

Efficient Use of ProASIC Clock Trees

One of the main architectural benefits of ProASIC is the clock tree. Each device of the ProASIC FPGA family offers 4 global trees. Each of these trees is based on a collection of spines and ribs that reaches all the tiles in their regions (Figure 1). This flexible clock tree architecture allows users to map up to 56 different internal/external clocks in an A500K270 device. Table 1 summarizes the total number of clock spines available for each device.

This application note focuses on the use of these clock trees and spines to meet the requirements of clock intensive applications. It starts with a brief review of the ProASIC routing architecture using a global to route a high-fanout clock. It also gives design recommendations to exploit these low-skew routing resources and to deal with timing-critical high-fanout signals.

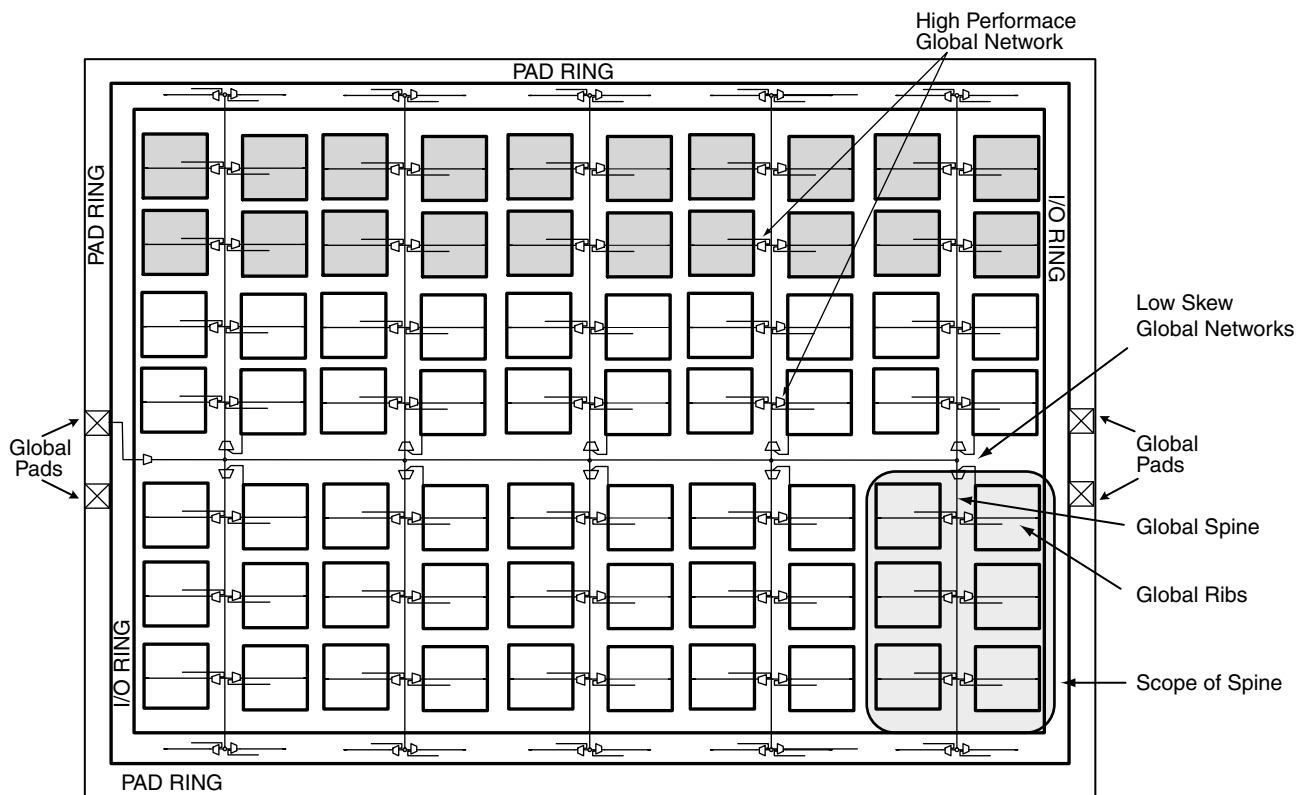


Figure 1 • A500K130 Global Routing Resources

Table 1 • Distribution of Spines per ProASIC Device

| Device | Number of Global Networks (clock Trees) | Number of Spines per Clock Tree | Identification of Spines | Total Number of Spines in the Die |
|----------|---|---------------------------------|--------------------------|-----------------------------------|
| A500K050 | 4 | 6 | T1...T3 & B1...B3 | 24 |
| A500K130 | 4 | 10 | T1...T5 & B1...B5 | 40 |
| A500K180 | 4 | 12 | T1...T6 & B1...B6 | 48 |
| A500K270 | 4 | 14 | T1...T7 & B1...B7 | 56 |

