

Board Design Panacea

The 7Circuits tool algorithmically solves FPGA pinout problems and synthesizes PC board schematics.

by Nagesh Gupta
Founder/CEO
Taray, Inc.
nagesh@tarayinc.com

PC board design is a cumbersome and time-consuming task. Although some of the steps require knowledge and intelligence to complete, most of the process is mundane and routine. Add FPGAs to the mix, and the complexity of the board grows significantly. FPGAs have a myriad of complex I/O rules that are multi-dimensional and can present difficult problems:

1. In most cases with large and complex designs, FPGA pinouts are hardly optimal, and non-optimal pinouts result in lower design performance. The cost of the PC board also increases because of the higher number of layers.
2. Today, pins for FPGAs are mostly selected manually. The pin selection is aided by large spreadsheets with signal names, I/O standards, clocking types, interface, and so on.
3. Drawing schematics is a fully manual process. The FPGA symbol has to be created, and then the FPGA pins have to be connected up to the interface pins. To avoid expensive mistakes, all of the pins have to be correctly connected. The configuration and power supply pins have to be connected as well.

Taray, which brought you the Xilinx® Memory Interface Generator, has developed a new tool called 7Circuits. 7Circuits solves these problems in an innovative way.

7Circuits

7Circuits is a highly intuitive tool that not only selects all of the FPGA pins but also generates PC board schematics for the FPGA and its interfaces.

7Circuits solves FPGA pin allocation problems algorithmically after considering the different constraints. At a higher level, the constraints that the tool considers are:

- Physical constraints. An example of a physical constraint is the physical placement of the FPGA and the interfaces on the PC board.

- Electrical constraints. I/O voltage levels, use of DCI termination, and I/O signaling standards form the electrical constraints.
- Logical constraints. The logical constraints are derived from the interface protocol. For example, if the FPGA is interfacing to a DDR2 memory, the DDR2 protocol will dictate the logical constraints of the interface.
- User preferences. You can tune the performance features of 7Circuits to achieve optimal results.
- FPGA. The location, type, and number of I/Os are among some of the parameters considered.

7Circuits comes with a board view on startup. You begin by placing the FPGA on the board. Next, you place the different components with which the FPGA interfaces. The FPGA and all of the components are shown to scale. The components should be located correctly with respect to the FPGA and the placement should be identical to the actual board placement. An example of the component and FPGA placement is shown in Figure 1.

7Circuits supports a large blend of standard components that you can select and place on the board. If a particular component is not already supported, 7Circuits provides a simple user interface to create the custom interface (alternately, Taray can help you create the interface). Defining the interface component correctly is key to the generation of correct outputs.

7Circuits can block off the pins selected outside the tool. Reading a UCF file with the pin location constraints supports this functionality. 7Circuits can also generate interfaces incrementally. In other words, you can open a saved project and add more interfaces to it without disturbing the existing connections.

If you want to use specific banks for certain interfaces, you can make 7Circuits do

it. You can also specify the percentage of pins to be used within each bank. This enables 7Circuits to be customized for any requirement.



Figure 1 – Placement of the FPGA and interface components on the board

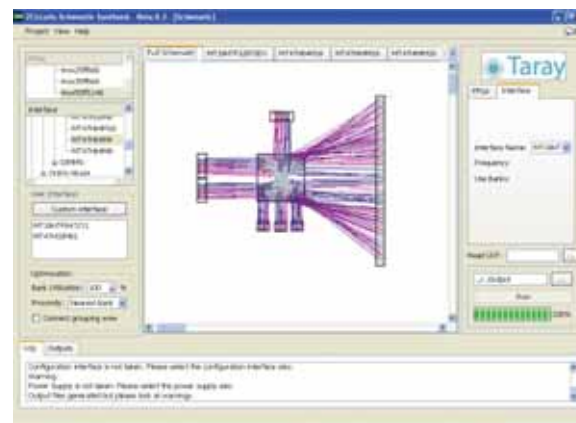


Figure 2 – A ratsnest view of the connections determined by 7Circuits

7Circuits goes through multiple optimization phases to select the pins optimally. After running through different optimization phases, 7Circuits displays the ratsnest connections to enable you to view any bowtie effects. Such interactive output at this stage is a key enabler to optimal results. You can try out different placements or different optimization options within 7Circuits to improve the bowtie effects. An example of the ratsnest is shown in Figure 2.

7Circuits produces a UCF file for pin locations; an EDIF schematics file for the FPGA, interface symbols, and schematics; and a top-level RTL file with all interface port declarations.

Key Advantages

7Circuits produces results with a holistic understanding of the problem space. This makes 7Circuits the first tool to bring system-level understanding into the FPGA solution. By doing so, 7Circuits comes up with the most optimal solution for pinout.

7Circuits reduces the time it takes to create an FPGA-based board from weeks to hours. The pinouts are very dependant on placement. In the current mode of operation, you do not have the luxury of trying out different placements to optimize results. Each placement and generation of the corresponding pinouts is at least a three-man-week task. This makes it impossible for you to try out various placements. With 7Circuits, you can try out four to five different placements and decide on the best placement within a few hours.

7Circuits offers you the added benefit of generating schematics for all of the mundane connections automatically. This task not only saves time, but also ensures correctness.

