Speeding Technology Development Through Improved Processes for Test Structure Generation

Abstract: The cost, complexity and implementation time associated with parametric test structure creation, testing and data management make this area a rich opportunity for improved processes for mixed analog/digital technology development. This paper explores the specific improvements in parametric test chip creation implemented by a leading edge analog IDM in the development of a TSMC-compatible high-voltage mixed analog/digital process through the use of Stone Pillar TestChipBuilder.

Introduction: The high cost and complexity of parametric test chip generation

Wireless and other mobile technologies are dependent on low cost in order to encourage mass adoption. This low cost is often best achieved by including analog or power components on the same chip with digital components, reducing part count and size. However, the optimization of analog applications within a digital process brings new challenges, including the optimization of individual device performance. Optimizing device performance can be accomplished, in part, through a combination of intuition based on previous device technologies and TCAD simulation; but a typical technology development cycle makes heavy use of test structure fabrication and characterization, with variations in the test structure to try to locate an optimum for one or more desired operating points.

Analog structures typically encompass a large number of design rules or device geometric parameters that can be varied in order to optimize performance. But the exploration of the full combinatorial set of parameters results in the requirement for test chips that encompass dozens to as many as thousands of variations of a single device type. In order to make full use of the structure variations, a targeted set of devices must be selected, layouts generated (oftentimes by hand) and documented, and the resulting structures tested. The test results must be analyzed in order to find one or more devices to target for use in designs.

The time pressures in bringing new processes to market are immense. Because of these pressures, often the limiting factor is not mask space but layout time, resulting in a difficult choice: Limit the number of test devices to be placed on a test chip (and quickly fill in available mask space by duplicating a limited set of devices) or delay the creation of the test structures while the layout for additional or revised device variations are created.

The Goal: Faster, more efficient creation of parametric test structures

The customer described in this paper is a leader in data conversion and signal conditioning technology. Their innovative products combine high power capabilities with digital processes for mixed signal applications such as backlight drivers. Such processes, which need to produce high power devices within the constraints of a digital process, are particularly challenging in terms of device design, often requiring hundreds of test devices to qualify a particular device type.

In planning for an advanced, TSMC-compatible, high power process, technology development engineers at the

Stone Pillar customer recognized an opportunity in the creation and management of test structures for technology development. If the creation of devices could be automated in a way that permitted the test structures to be generated automatically based on a set of desired device parameter values, the work of creating the test chip could be done in parallel with selecting the parameter set of interest and the exact device parameters could be included in the test chip generation at the last minute. With this flexibility, coupled with the ability to automatically generate large numbers of device variations, more devices could be included in the test chip, and valuable mask space wouldn't need to be compromised in the interest of speed. To achieve these goals, the customer adopted the use of Stone Pillar TestChipBuilder, one component of Stone Pillar Suite, a complete technology development software solution.

They identified several areas which they targeted for improvement by automating their test structure generation:

* Accelerate the creation of test structures: The early stage test chips for a high voltage analog/mixed signal process often require hundreds of device variations. Often, such test chips are not constrained by mask real estate so much as the time required to create the device instances. This time pressure can sometimes force the filling of mask area with duplicate device structures because there isn't enough time between when the device specifications are available and when the mask information needs to be delivered.

* Speed the placement and routing of devices: Once device instances have been created, the process of placing these into a pad ring and routing them is also time consuming and prone to errors.

* Reduce errors: Errors in test structure layout or documentation can be a source of delay or confusion. The transition from layout to subsequent testing requires accurate documentation that can be time-consuming to produce and can be the source of errors. Improving the documentation makes for a seamless handoff from test chip to test engineer. Hunting for the source of errors absorbs critical resources that would otherwise be dedicated to analyzing the results produced by the test chip.

* Improve work flow: The customer wanted to find a way to allow the process of creating the test devices to be performed in parallel with selecting the final device parameters and characteristics. This flexibility would effectively shorten the development cycle by allowing last minute changes to device parameters.

The Solution: An automated system for test structure generation

Stone Pillar TestChipBuilder is built around a parameterized representation of the device structure that enables the automatic creation of numerous device variations. When the values for the parameters are specified (either on a single device, or globally across all devices) the parameterized device definition is used to create one or more device instances. These device instances are then placed within a customized pad ring and routed according to a customized routing strategy that might include common terminals or various strategies to minimize resistance or capacitance at one or more terminals or to create matching terminal resistance.

Thousands of devices can be specified or modified in minutes, all systematically placed and routed within a selected pad configuration, eliminating the manual placement of devices that was a source of errors. The device parameter specification can be performed either within the GUI of TestChipBuilder or through edits within a spreadsheet. The systematic approach enabled by TestChipBuilder and the automatically produced documentation make the process of test creation easier and less error-prone. The final output from TestChipBuilder is the complete test chip documentation and GDS that can be loaded into whatever tool flow is used for reticle creation.