Background

Designers typically pay close attention to clock tree structures during their evaluations and design decisions. Although many FPGA devices offer several clock trees, they are used as full segments; i.e., users cannot split these segments to handle multiple clocks. The most sophisticated FPGA architectures offer quadrant clocks as the only remedy for this limitation (none of them allows users to split these wide segments). ProASIC devices provide an innovative architecture for global routing trees, allowing users to split them into spines for routing either internal/external clocks or high fanout nets.

Global Clock Tree Distribution

When needed, a high fanout global signal or internal net can be mapped to low-skew global routing and can cover the whole die. [Figure 2](#) shows an example of a clock signal that reaches most of the cells in the device.

Besides the explicit use of global pads, Designer, Actel's place-and-route software, offers other ways to assign nets to global resources by means of the "set_global NetName" constraint (see the *Designer Series User's Guide* for more details). Assigning nets to global resources is recommended in particular when the critical path includes high-fanout nets with a high delay penalty. Notice that the tool has a default assignment of the highest-fanout nets to global routing resources. However, if the nets assigned automatically have enough timing margin, it is recommended that you reassign these nets by means of the "set_noglobal NetName" constraint, and replace them with critical ones, even if the latter have lower fanout.

Users can assign a particular net or signal and reassign an automatically-assigned high fanout signal or net to make best use of the global routing trees and to cope with delay and skew issues. Other constraints, introduced in the "Appendix" on page 7, help set the minimum fanout limit to consider nets for automatic assignment to a global tree.

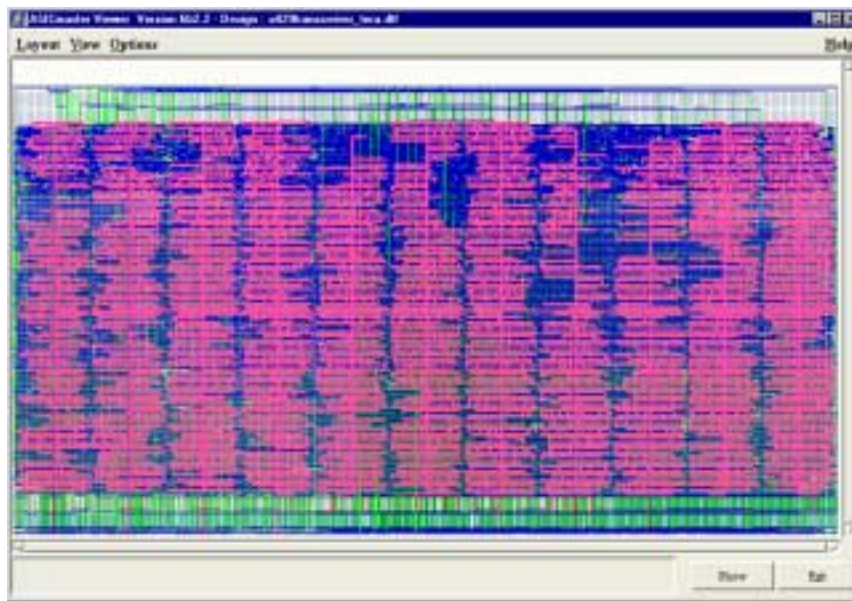


Figure 2 • Global Clock Tree Distribution

Individual Spine Mapping

To force the place-and-route tool to map one particular spine to a designated signal or net, users need to use the following constraint:

```
use_global_spine [NetName | SignalName];
```

Where:

“spine” is one of the spines T1 to T<n> or B1 to B<n>

“NetName” is the name of the net

SignalName is an external or internal user signal

Applying such a constraint guides Designer Software to place all destination cells within the spine region ([Figure 1 on page 1](#)) and to route the net using the specified spine from the global network.

