>
ext_clk, y => int_clk);
end
```

Verilog:

```
module design (_____);
   input _____;
   output _____;
clkbuf_lvcmos25 u2 (.y(int_clk),
.pad(ext_clk);
   endmodule
```

HCLKBUF_LVDS Driver

VHDL:

```
component hclkbuf_lvds
port (padp, padn : in std_logic;
   y : out std_logic);
end component
begin
-- concurrent statements
u2 : hclkbuf_lvds port map (padp =>
ext_clk_p, padn => ext_clk_n, y =>
int_clk);
end
```
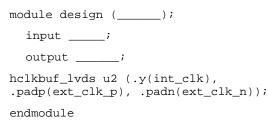
Verilog:

```
module design (_____);
   input _____;
   output _____;
hclkbuf_lvds u2 (.y(int_clk),
.padp(ext_clk_p), .padn(ext_clk_n));
   endmodule
```

Viewing Clock Networks in Actel's Designer

Designer's user tools (NetlistViewer, ChipEditor, Timer) display a post-compile version of your original netlist. This version of your netlist includes some of the internal modules that allow the Axcelerator global networks to be so flexible. It also includes buffers that were added during compilation, and some of the logic optimization that has been done in compiler. Therefore, you may see a few things in these user tools that are different from your original netlist. Figure 3 gives a summary of the most common differences between your netlist and what you see in Designer's user tools for global networks.
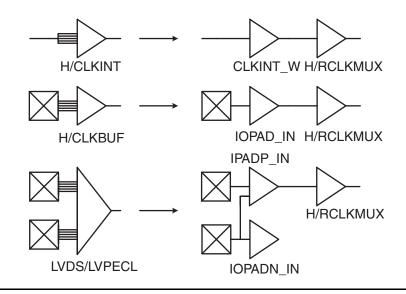

Figure 3 • Differences Between an Original Netlist and a Netlist Viewed in Designer

Working with PLLs

In this section, we will discuss legal global network connections to PLLs in the Axcelerator family. For detailed information on using PLLs, please refer to the *Axcelerator Family PLLs and Clock Management* application note. Actel recommends using the external RefCLK pins to directly drive the RefCLK of the associated PLL for reduced propagation delays and clock distortion. However, Axcelerator does offer the flexibility to connect other signals to RefCLK. You can assign any other signal in the device to drive RefCLK through use of the PLLINT macro. When the PLLINT macro is used, the CLKINT_W buffer that is described in the "CLK and HCLK Global Network Drivers" section on page 2 will be used.

Connections from PLL outputs are done by using the PLL output macros (PLLOUT, PLLHCLK, PLLRCLK), and connections to the PLL are done by using the PLL input macro (PLLINT). Simply instantiate these macros into your source as shown in Figure 4.

Each PLL is associated with a single clock network. Therefore, there are some limitations as to what can and cannot be driven by the CLK1 and CLK2 outputs of the PLL. In Figure 4, we show the basic mechanism for the PLL to connect to local routing, and the global networks. Based on this structure, Table 2 on page 5 describes the limitations.
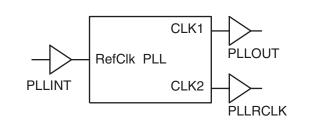

Figure 4 • *Using PLL Access Macros*

Table 2 • *Limitations for PLL Connections to Global Signals*

CLK1 and CLK2 of the same PLL cannot both drive global networks, however, CLK1 can be routed to drive the global network of an adjacent PLL.
If RefCLK is driven by an internal signal, or a non-RefCLK I/O, the PLLINT macro must be used.
CLK1 and CLK2 of a PLL cannot both drive local routing.
If one PLLOUT macro is used, and one PLL output pin is unused, RefCLK must be driven by the dedicated external RefCLK pin. Otherwise, Designer will not be able to determine where to place the PLL.
If the PLL RefCLK pin is driven by a standard input or internal signal, then the PLL output must be connected to PLLHCLK or PLLRCLK. Otherwise, Designer will not be able to determine where to place the PLL.
If CLK2 drives the PLLHCLK macro, and RefCLK of the PLL is driven by an HCLK input macro, then the fanout of the input macro cannot be greater than one. This is because the "CLKMUX" module in Figure 2 on page 2 is already used by the PLLHCLK macro. This limitation does not apply to the PLLRCLK macro.
HCLKINT cannot drive the RefCLK input of the PLL.
Only four CLKINT_W modules are available on each PLL Cluster.
BIBUF macros cannot drive RefCLK.
Either CLK1 or CLK2 must drive one of the following macros (PLLOUT, PLLHCLK, PLLRCLK) when a PLL is used.

Unused Global Input Pins

When pins that are designated as CLK or HCLK are not used as clock inputs, they may be used as regular I/Os, or may be left floating. Actel's Designer software will automatically configure these unused I/Os as tristated outputs. If a nondifferential global is being used such as CLKBUF, the associated "N" pad is available to be employed as user I/O. If not used, this pin will also be treated as an unused I/O as stated above.

Using the Dedicated Clear/Preset Networks

By default, all flip-flops in the Axcelerator family will power-up in the reset state due to the hardwired power-on reset circuitry via the GCLR network. This feature is built in to the device, and cannot be controlled by the user. Actel has implemented an option to power-up the device with flip-flops in the Set state (logic '1') instead of the Reset state via the GPSET network. This may be accomplished by selecting to program the GBSET Fuse in the "Generate Programming File" window when "Fuse" is selected in Designer. In addition to these built-in networks a user-defined clear/preset network may be designed in addition to this feature to control reset of flip-flops during normal operation. This network can be driven by one of the RCLK networks, or by local routing resources.

Clock Segmentation

The Axcelerator global network architecture has several points at which the networks are multiplexed. One level of MUXes is at the core tile level as described in "Global Network Architecture Overview" section on page 1. The second level of MUXes is at a lower level within the tile. Both levels of MUXes have the capability to segment the networks into small, local segments that operate independent of other tiles in the device. This can add large performance improvements for designs that have many smaller clock networks. However, this capability is not supported in current software versions. It will be added in a later release of the Designer software package. Please see future versions of the Designer software release notes for updates.

Conclusion

The Axcelerator family offers flexibility in its global networks that is unparalleled by any other antifuse FPGA. While a few limitations do exist, by understanding those limitations, designers can swiftly implement their designs in Axcelerator and reach production with ease.

Related Documents

Application Notes

Axcelerator Family PLLs and Clock Management

http://www.actel.com/documents/AX_PLL_AN.pdf

Axcelerator I/O Selection Guide

http://www.actel.com/documents/AX_IOSelection_AN.pdf

User Guides

Antifuse Macro Library Guide

http://www.actel.com/documents/libguide_UG.pdf

Actel Quick Start Guide

http://www.actel.com/documents/gettingstarted_UG.pdf

http://www.actel.com

Actel Corporation
955 East Arques Avenue
Sunnyvale, California 94086
USA
Tel: (408) 739-1010
Fax: (408) 739-1540

Actel Europe Ltd.
Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Tel: +44 (0)1276 401450
Fax: +44 (0)1276 401490

Actel Japan
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150   Japan
Tel: +81 03-3445-7671
Fax: +81 03-3445-7668

Actel Hong Kong
39th Floor
One Pacific Place
88 Queensway
Admiralty, Hong Kong
Tel: 852-22735712

5192698-1/8.03