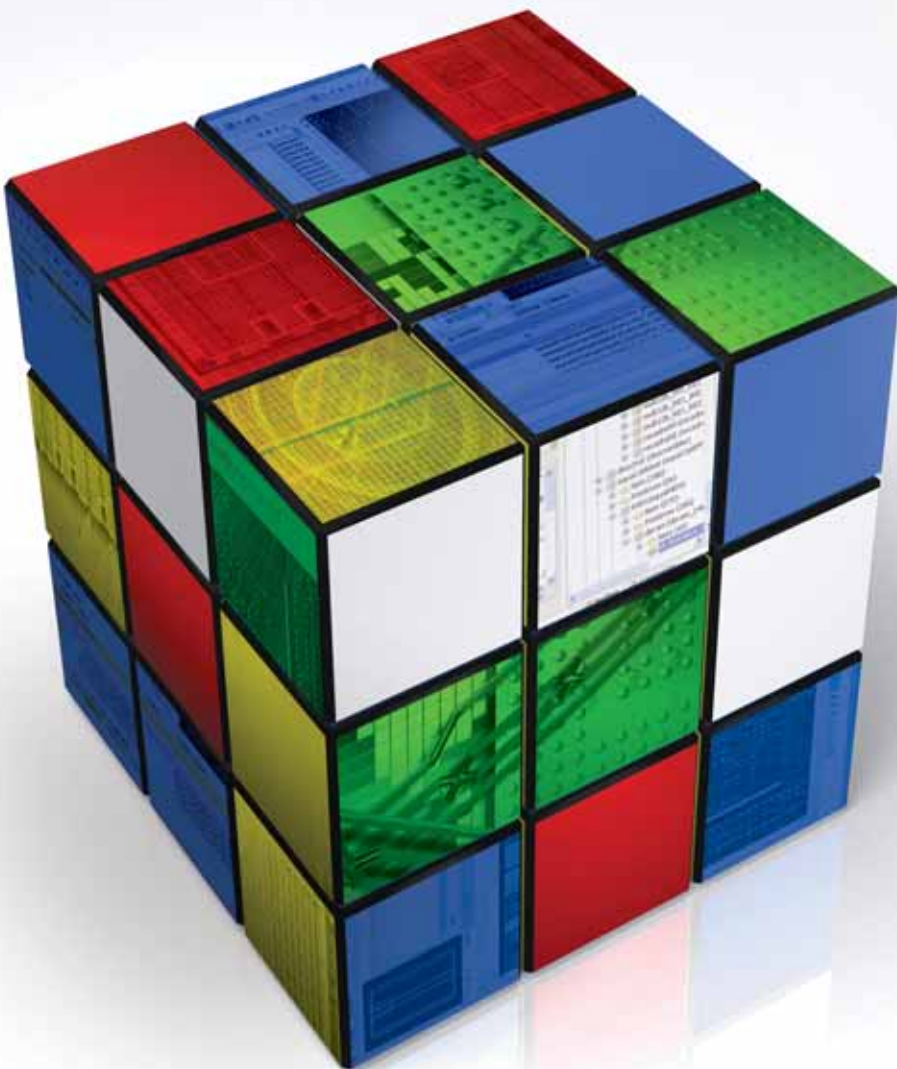


Identifying and Solving Timing Problems in Xilinx FPGA Designs

In this first edition of Ask FAE-X, the all-knowing, all-seeing FAE-X gives advice on finding and fixing timing constraint problems.

Welcome to Xcell Journal's new column, "Ask FAE-X." In each issue going forward, one of our many field application engineers worldwide will assume the identity of the FAE-X and answer one of your questions or a question that users commonly ask our field support team.

If you have a question that you would like the FAE-X to answer in Xcell, send your question to Ask FAE-X's minion, mike.santarini@xilinx.com.



by Chris Dunlap
Field Application Engineer
Xilinx, Inc.

It is heartbreaking to spend months on a design and then not meet your timing constraints. It can be almost as heartbreaking trying to figure out how to properly constrain the design to make sure that your design meets timing. Finding and fixing timing constraints is a common yet very tricky problem.

The frustrating thing about timing problems is that no one single cure solves every type of timing problem. The good news is that we can solve most timing problems by pushing the limits of the software, optimizing RTL code, or a little of both.

Because customers are usually very wary of sharing source code with FAEs, we generally try to solve their timing problems by helping them push the limits of their tools.

But before I get into that, the first thing we need to do is a basic analysis of the timing problem. The goal here is to root out anything obvious such as a clock pin at the top of the device, driving a digital clock

manager (DCM) at the bottom of the device, which in turn drives a global buffer (BUFG) at the top of the device.