[Figure 3 on page 3](#) shows the mapping of spines T1 and B2 to two signals of the design and the mapping of the whole clock tree to an external clock (highlighted in blue). The following are constraints that were used for the design of [Figure 3 on page 3](#):

```
use_global T1 transmit1/lock1;
use_global B2 receiver_2/edge;
```

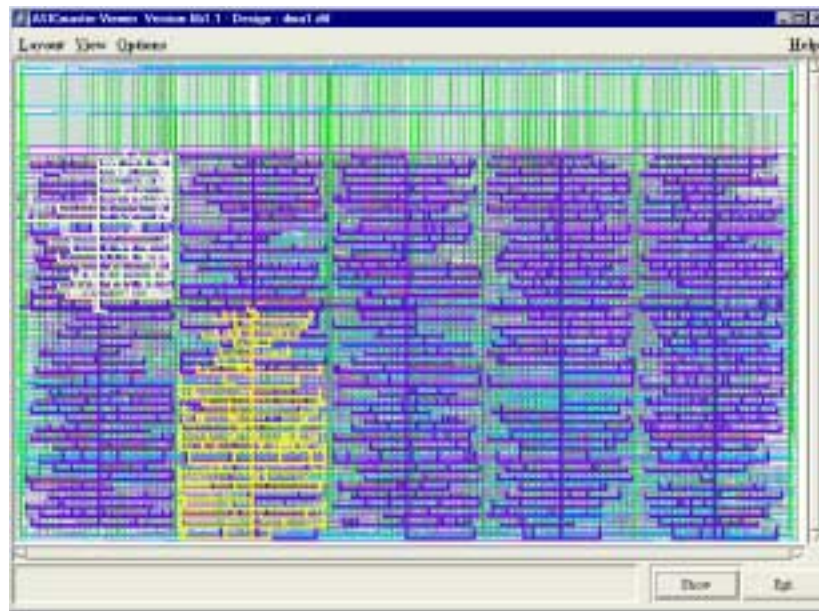


Figure 3 • Mapping Individual Spines

Assigning Multiple Spines

The clock tree and the place-and-route software allow users to assign one spine to a signal. (Multiple spine assignment must be done with care!) It is recommended that you target only a bottom/top pair of the same spine (Examples are T1 and B1, which are depicted in [Figure 4 on page 4](#)). Moreover, the user should split the signal with an additional buffer, assign the top spine to the signal, and assign the bottom spine to the output of the nets.

For example, if Finger0/Clock40 is the signal that needs to be routed through T1 and B1, the user needs to add an auxiliary buffer and split the original destination cells (of Finger0/Clock40) between Finger0/Clock40 and the output of this buffer. The user must enter the following constraints:

```
use_global B1 Finger0/Clock40;
use_global T1 Finger0/Clock40_AuxiliaryBuffer_output;
```

Notice that this involves a choice between the destination cells of the original signal and the destination cells of the newly-introduced buffer. The user should check the potential skew that may occur because of input and output routing of the auxiliary buffer.

Note that following the recommendation to use a spine pair (Tn, Bn) may lead to uncontrollable skew and possible setup- or hold-time violations. This is because the horizontal clock rib is a single segment that cannot be split.

Assigning T4 and T5 in [Figure 4 on page 4](#) will make use of routing resources other than the horizontal clock rib and will introduce skew.

As a generalization of this rule, users should either use the entire global for very-high-fanout nets/signals, or designers should limit the mapping of a net/signal to either one spine or a top/bottom spine pair (Tn, Bn).

Design Recommendations

The following sections introduce some design suggestions to efficiently use these resources and avoid timing pitfalls.

Economy of the Global Network

While the place-and-route tools are sophisticated enough to do the most compact placement, the user should guide them by setting placement constraints on the destination blocks of each clock signal. Placing these blocks in a limited zone of the die forces the router to leave the spines outside the zone for other mapping purposes such as another limited internal/external clock or a high fanout net. In [Figure 5 on page 4](#), the three clock signals, (highlighted in red, yellow, and light-blue) as well as their destination cells, have been limited to particular zones. The rest of the free spines for each of these global low-skew networks can be used for assigning other signals. Notice that there is no requirement for interleaving regions with each of the clocks. In [Figure 5 on page 4](#), the blue and yellow clock zones as well as the red and the yellow zones interleave.

