XILINX®

2.1i Floorplanner Support for Virtex FPGAs

XAPP402 (v1.0) October 13, 1999

Application Note

Summary/ Introduction

With the release of M2.1i, the Floorplanner will support the Virtex family of FPGAs. This application note will show you how the major Virtex-specific architectural features such as BlockRAMs, global clock buffers, DLLs, and carry logic are represented within the Floorplanner GUI and how you can manipulate a design containing these elements.

The general operation of the 2.1i Floorplanner is identical to that of the Floorplanner in the previous, 1.5i release.

Part 1: BlockRAM Representation

In Figure 1 notice how two BlockRAM elements have been selected. BlockRAMs reside on the left and right sides of the device, occupying area between the IOBs and the main device array.

Also notice that by selecting a particular BlockRAM (in this picture, the upper two BlockRAMs on the left side two are selected simultaneously), that the fanout connectivity of each RAM is displayed using ratsnest arrows. Though this is a simple design that does not tax the Virtex device's ability to handle areas with a high density of routing connections, it is worth noting that BlockRAM placement can be an area where routing congestion issues may first arise. The Floorplanner can be a useful analysis tool to identify, diagnose, and remedy such problems if they exist. A picture is sometimes worth a thousand CPU minutes!

Part 2: CLK and DLL Representation

There are four global buffers in Virtex which are often used for driving clock signals; two each at the center top and bottom edges of each device. There are also four DLL components, paired with each global buffer. Here we show the two global buffer/DLL pairs that reside on the top edge of all Virtex devices (see Figure 2). A similar pair reside on the bottom edge.

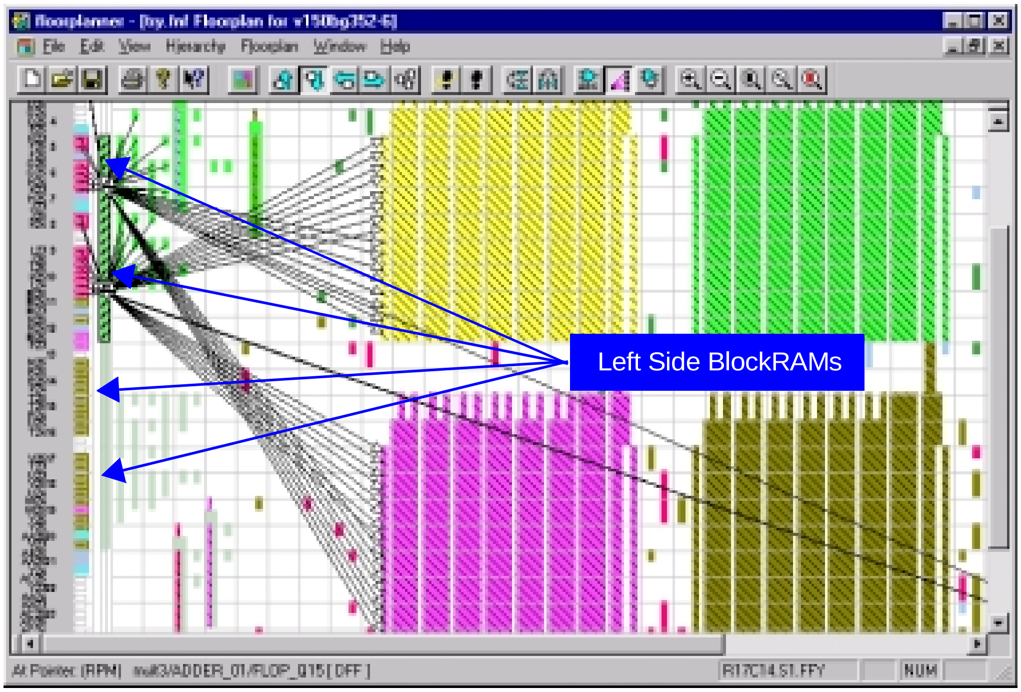

Floorplanner - [by.fnf Floorplan for v150bg352-6]
File Edit View Hierarchy Floorplan Window Help
Left Side BlockRAMs
At Pointer: (RPM) mult3/ADDER_01/FLOP_015 [DFF]
R17C14.S1.FFY
NUM

Figure 1: BlockRAM Elements

Unlike most other signal types, when a global buffer is selected signals driven by the global buffer are not shown (by default) in the ratsnest connectivity display. (It is most often the case that global buffers drive clock signals; for most floorplanning purposes having ratsnest displayed for what are typically very high fanout nets would produce excessive screen clutter.) Sometimes it is desirable to enable the display of signals driven by the global buffers. To accomplish this use the following procedure to enable ratsnest display for signals driven by global buffers:

- In the **Placement** Window (not the **Floorplan Window**; (use **Window -> Tile Compare**)) select one of the BUFG sites.