Sometimes you can place pins in such a way that makes it impossible to meet timing. Usually you can discover obvious timing problems by inspecting the delays in the timing report. In any case, you will need to use a floorplanner to solve obvious problems by locking down the offending instance(s) to proper locations. A floorplanner will also help to visually understand the timing problem.

Use Current Software Tools

Assuming that the problem isn't an obvious one, to zero in on the cause we need to understand which device family and software version you are using. In general, each device family tends to have a single optimal software release. The best software for Xilinx® Virtex™-4 devices is ISE™ software version 9.2i, while the best software to date for Virtex-5 FPGAs is ISE software 10.1. (For this month's column, I used ISE software v10.1 and PlanAhead™ software v10.1.)

Synthesis tool versions matter as well, so when using the latest device architectures, it is really important to make sure that you have downloaded and are using the latest software builds. Software development almost always lags behind the hardware capabilities, so using old software with new device families is not something I advocate.

Unfortunately, some customers shy away from upgrading software for fear of dealing with new or unknown bugs. However, if you want to deal with timing challenges properly when using the latest device family, I strongly recommend downloading the latest software.

Once you have the best software release for the device family you are targeting for your design, you need to find the best set of implementation options. Unfortunately, there is no such thing as a magic set of options. There are thousands of different combinations that you can apply to implementation tools. The timing score – that is, the summation of total failed timing paths in picoseconds – will vary greatly depending on what implementation options you use.

Xilinx has several tools to help you find the best set of options for your design. ISE software now includes two tools: Xplorer, and more recently, SmartXplorer. SmartXplorer can take advantage of multiple processors to run multiple implementations with different options.

SmartXplorer requires Linux but is very easy to use. The command line is simple: `smartxplorer designname.edn -p xc5v1x110t-1ff1136`.

As long as the user constraints file (UCF) and netlist constraints file (NCF) have the same name, SmartXplorer will use the correct options automatically. The only other thing you'll need is the host list file.

cantly improves performance. Good constraints in synthesis can also make a difference in meeting timing.

PlanAhead Software

After you have played around with implementation options to come up with the best timing score, you can now start to effectively analyze the timing problem. PlanAhead software is a very valuable tool at this point because it allows you to visualize the post place-and-route design. You can also import the timing constraints and cross-probe the failed path to the placed component window.

When the tools make bad decisions, you can correct them by floorplanning modules

Options	Timing Score	Run Time
Map : -timing -ol high -xe n -global_opt on -retiming on	651262	6h 26m
Par : -ol high		
Map : -timing -ol high -xe n	0	4h 25m
Par : -ol high		
Map : -cm area	380834	5h 19m
Par : -ol high -xe n		
Map : -timing -ol high -xe n -register_duplication -logic_opt on	0	5h 46m
Par : -ol high -xe n		
Map : -timing -ol high -xe n -global_opt on -retiming on -ignore_keep_hierarchy	666567	12h 34m
Par : -ol high -xe n		
Map : -cm balanced	172171	4h 4m
Par : -ol high		
Map : -timing -ol high -xe n -pr b -cm balanced	7235	6h 33m
Par : -ol high		

Table 1 – A SmartXplorer 10.1 example on Virtex-5 FPGA-based design

SmartXplorer can log in to other machines through secure shell/remote shell (SSH/rsh). Just add each machine name on a separate line in a file called `smartxplorer.hostlist`. If a machine has two processors, list that machine twice. Table 1 shows a set of results from SmartXplorer.

PlanAhead software also includes a feature similar to SmartXplorer called ExploreAhead, which supports distributed processing of place-and-route jobs across multiple Linux machines at the same time. The point of all of these tools is similar: to find a set of options in the implementation tools that will get the best timing score.

Notice how the options can have a drastic effect on the timing score and run times. It is also important to play with synthesis options. For instance, turning off hierarchy in synthesis often signifi-

or hand-placing components. This can be – and often is – an iterative task that requires multiple runs to find the best way to floorplan the design to optimize timing. PlanAhead software's visual capabilities really do make that job much easier.

Using PlanAhead software, first create a project, importing either the HDL or netlist files into the tool. Once you've created a project, choose **File** → **Import Placement**. Select the post PAR (ncd) file from the best run to import the placement into the PlanAhead software project.

The tool will organize the PlanAhead software project into several windows. The top left window is the physical hierarchy, which describes the current area groups in the design. The selection window is beneath that and contains details on whatever data you have currently selected. The middle

Timing problems will vary, and the solutions to tackle them in PlanAhead software will also vary. Solving timing problems inside PlanAhead design tools requires practice.

window is the netlist window, which outlines the full netlist hierarchy. The far-right-side window is the Device view, which will now be populated with logic you've implemented in your design.

