Certify™: RTL Multichip Partitioning and Synthesis for ASIC Prototyping Using FPGAs

Sunil Ashtaputre, Technical Marketing Director, Synplicity, Inc.

Overview

Verification is the biggest bottleneck for today's complex ASIC designs. Continually advancing process technology, and the corresponding explosion in design size and complexity, have lead to verification problems that are impossible to solve using traditional software simulation tools. In addition, an increasing number of applications that process large quantities of data in real time (such as video) require verification techniques that run at near real-time speeds. As a result, a growing number of ASIC designers are building prototype boards using FPGAs to verify their ASIC designs. Another advantage of building prototype boards using FPGAs is that it allows software development and debugging to happen in parallel with the design and debug of the final ASIC, reducing time to market.

ASIC designers face several challenges in prototyping and debugging the design using FPGAs. One of the biggest challenges is that the capacity of even the largest FPGAs is much smaller than the size of a complex ASIC. This means that designers must struggle to partition their design into multiple FPGAs, with no tools to help them make good partitioning decisions, and no way to model the characteristics of the prototype board during synthesis and partitioning. Iterations between synthesis, partitioning, and board implementation are time-consuming and tedious.

Certify is a new tool developed by Synplicity® to address the challenges faced by designers building FPGA-based ASIC prototypes. Certify combines RTL multi-chip partitioning with best-in-class FPGA synthesis techniques. It is the first and only synthesis product targeting ASIC prototyping and verification using multiple FPGAs. Using Certify makes ASIC prototyping significantly easier, shortens prototype development time, improves prototype performance, and enables faster time to market.

Verification methodologies for System-On-Chip ASICs

ASIC designers are under a whole series of pressures. They need to produce designs that provide maximum integration, pushing to the limits the capacities of the latest process technologies. They also have to respond to time-to-market pressures, and quickly produce a fully functioning and working device.

Because of the enormous time to market pressures System-on-Chip and other devices with software content face, it is no longer possible to delay software development until working devices are available. Hardware-software co-design is imperative.

The EDA industry is responding by producing new tools and approaches. High level description languages and improved libraries have dramatically improved designer productivity. The rise of IP cores have allowed the designer to use proven elements to build large sections of the design quickly and confidently.

The bottleneck comes from verifying that the ASIC design is correct before moving to production. With huge NRE costs associated with every spin of a million-gate System-On-Chip ASIC, there is an understandable reluctance to move to production before verification has established whether the design will function. As design size grows, the number of vectors required to adequately simulate the design grows even faster. Verification consumes 50 percent of today's ASIC design cycle, and is projected to consume 70 percent of the design cycle in the next two years

There are several verification routes available to the System-On-Chip designer, each with its own cost-performance trade-offs as shown in **Figure 1**. Traditional simulation techniques fall far short of the verification speed needed to verify today's complex ASICs. Cycle-based

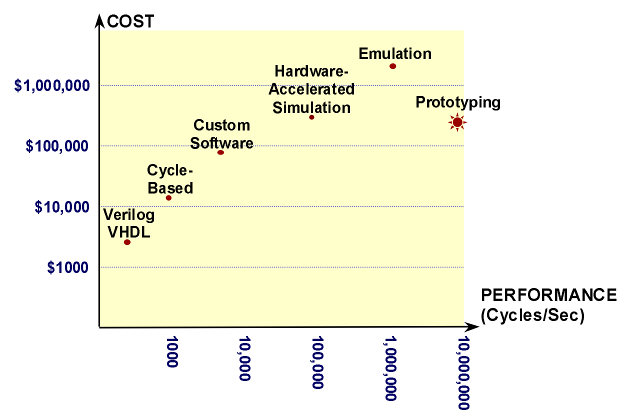

Figure 1

simulators, and custom software models are gaining in popularity, but they cannot achieve the required throughput. Hardware accelerators can achieve better performance. However they are expensive, difficult to use, and still do not reach the performance speeds needed to verify high data rate applications, such as MPEG video processing. Even more expensive and even more difficult to use is emulation hardware, yet this is still not beginning to approach the performance that today's designs need.

ASIC designers are increasingly turning to custom or reconfigurable prototyping environments as the solution for system verification.

Custom prototyping using FPGAs

Custom and reconfigurable prototyping systems allow designers to mix multiple FPGAs along with standard components on the same prototype board. The random logic portion of the design is partitioned across the multiple FPGAs on the board, and standard components (such as memories and microcontrollers) are used to implement the rest of the design. The prototype board typically runs at MHz speeds, allowing the attachment of real-life interfaces. A video processor can be connected to video cameras, frame buffers and display screens, and mobile telephony applications can be configured to make calls.

Typically an FPGA array prototype will be used when output needs to be converted into large volumes of audio or visual data, as is the case with telephony or video applications; when test vector requirements are enormous or generating vectors is impracticable as happens in many networking environments; or when hardware-software integration must begin early in the product life-cycle, requiring multiple system prototypes before the first test silicon is available.