  Note: As explained above even though each of the global buffers typically drives many clock pins, there is no indication how the clock signal fans out from the buffer to its destination loads.

- Use **Edit -> Find** with the type set to *Logic Symbols* to search for the instance name of the global buffer. The design hierarchy window will automatically scroll to, and select, the buffer instance. From the line displayed, note the buffer's output signal name.
- Again using the search capability, this time with the type set to **Nets**, search for the clock signal name you found in the previous step. This will cause the **Design Nets** listing to automatically scroll to, and select, the specified net name. As this is an output of a global buffer, note that the list entry is "grayed out". Being grayed out means that the ratsnest is disabled for this signal.
- Finally, with the signal selected in the **Design Nets** window, use **View -> Options** and select the **Ratsnest** tab. In the **Selected Nets** portion of the dialog box, click on the enable button. Notice how the clock signal ratsnest (Figure 3) is now displayed in the placement window.

Though in Virtex the performance of global clock signals driven by BUFGs is not sensitive to which buffer site is chosen, sometimes there may be board-level layout requirements that could require analysis of global clock routing within the FPGA device. This is particularly true in Virtex where the DLLs are often used to mirror a clock signal that is to go off-chip.

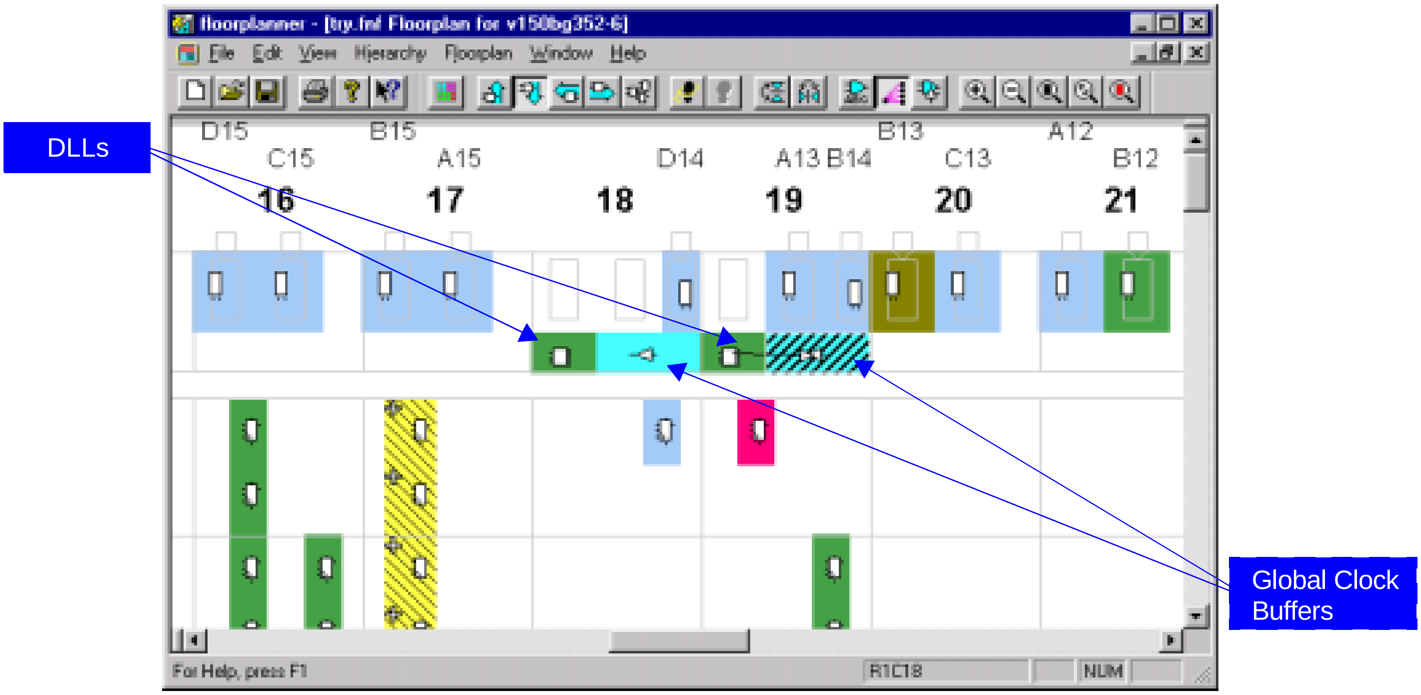
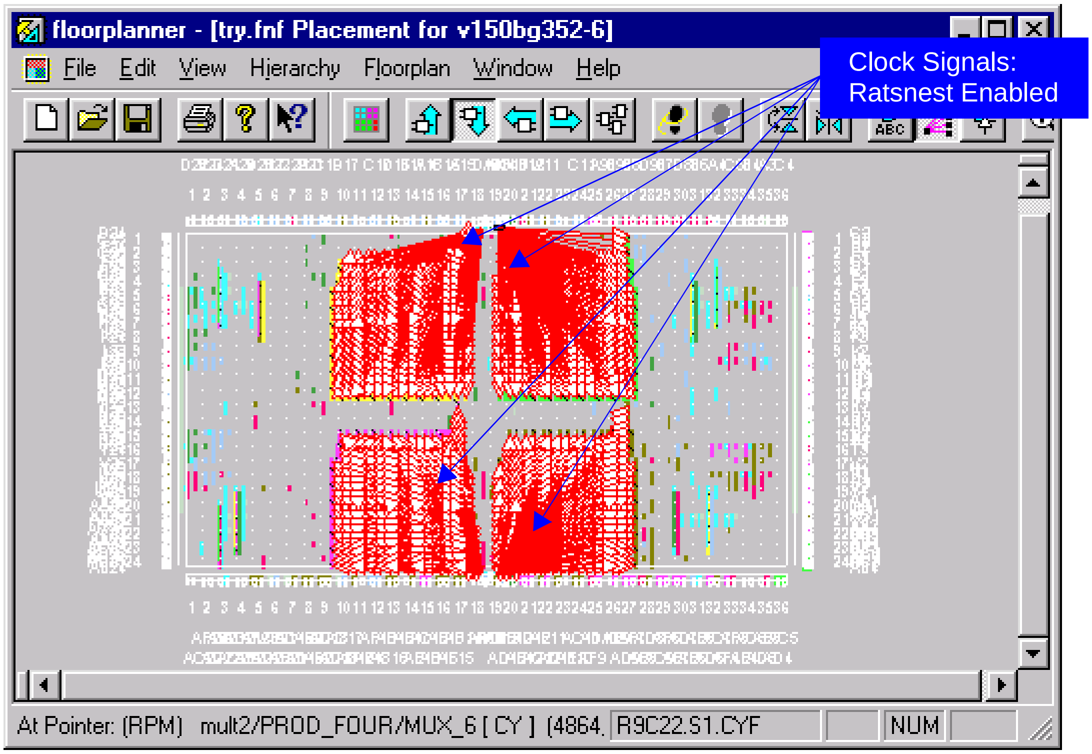

floorplanner - [try.fnf Floorplan for v150bg352-6]
File Edit View Hierarchy Floorplan Window Help
D15 C15 B15 A15 D14 A13 B14 B13 C13 A12 B12
16 17 18 19 20 21
DLLs
Global Clock Buffers
For Help, press F1
R1C18
NUM

Figure 2: Global Buffers and DLLs



Clock Signals: Ratsnest Enabled

At Pointer: (RPM) mult2/PROD_FOUR/MUX_6 [CY] (4864, R9C22.S1.CYF

Figure 3: Clock Signals Ratsnest Enabled

Part 3: Representation of Carry Logic Elements

To reduce the amount of clutter in the floorplan display, the carry logic in Virtex is represented by a simple block that encompasses the functions of the XORCY, MUXCY and MULT_AND elements. In our design the multiplier RPMs have structures that are built from elements in this configuration as seen in Figure 4.