Next, import the timing analyzer report (TWR/TWX) into the tool. Choose **File** → **Import TRCE Report**. This step will add the timing report data in the bottom window. You should sort this window by worst-case timing slack to focus on the worst-case offenders first. Fixing the worst-case offenders will often solve timing problems in the whole design.

Once you select a failing path, PlanAhead software will select the instances and the connections in that failed path. Hit **F9** to zoom into that selection (see Figure 1).

This is where things might get a bit complicated. You must do some amount of inspection to understand why the place-and-route tool placed the primitives the way it did, and whether there is a better method for your particular design.

You can zoom into any of the highlighted primitives. Left-click a primitive and move it around to understand the connectivity. In the netlist window, you can also move up to the top-level block, which has the selected instance in it. Right-click the block name and choose a color for those selected instances. This will now show how this block was placed in the chip and if the grouping is close together or spread out.

Sometimes you will see that you need to lock down particular primitives better. Block RAM and DSP auto-block placements tend to be common problems in timing failures. In Figure 1, it is easy to see that PAR placed the Mult18 in a non-optimal location. The failing path has a block RAM, which feeds a Mult18, which in turn feeds a carry chain. The block RAM is high, the Mult18 is low, and the logic is high. If the routing didn't have to go up and down as well as sideways, it would have met timing.

Timing problems will vary, and the solutions to tackle them in PlanAhead software will also vary. Solving timing problems inside PlanAhead design tools requires practice. Let's tackle the timing problem in Figure 1 as an example. Following these steps will lead us to a solution:

1. Select the failed timing path so that the essential path is highlighted.
2. Right-click on one of the instances and choose **Highlight With** → color of choice.
3. In the left column, unplace the Mult18 that is second from the bottom. Right-click **Unplace**. This will open up room for the block RAM that is failing timing.
4. Move the Mult18 on the bottom left up one location by clicking and dragging.
5. Click and drag the block RAM on the right to the free block RAM location at the bottom left.
6. Select the failed timing path and verify that the path looks optimal (see Figure 2).
7. Choose **Tools** → **Clear Placement Constraints**. Click **Next** through the first option.
8. Select **Unplace All But Selected Instances**. Click **Next** through the remaining wizard.
9. If you wish to run implementation outside of PlanAhead software, choose **File** → **Export Floorplan**.
10. The tool will write out a new UCF called top.ucf. You can either use that or take the constraints in the file and add them to your original UCF.
11. Another option is to run implementation inside PlanAhead software.

Choose **Tool** → **Run ISE Place & Route** with ExploreAhead.

ExploreAhead has a couple of nice features, including:

- a) Automatic importing of constraints from the Device view to the UCF.
- b) Simplified importing of placement and timing results from runs implemented inside ExploreAhead.

Pblocks and Floorplanning

If you find a lot of timing failures in your layout, hand-placing typically won't solve these timing issues. Instead, you should create area groups.

One of several ways to create an area group (Pblock) is to right-click the block name in the Netlist window and choose **Draw Pblock**. Then draw a rectangle in the Device view where you would like to place that area group.

The tool will create a Pblock, and you'll be able to see a lot of information about it. The properties window will display the logic resources the Pblock requires and the resources available in the drawn group.

It can be quite an iterative process to floorplan a design. You'll find that you may have to do several runs to tune the area groups before they start working well. Also remember that less is sometimes more. You do not always need to area group the entire block. Focus on the failing primitives; you can group those and floorplan them alone.

You should use metrics on Pblocks to better understand functionality, such as how well-utilized the configurable logic blocks (CLBs) are in that area group (to see the available metrics, click on the **Metrics** tab in the top left window of PlanAhead software). This can help you determine if a lot of congestion exists in a particular area that is inhibiting routing. If there is a lot of congestion, you'll need to spread the logic out through floorplanning.

Once you've used PlanAhead software to squeeze out the best timing score, the last remaining task is to optimize the code. PlanAhead design tools now support HDL source. You can cross-probe timing problems back to the netlist or HDL, depending on what sources you imported.

Using the Schematic view, you can view the entire timing path. Just select the timing path from the timing results and hit F4. Watch out for logic that fans in and out of other modules. The tools must often spread out modules over the chip because of external interface requirements. If this happens,

you'll probably need to use pipelining in your design.

Block RAM or DSP output timing will commonly contribute to timing failures. You can often recover a full nanosecond or more by registering the output of these blocks.

Final Recommendations

If you are having problems with timing closure and you don't have PlanAhead software, I strongly recommend that you get it. This tool allows you to manage every aspect of solving a timing problem.