Here are some of the key advantages of using 7Circuits:

- 7Circuits connects all of the interface pins correctly. In addition, it connects up the power supplies to the right voltage levels.
- It connects Vref pins to the correct voltage levels depending on the I/O standard used.
- It reserves Vrp/Vrn pins when DCI is used. If DCI is used, the Vrp/Vrn pins are connected to the appropriate voltage levels.
- All configuration modes such as JTAG, slave serial, and master serial are supported. The connections are made automatically.

Because most of the mistakes are made in the unexciting and routine connections, the schematics are of a great benefit. They save greater than three man weeks of time and, more importantly, ensure correctness.

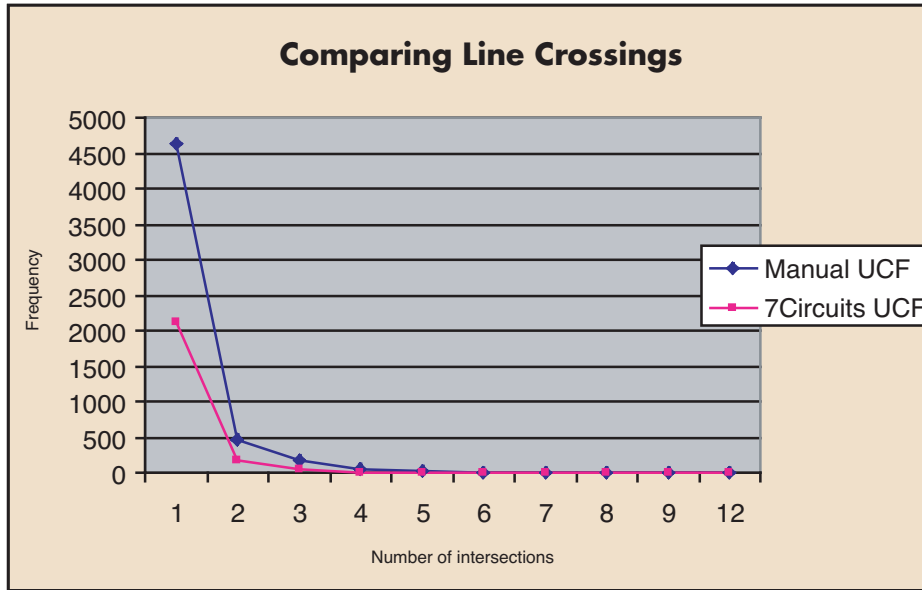


Figure 3 – Bowtie effects are significantly reduced, thus simplifying layout and reducing PCB layers.

Figure 3 shows an analytical comparison of the results for a memory reference board. The board has a Xilinx FPGA and interfaces with two DDR2 SDRAM DIMMs. This makes a 144-bit-wide interface. It also interfaces with DDR2 components to make a 24-bit-wide interface. The figure charts the frequency of line crossings against the number of line crossings. These comparisons clearly show the efficiency of the tool:

1. The original number of line crossings was 5,337. The line crossings with 7Circuits were reduced to 2,339 – a reduction of more than 50%.
2. There are 4,600 lines that cross each other manually. With 7Circuits, only 2,050 lines cross each other (1 point crossing each other).

Technology

The key to producing effective results is in the algorithms and the technology behind the tool. 7Circuits uses patent-pending technology to solve the issues identified in this article. Here are some of the key innovations in 7Circuits:

- Identifying and representing information. 7Circuits requires physical as well as architectural information on every interface and protocol. All of this information has been precisely identified for the components already supported. For new components, the tool provides a simple and intuitive GUI for you to give this information.
- Special signals are correctly identified and represented so that these signals can be associated to special pins. One example is the Xilinx RocketIO™ pins.
- 7Circuits also considers the logical and architectural aspects. Pins that are logically related will be placed together. This ensures quicker design convergence through the synthesis and PAR phases.
- 7Circuits constantly monitors the number of wire crossings and minimizes them, minimizing the number of board layers. This is key to reducing manufacturing costs.

- Length matching. Various heuristic algorithms are applied to reduce the delta length of signals that are to be length-matched. Applying these algorithms early on avoids long traces on the board. This improves signal quality and enables the PC board router to converge faster.

Results

7Circuits has been going through beta trials since Fourth Quarter 2005. Some of our customers have successfully laid out the board using our outputs.

Additionally, we have tested our results with many Xilinx reference designs. Our test process is as follows:

1. Generate a design for the same interfaces as the standard Xilinx reference board using 7Circuits.
2. Compare the ratsnest of the reference design against the ratsnest from the tool. In all cases, we found that 7Circuits produced a lower bowtie than the reference design.
3. Use the UCF generated by the tool and go through synthesis, build, map, PAR, and bitgen. Ensure that timing results from 7Circuits' UCF meet the reference design requirements.

Conclusion

Taray is committed to ensuring your success through the use of 7Circuits. Having created the Memory Interface Generator for Xilinx FPGAs, Taray's engineers have the depth of experience required to understand the issues facing you.

We are planning rich feature sets for future releases of 7Circuits, including:

- Schematics. 7Circuits will generate Orcad and DxDesigner schematics natively.
- Symbols. 7Circuits will be able to use symbols from your symbol library. Additionally, 7Circuits will also be able to use fractured (split) symbols to ensure that the schematics are consistent with your company standards.
- Parts. 7Circuits will support other Xilinx FPGA families and support more interface components.
- 7Circuits will offer a verification mode. This will be a great feature for you to check that your files are consistent and that your choices are optimal. You will be able to make incremental changes to improve your results.

A demo version of the 7Circuits tool is available at www.tarayinc.com. Revision 1.0 will be released in Second Quarter 2006. 🌈