In the, Floorplanner these discrete elements of carry logic would be represented in a single abstract block as shown in Figure 5

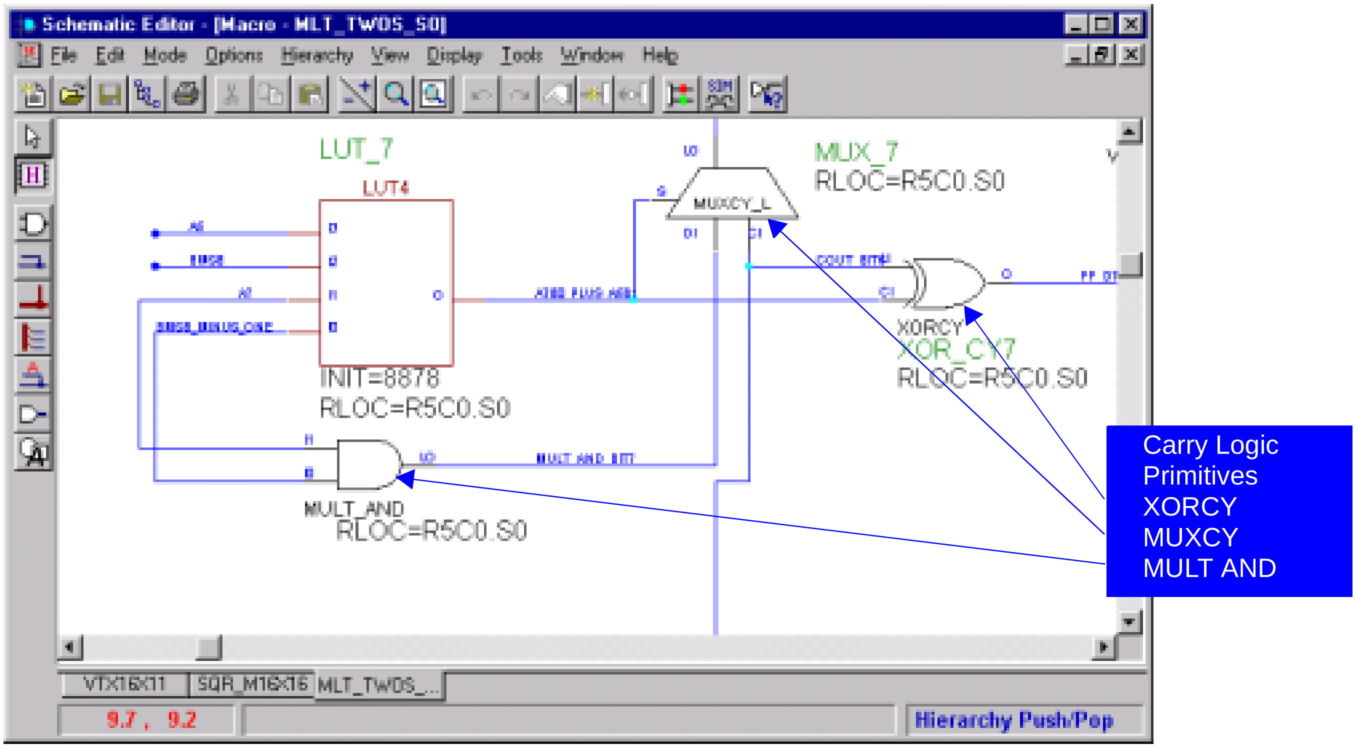


Figure 4: Schematic Editor

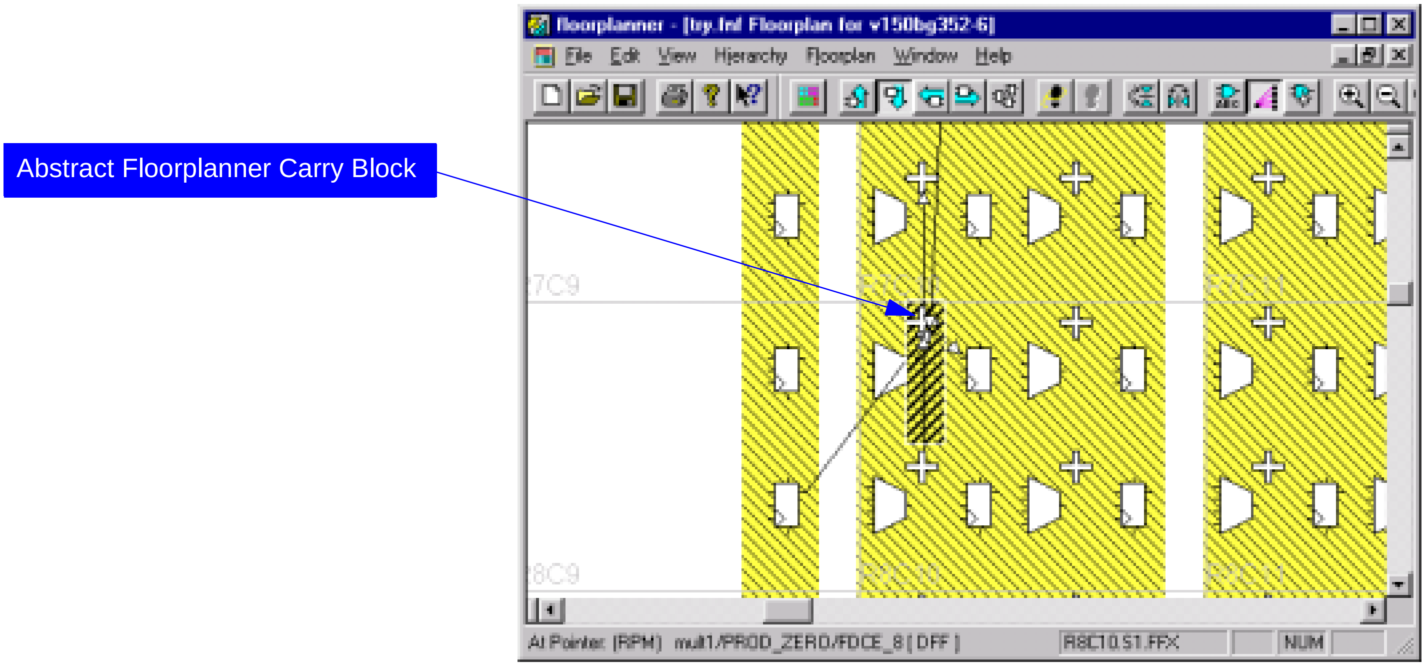

Abstract Floorplanner Carry Block
Floorplanner - [try.fnf Floorplan for v150bg352-6]
File Edit View Hierarchy Floorplan Window Help
R7C9
R8C9
At Pointer: (RPM) mult1/PROD_ZERO/FDCE_8 (DFF)
R8C10.S1.FFX
NUM