Using PlanAhead design tools, you can run ExploreAhead to find the best set of options. And because PlanAhead software is not tied to a particular ISE software release, you can also try different ISE software versions if you have multiple versions installed on your workstation. PlanAhead software's cross-probing features allow you to fully understand both the source and implementation of a design.

Timing closure can be a pretty tricky problem, but with the right versions of the right tools and a comprehensive timing closure plan, you should be able to locate timing problems in your designs quickly and get the job done. And with that, FAE-X bids you happy designing. 🎯

Chris Dunlap is an FAE in the Xilinx Santa Clara sales office in California. He originally joined Xilinx in 1999, working in the technical support group.

What Chris enjoys most about his job as an FAE is helping customers in need. He enjoys the diversity of his position, as well as the variety of challenges he faces on a daily basis. In his spare time, he lets off steam on the ice, playing hockey. At home, he loves to spend time with his wife and son, while anxiously awaiting a newborn daughter.

If you need assistance, be sure to check in with your local FAE, contact Xilinx tech support at (800) 255-7778, or visit www.xilinx.com/support/clearexpress/web-support.htm.

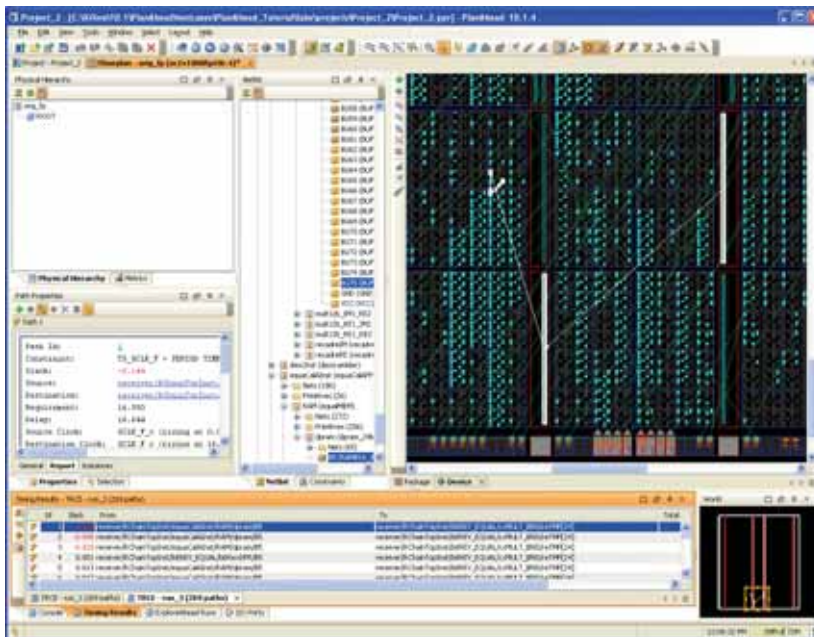


Figure 1 – PlanAhead software 10.1 shows an implemented run with a failed timing path highlighted

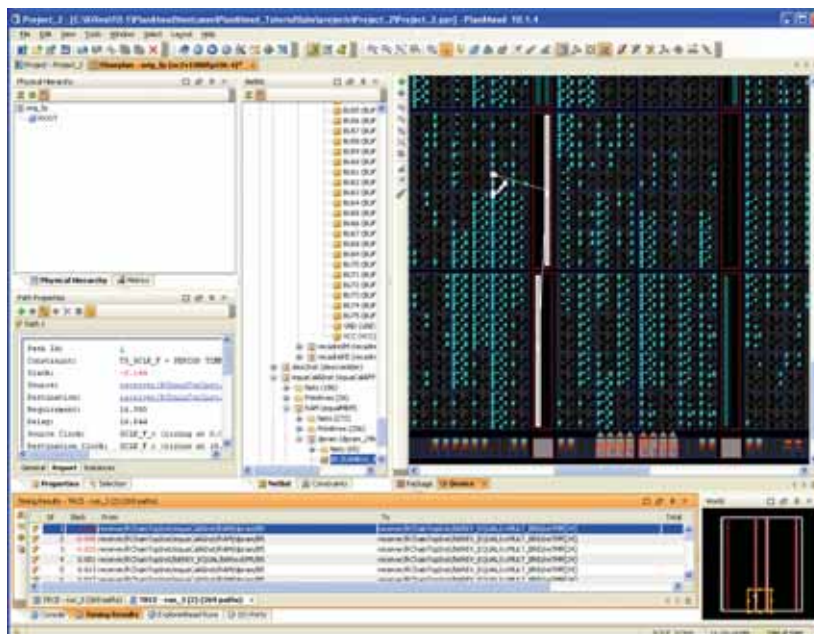


Figure 2 – PlanAhead software 10.1 shows a corrected path after locking down DSP48 and block RAM