Hierarchical Blocks and Placement Constraints

The place-and-route tools support floorplanning constraints such as the placement of hierarchical blocks in particular zones, avoiding other zones, etc. Also, as stated earlier, splitting a spine introduces implicit placement constraints

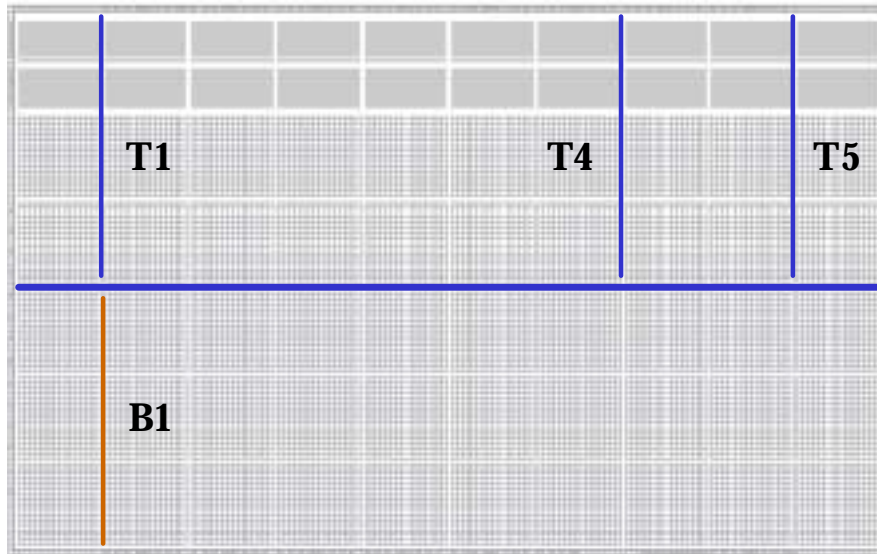


Figure 4 • Multiple Spines Assignment



Figure 5 • Limitation of the Clock Scope and Economy of Low-Skew Spines

on the destination cells. Trying to ease the delay penalty of some nets in the various hierarchical blocks by splitting spines requires that users check the timing margins for signals (or nets) that connect cells of different hierarchical blocks because these may introduce larger routing delays.

This pitfall is illustrated in Figure 6 on page 5 where Block A and Block B have significant interconnection and both blocks include high-fanout nets mapped to distant spines, namely B1 and T4. The delay reduction on the high fanout nets is counter-balanced by the delay associated with the interconnect net (indicated by dashed lines).

Pin Placement Constraints and Implications

The potential problem discussed in the previous section may apply when dealing with pin assignment or placement constraints.

If a destination block is far away from the pin location, the delay penalty introduced by routing the I/O to the hierarchical block may make the timing path critical. This may occur even if users map high-fanout nets to very distant spines.

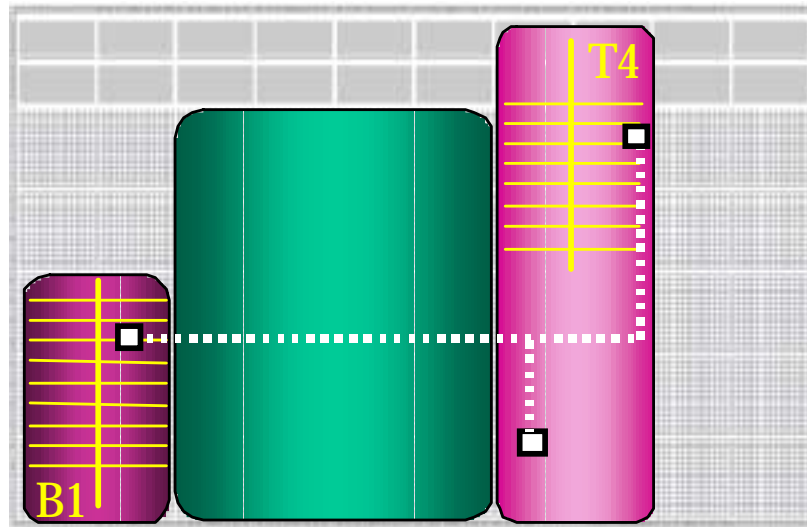


Figure 6 • Spin Inference and Potential Pitfalls

Splitting Spines and “Macros”