Synthesis/Partioning bottleneck

While the prototype approach offers many advantages as a verification tool, partitioning the design across multiple FPGAs can introduce its own delays into the design cycle.

Figure 2 shows the traditional design flow for building custom prototypes. Synthesis and partitioning are typically separate, disjoint steps in the flow. Synthesis decisions are made without understanding how the design is to be partitioned across multiple FPGAs, resulting in prototypes that do not run at the desired speed. Partitioning decisions are made without any feedback about their impact on utilization and system performance. The result is that designers must typically iterate several times between synthesis and partitioning. Each iteration is time-consuming and extremely tedious, resulting in loss of productivity and increasing time to prototype.

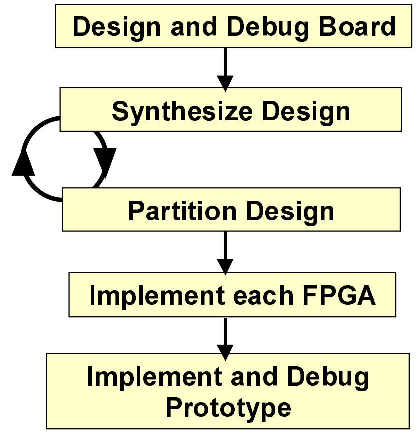

Figure 2

This process may need to be repeated when the prototype has been exercised, and changes are made to the original RTL source.

A further problem is that the carefully planned partition is not used by the synthesis tool to optimize the circuit across the FPGAs. This can slow down the prototype performance, and may force the prototype on to more devices, or on to larger, and more expensive devices.

Certify combines the synthesis and partitioning steps into a single tool. This eliminates the need for time-consuming iterations, and reduces the time required to build the prototype. The net result is a tool that shortens prototype development time and enables faster time to market.

Partition-driven synthesis

Certify is built on top of the core synthesis engine used in Synplify®. The Synplify engine includes architecture-specific synthesis algorithms for each FPGA architecture, and includes powerful timing-driven synthesis algorithms designed to achieve the highest possible system speed.

The core synthesis algorithms have been augmented in Certify with a unique partition-driven synthesis approach. With this approach, the decisions made during synthesis are based on an understanding of the defined partition. Automatic time budgeting across multiple FPGAs is included in the partition-driven synthesis algorithms.

This approach is at the heart of the Certify solution, improving prototype performance,and allowing designers to verify their designs at near real-time speeds.

The Certify synthesis engine, based on the proven Synplify technology, has the capacity to handle designs in excess of 1 million gates. It also has the power to rapidly synthesize. In a typical synthesis test running on a 450 MHz PC platform with one gigabyte of memory, Certify synthesised a 1 million gate block partitioned across several devices in under nine hours.

The Certify user interface

Certify provides an easy to use graphical user interface (see **Figure 3**) that allows users to quickly create legal partitions, and estimate the speed and efficiency of the prototype. It draws on the successful HDL Analyst™ tool, and automatically generates RTL block diagrams from HDL code and displays both in windows. Other windows display a graphical representation of the prototype system board and the target FPGAs.

The first step in Certify is to estimate the area and I/O

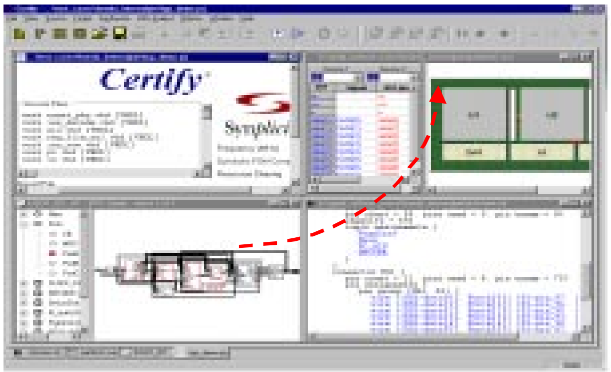

Figure 3

count for each module in the design. The estimation is done by invoking the synthesis engine in estimate mode; the results of the estimates are displayed as annotations to the RTL block diagram.

The user partitions the design by selecting blocks from the RTL block diagram and dragging them onto the target FPGA device. Certify provides immediate feedback on the percentage of I/O and area utilization for each device. This gives the user complete control over the assignments, and quick feedback of the consequences of partitioning decisions.

Certify includes powerful "what-if" analysis that allows users to quickly analyze an area and I/O impact of partitioning decisions. The "what-if" analysis is invoked by clicking the right mouse button on any module in the design. Using the connectivity between the selected module and all previously partitioned modules, Certify computes the area and I/O impact of assigning the selected module to each of the FPGAs in the prototype board, and feeds this information back to the user. The designer can then use the information to assign the selected module to the most appropriate FPGA while ensuring the resulting partition is legal for area and I/O utilization.