The automatic generation of devices by TestChipBuilder means that the real work of creating the parameterized devices can be done before the device dimensions have been determined. By creating the parameterized device definitions early, it is possible to move from the desired parameter set to a completed test chip rapidly, making it easy to accommodate last minute changes to design rules or device parameters or attributes. Once the parameterized device definition has been created and the parameters are provided, TestChipBuilder crunches through the variations to produce the targeted test chip in a matter of minutes.

Another advantage of this approach is the re-usability that stems from parameter independence. Parameter independence makes it possible to use a single parameterized device across a range of operating voltages or technology nodes.

The scripting and debugging capabilities of TestChipBuilder use javascript. But by using the included support for object-oriented (OO) construction and re-use, a variety of device structures can be built from one master template. The integrated visual debugging capabilities in TestChipBuilder make it easy to ascertain that a device has been correctly implemented. Because of the OO support in TestChipBuilder, extending the device template to add, for example, a second drain contact, would require minimal programming, and could be done expeditiously.

The output from TestChipBuilder provides full documentation of the final test chip, including all device dimensions and terminal connections. Error-prone manual placement and routing are replaced by automatic and consistent routing using user-configurable strategies. Because the developer of the parameterized device in TestChipBuilder is often the same person who is familiar with the desired device characteristics, another source of errors, the handoff between technology development and CAD support, is also eliminated.

A process development engineer, described their use of Stone Pillar TestChipBuilder: "Using TestChipBuilder, we were able to create parameterized device definitions before the parameter values had been finalized. Once the values were set, the creation of the test chip was simply a matter of filling in the final parameters and letting TestChipBuilder instance the devices, place and route them in our pad ring, and document the whole thing. With this capability, last minute changes to any of the design rules meant only a regeneration of the test chip, a matter of some number crunching on a PC, rather than a complex reimplementation.”

In the same way, if designers want to include a larger device in the test chip, it is simply a matter of changing a

few parameters and including the device. Now throwing out devices that end up at the bottom of the priority queue doesn't mean discarding days of implementation effort. It is now simply a re-shuffle of the selected devices that were all derived from the same master device definition or template.

“Device variations weren't limited to parameter values,” explained the user of TestChipBuilder. “With the flexibility provided by TestChipBuilder, devices could be varied based on boolean attributes such as multiple or single gate stripes, device rotation and mirroring, or guard ring structures.”

This flexibility makes it possible to create a complete set of modeling devices with a particular parameter set as a block and then create variations of this device set for experimentation for issues such as misalignment or “corner lot” attributes as a way to explore the robustness of the devices. “With TestChipBuilder, I'm able to set up a group of devices that includes all the corners needed for modeling. Creating variations on this set to explore the design space is as simple as changing parameters and regenerating the complete set of devices,” said one user.

One example of device optimization is engineering the tradeoff between breakdown voltage and R-on. Finding the desired optimum in this tradeoff, and providing designers with the right combination of possibilities involves testing a tremendous number of device instances. Using TestChipBuilder, engineers were able to create the needed variations, and include devices that might otherwise be difficult to include such as those with large numbers of stripes. A new device variation such as this might have been dropped from the test chip in the old methodology just due to the sheer effort of laying it out. Using TestChipBuilder, such devices could be added “on-demand” based on the latest design requirements or experimental applications.

The improved capabilities provided by the new methodology have enabled creation of test structures for new analyses such as alignment tolerance studies. By including devices with skewed layer positioning, creating intentional misalignment in the device structures, the customer is now able to readily explore the effects of possible misalignment during production. Such structures provide a prediction of the manufacturability of the devices and their sensitivities in foundry environments where the alignment tolerances might vary from those in their own facilities. This analysis would have been highly impractical prior to using TestChipBuilder.

Conclusions

By automating the generation of device structures for parametric test chip formation, the customer was able to reduce the engineering cost of test chip creation and allow technology developers to determine the final parameters for inclusion in the test structures at the last stage of test chip development. The increased flexibility resulted in better utilization of the available mask area and reduced the number of pilot runs for qualification of new processes by a complete iteration.

The improved processes produced a reduction in errors and reduced the effort involved with test chip debugging virtually to zero. In addition, the automatically produced documentation, and the guaranteed consistency with the implementation that it provided, reduced the effort of subsequent device testing and allowed test planning to be

started at an earlier stage.

Since deploying TestChipBuilder, they have been able to run a full set of devices, using all of the available mask area for unique device configurations, and have done this within the short development timetable. They ended up gaining another benefit in addition to the time savings and increased flexibility. "The consistency we achieved saved immensely in error reduction,” said one of the users of TestChipBuilder. "The ability to fully utilize the mask area with the best possible parameter set probably saved us a complete mask spin."