A macro is a hierarchical block with associated placement constraints. Its usage optimizes the place-and-route run-time and the quality of results, especially when the block is instantiated several times. Once a macro is defined, users can flip it, rotate it and slide it to the left or right. [Figure 7](#) shows a macro and its use in four instances of the same block. The macro has been moved and flipped. (See the *Designer Series User’s Guide* for more details on the concept and usage of “macros”).

If users create a macro (i.e., explicit placement constraints associated with cells of a hierarchical block) and need to map a critical signal or internal/external clock to a spine, they need to ensure that:

1. The implicit placement constraints, generated by the mapping of the spine, do not conflict with the explicit placement of the macro
2. They only flip or slide the macro and never rotate it.

When following these recommendations, it is easy to duplicate the macro and the spine assignment to handle multiple clocks or high-fanout nets. [Figure 8 on page 6](#) illustrates a networking application where the top-level design includes 14 instances of a channel (and hence 14 different clock domains). Notice that timing validation should be performed for each instance of the macro as well as for the top-level design.

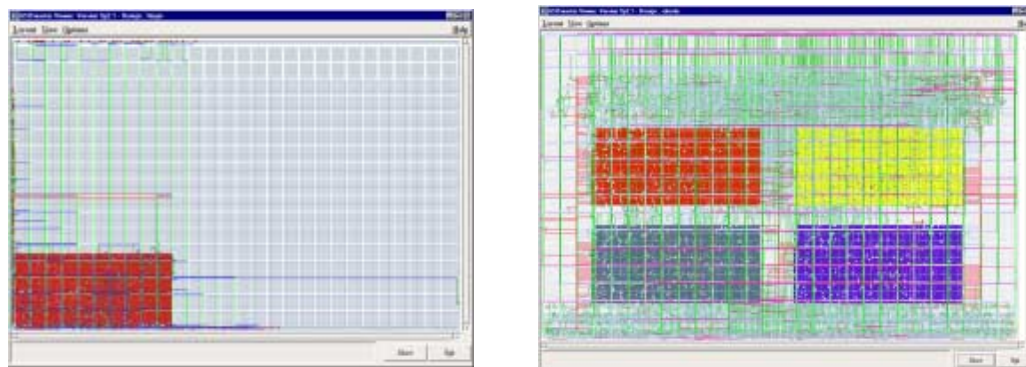


Figure 7 • Use of a Macro for Multiple Instances

When using the aforementioned concept of “macro” associated with spine routing resources, it is important to watch the number of logic tiles in the scope of each spine (Table 2). The ProASIC architecture does not allow top spines to reach as many logic tiles as bottom spines can reach. This is because the embedded SRAM blocks are on

the top (north) side of the die and they need to be fed by the top clock spines. The SRAM embedded blocks are equivalent to eight tiles in each row. All the ProASIC devices include two rows of embedded memory blocks. The ProASIC A500K050 device includes only one row (equivalent to eight tiles).