When the assignment is complete, Certify uses the physical partitioning information to drive the synthesis algorithms. The choices affect the circuit implementation so that timing critical paths can be optimized while taking into account the delays associated with going on and off chip, and the delays introduced by the board.

Logic replication without modifying RTL source

To improve the prototype performance and to minimize the number of connections across FPGAs, Certify enables designers to replicate logic without making changes to the HDL source. For example, in **Figure 4a**, the 16-bit output of a decoder drives logic in three other FPGAs. Routing the bus across the board could cause slower system speed, could create boards layout problems, and requires a heavy usage of precious I/O resources.

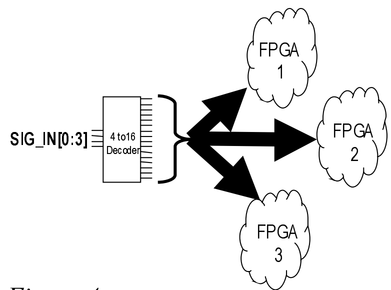

Figure 4a

Instead, as shown in **Figure 4b**, the decoder can be replicated on each of the three FPGAs. While this takes a greater total chip area, it reduces the amount of board traffic and cuts down drastically on I/O requirements. In Certify, this replication can be done by simply dragging and dropping the decoder from the RTL view into each of the three FPGAs in the board view.

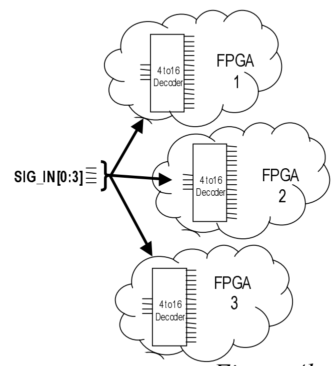

Figure 4b

Since the main goal of prototyping is to verify the ASIC design, designers are very reluctant to make changes to the HDL source solely for the purpose of prototyping. In fact, in many organizations, the verification team is strictly prohibited from making changes to the source code. In traditional synthesis and partitioning flow used to build prototypes, it is impossible to replicate logic across FPGAs without making changes to the HDL source. This leads to inefficient - and sometimes infeasible - prototypes. Only Certify enables designers to replicate logic without making changes to the original HDL source code, simply by dragging and dropping objects from the RTL view into multiple target devices.

Probe point creation without modifying RTL source

In traditional approaches to synthesis and partitioning, the specification that a signal needs to be exported to the boundary of the FPGA as a probe point, requires modifications to the HDL source. Only Certify enables designers to specify probe points without modifying the HDL source, simply by dragging and dropping the signal to the pin assignment table associated with the FPGA.

Iterative design flow

One of the main goals of prototyping is to quickly find and fix problems in the original design. Tools for prototyping must be optimized to support an iterative prototype-debug-fix-prototype-debug cycle. Certify is optimized for this design flow. In addition to creating a gate-level netlist for each of the FPGAs in the design, Certify also outputs source-level HDL code for each of the FPGAs. Iterations to fix bugs in the design are typically limited to a small number of modules in the design. With Certify, only those FPGAs impacted by the change need to be re-implemented, reducing the time required to make incremental design changes

Intellectual property

One of the increasing trends in large ASIC designs, particularly those regarded as System-on-Chip, is the reuse of designs, either those developed in-house or bought in as IP cores. Certify accommodates these easily.

If the RTL is available, this can be fed into the design process, and prototyped onto one or more FPGAs. Another mechanism is to acquire the IP as a bonded out core, and then mount it into the prototype system. This is then treated as a black box chip within Certify. The device is defined in the board layout, and then instantiated during the partitioning process. This same approach can instantiate standard functionslike UARTs and memory.

Flexibility in implementation choices

The output from Certify creates a top-level netlist for the prototype board and an optimized netlist for each FPGA, ready for place and route. This allows the user to define and build a custom prototype board, and to create prototype boards with routing interconnect devices such as I-Cube. It can also be used as input to the popular System Explorer product from Aptix. System Explorer takes the netlists created by Certify as input, places the devices on the prototype board, configures the programmable board interconnect, and then invokes the FPGA place & route tools to implement each device

Certify supports the most popular FPGA devices from leading FPGA vendors. The Altera Flex 10K, Xilinx 4K, and Xilinx Virtex families are supported in the first release of Certify. The ability to use the most popular FPGA devices and target virtually any prototype board using Certify provides the user with tremendous flexibility in implementation choice

Certify: Easing the verification bottleneck

Certify combines RTL multi-chip partitioning with FPGA synthesis techniques, and is the first and only synthesis product targeting ASIC prototyping and verification using multiple FPGAs. Certify's unique partition-driven synthesis approach enables designers to achieve the fastest prototype speed. Using Certify makes ASIC prototyping significantly easier, shortens prototype development time, improves prototype performance, and enables faster time to market.

Synplicity, Inc.
610 Caribbean Drive
Sunnyvale, CA 94089
Phone: (408) 548 6000
Fax: (408) 548 0050

http://www.synplicity.com