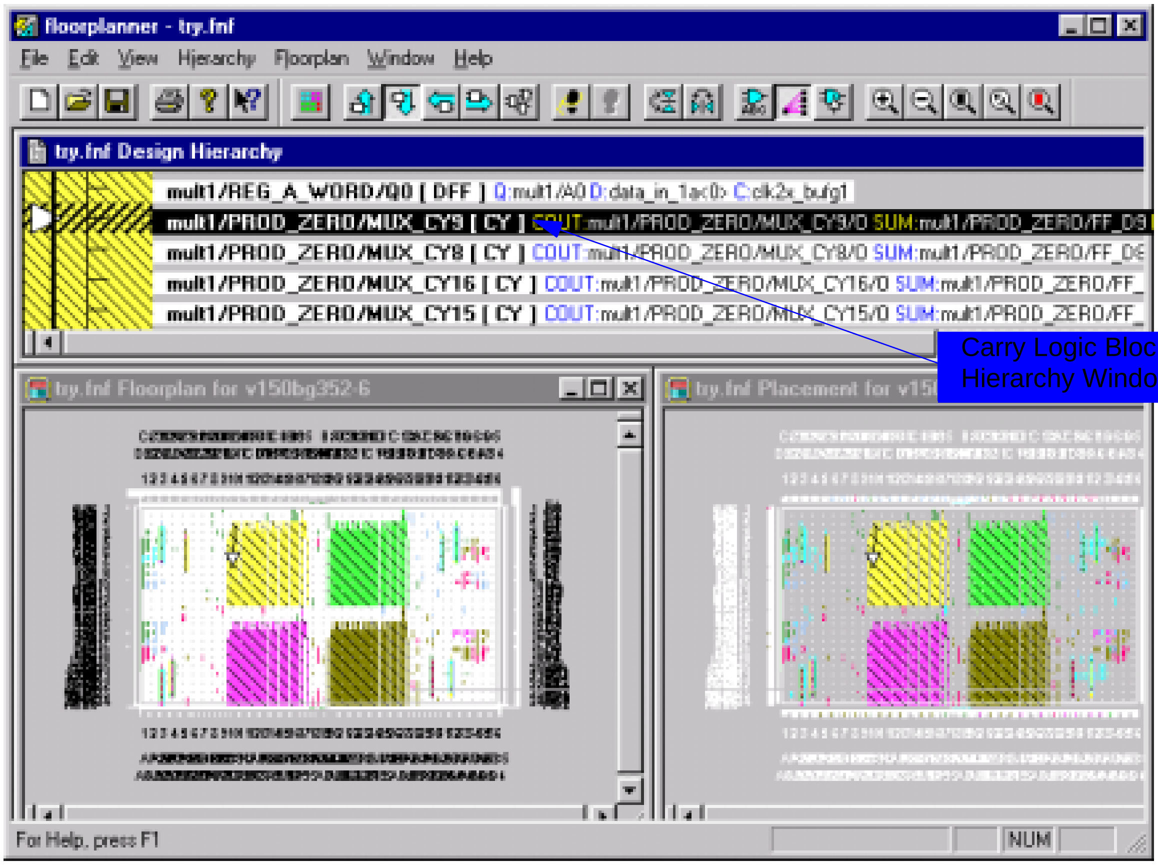
Figure 5: Floorplanner - Discrete Elements of Carry Logic

In the above example, we used the Edit -> Find function with the search type set to Carry Logic to search for mult1/PROD_ZERO/MUX_CY9, which is the logical instance name of the carry logic block selected in the above screen shot. The design hierarchy window should automatically scroll to the logic element that has the name mult1/PROD_ZERO/MUX_CY9, as shown in Figure 6.

For each pin listed for mult1/PROD_ZERO/MUX_CY9 in the hierarchy display, you can use the Table 1 to determine how the individual carry logic symbols have been mapped onto the abstract representation of the Floorplanner carry "block". The Floorplanner carry block has six possible external pins, whose functions are shown in Table 1.

Table 1: Floorplanner Carry Block Pin Assignment

Pin	Assignment
LI	Directly connected to the XORCY "0" input, optionally connected to the MUXCY "S0" input.
I1	Directly connected to the MULT_AND "1" input.
I0	Directly connected to the MULT_AND "0" input, optionally connected to the MUXCY "0" input.
CIN	Directly connected to the MUXCY "1" input and the XORCY "1" input.
COUT	Directly connected to the MUXCY output.
SUM	Directly connected to the XORCY output.


Floorplanner - try.fnf
File Edit View Hierarchy Floorplan Window Help
try.fnf Design Hierarchy
try.fnf Floorplan for v150bg352-6
try.fnf Placement for v150
For Help, press F1
NUM

mult1/REG_A_WORD/Q0 [DFF] Q:mult1/A0 D:data_in_1a<0> C:clk2x_bufg1
mult1/PROD_ZERO/MUX_CY9 [CY] COUT:mult1/PROD_ZERO/MUX_CY9/O SUM:mult1/PROD_ZERO/FF_D9
mult1/PROD_ZERO/MUX_CY8 [CY] COUT:mult1/PROD_ZERO/MUX_CY8/O SUM:mult1/PROD_ZERO/FF_D8
mult1/PROD_ZERO/MUX_CY16 [CY] COUT:mult1/PROD_ZERO/MUX_CY16/O SUM:mult1/PROD_ZERO/FF_
mult1/PROD_ZERO/MUX_CY15 [CY] COUT:mult1/PROD_ZERO/MUX_CY15/O SUM:mult1/PROD_ZERO/FF_

Carry Logic Block Selected in Design Hierarchy Window

Figure 6: Carry Logic Block Select

Revision History

Date	Version	Revision
10.13.99	1.0	Initial Xilinx release.