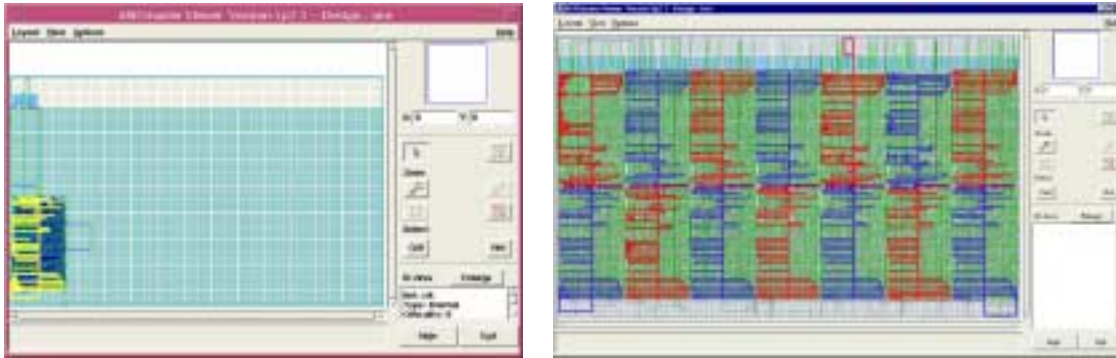


Figure 8 • Macros with Clock Spine Inference to Clocks

Table 2 • Number of Tiles in Bottom and Top Spines of ProASIC Devices

| Device | # of Tiles in Top Spine | # of Tiles in Bottom Spine | Top Spine Height | Spine Height |
|----------|-------------------------|----------------------------|------------------|--------------|
| A500K050 | 768 | 1024 | 24 | 32 |
| A500K130 | 1024 | 1280 | 32 | 40 |
| A500K180 | 1280 | 1792 | 40 | 56 |
| A500K270 | 1792 | 2048 | 56 | 64 |

Conclusion

The flexible use of the ProASIC clock spine allows designer to simultaneously handle multiple design requirements. Users implementing clock-resource-intensive applications can easily route external or gated internal clocks using global routing spines. Users can also drastically reduce delay penalties and save buffering resources by mapping high-fanout critical nets to spines. The design suggestions introduced in the previous section help designers to make efficient use of these resources and combine some other software features to reach their design goals. The exploitation of unused spines to map internal/external clocks with limited scope is definitely an easy task when targeting ProASIC.

Appendix

Summary of Constraints to Manage ProASIC Global Routing Resources

Automatic Assignment of High-Fanout Nets to Globals

Users can set the minimum fanout of nets to be considered for automatic assignment to globals. The default value in Designer Software is set to 32. To change it to a higher or a lower value, users can use the following constraint:

Syntax

```
set_auto_global_fanout <IntegerNumber>;
```

Example

```
set_auto_global_fanout 12;
```

This implies that a net must have at least a fanout of 12 before being considered for automatic assignment to a global resource.

Forcing a Signal or a Net to Global

To force nets or signals to use global resources, users can use the following constraint:

Syntax

```
set_global NetName;
```

Reassigning a Signal or a Net

If the automatic assignment mechanism forces a net or a signal on a global resource, users can override this assignment by means of the following constraint:

Syntax

```
set_noglobal NetName;
```

Turning Off The Automatic Assignment

To turn off the default action that automatically assigns the global resources to high fanout nets, users can use the following constraint:

Syntax

```
dont_fix_globals;
```

Spine Mapping to Signals or Nets

To route a net or a signal using a global spine, users can force the place-and-route tools using the following constraint:

Syntax

```
use_global_spine <NetName | SignalName>;
```

Remember that these constraints force the placer to place all other cell instances connected to that net within the spine region. The router will route this net using the specified global spine resource.

Also, there are some restrictions related to each device of

the ProASIC family:

A500K050 has 6 spines T1 to T3 and B1 to B3,

A500K130 has 10 spines T1 to T5 and B1 to B5,

A500K180 has 12 spines T1 to T6 and B1 to B6 and

A500K270 has 14 spines T1 to T7 and B1 to B7 spines

Bounding the Placement of Hierarchical Blocks

The following constraint can be used to place an individual cell or a block within a certain boundary of the die. This constraint can be used for either complete or partial floorplanning of the design.

Syntax

```
set_location (x1, y1 x2, y2) BlockName!*
```

Example

If the design has 4000 tiles and the targeted ProASIC device is the A500K130, use

```
set_location (1,1 80,70) *;
```

Actel and the Actel logo are registered trademarks of Actel Corporation.
All other trademarks are the property of their owners.



<http://www.actel.com>

Actel Europe Ltd.

Maxfli Court, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom

Tel: +44 (0)1276 401450

Fax: +44 (0)1276 401590

Actel Corporation

955 East Arques Avenue
Sunnyvale, California 94086
USA

Tel: (408) 739-1010

Fax: (408) 739-1540

Actel Asia-Pacific

EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan

Tel: +81-(0)3-3445-7671

Fax: +81-(0)3-3445-7668