White Paper

FPGA Prototyping of System-on-Chip Designs

The Need for a Complete Prototyping Platform for Any Design Size, Any Design Stage with Enterprise-Wide Access, Anytime, Anywhere

S2C Inc.

1735 Technology Drive, Suite 620

San Jose, CA 95110, USA

Tel: +1 408 213 8818

Fax: +1 408 213 8821

www.s2cinc.com

Overview

Today's off-the-shelf FPGA prototyping systems have established their value in every stage of the system-on-chip (SoC) design flow. Moving beyond traditional applications such as in-circuit testing and early software development, this technology has expanded to encompass functional design and verification. FPGA-based prototypes work with electronic system level (ESL) design environments to refine, validate and implement chip architecture[1,2], and with simulation tools to achieve an order of magnitude (or more) increase in verification speed[3].

There are several drivers of this technology: the need to quickly construct high-performance prototypes; the demands of growing design size and complexity; and the need to utilize prototypes as an enterprise-wide resource. Globalization has replaced localized design teams with teams that are geographically-distributed. Consequently, FPGA prototyping solutions must now provide network access, remote management capabilities, coupled with the ability to expand resources such as memory or add-on components. This allows realizing multiple hardware and software implementations for numerous, geographically-dispersed teams.

The FPGA prototyping system must offer enterprise-wide accessibility — a complete prototyping platform is one that operates at any functional design stage, with any design size, and across multiple geographical locations. All of these capabilities must be available on demand and remotely-accessible at all times. Such an approach would significantly increase engineering productivity and reduce the end-product's time to market, while increasing its return on investment (ROI), as well as increasing the lifetime ROI of the FPGA prototyping platform itself.

This paper addresses how the FPGA prototyping system must evolve into a complete prototyping platform to meet the ever-growing SoC design challenges. Specifically, the paper:

- Reviews the growing challenges in SoC development.
- Examines the ability of current FPGA prototyping approaches to meet these challenges effectively.
- Outlines the requirements for FPGA prototyping solutions to meet these challenges.

Growing SoC Design Challenges

SoC size and complexity are increasing at an exponential rate. According to a keynote presentation by Gary Smith at the International Technology Roadmap for Semiconductors Conference in 2013[4], potentially available SoC gate counts will quadruple from 420 million in 2014 to 1.68 billion in 2020. International Business Strategies (IBS) reported that software development and hardware verification are the

---

[1] How to Reduce Respins by Using FPGA Prototyping Inside a Complete Flow. K. Farnham. ESL Design Notebook 2014

[2] ESL Models and their Application, Chapter 5. B. Bailey and G. Martin. Springer Verlag 2010.

[3] FPGA Prototyping in Verification Flows. M. Litterick, Verilab GmbH. 2006

[4] Silicon Platforms + Virtual Platforms = An explosion in SoC design by Gary Smith. ITRS 2013.

two leading factors in total SoC design cost[5] (see figure 1). A 2012 report[6] by Semico Research reaches similar conclusions.

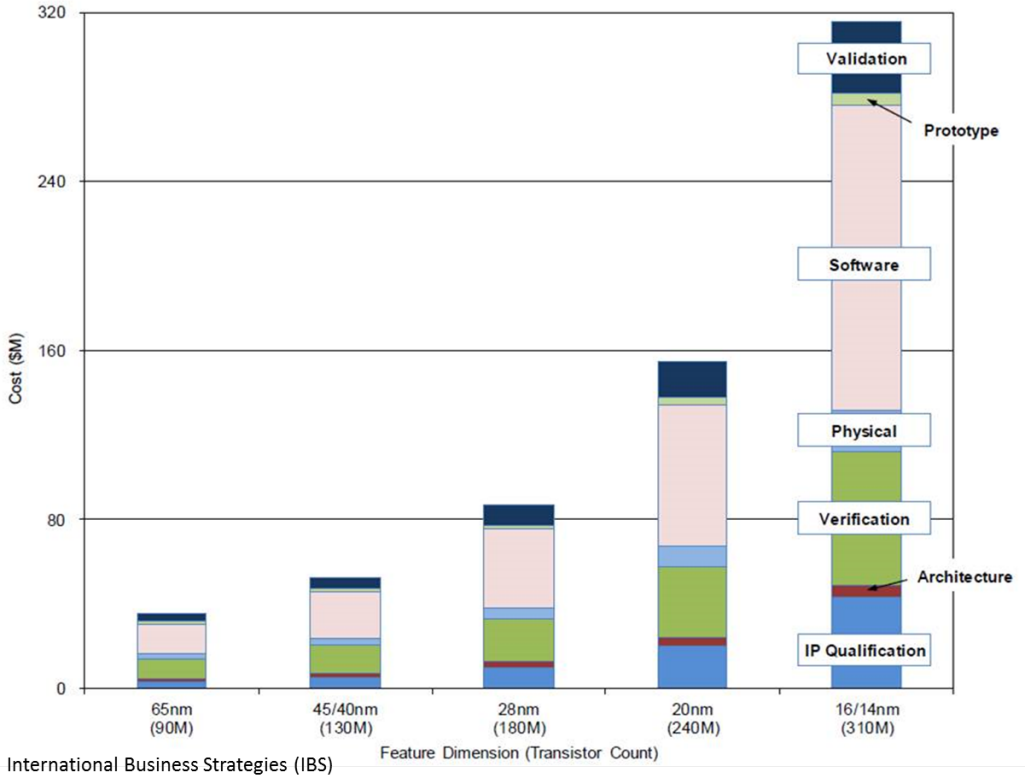

Figure 1: Software development and hardware verification are the predominant factors in SoC design cost (IBS)

These software- and complexity-driven cost and effort increases are accompanied by an elevated risk of late delivery — and even outright failure. Cost and risk are generally mitigated by the extensive use and reuse of intellectual property (IP) — both silicon and software — but the complete silicon/software design must nonetheless be prototyped and tested as a whole.

FPGA Prototyping Solutions: Addressing Todays Customer Needs

For an FPGA prototype to meet the requirements of this "whole design", it must address these criteria:

- User access
- Compile/partition efficiency
- System interface capability
- Scalability
- Extensibility
- Reusability
- Analysis and debug capability
- Application throughout the functional design flow

[5] How Much Will That Chip Cost? Ed Sperling, Semiconductor Engineering March 27th 2014
[6] SoC Silicon and Software Design Cost Analysis: Costs for Higher Complexity Continue to Rise. Semico Research. Solid State Technology, May 2013.

Utility of Current FPGA Prototyping Systems

These are the key criteria for evaluating the utility of an FPGA prototyping system:

1. **Access** to FPGA prototyping systems must not be constrained by the use of localized systems that require local management and control. Limited access can present a significant hindrance to modern SoC design teams — especially software development teams — which are often globally distributed.
2. **Compile and build environment** must incorporate important features such as the ability to partition a design automatically and/or with user guidance; automatic pin-multiplexing insertion and clock analysis. Also important are a convenient user interface to FPGA-specific place and route tools allowing for quick flow turnaround for changes and ECOs.
3. **Performance** – the key reason teams develop FPGA-based prototypes. FPGA-based prototypes can be expected to realize system speeds in the 10's of megahertz – some have been known to run at 100 Mhz and more. High-speed FPGA prototypes enable early software development.
4. **High-speed interfaces and add-ons**, such as PCIe, USB, 10GE, ARM Debugger, and DDR memory are important for building a complete development platform. Transaction-level interfaces such as a CAPI and AXI bus protocol support greatly expand the utility of the system.
5. The ability to **scale and extend** the system is also an important consideration. This involves adding gate-capacity and memory, as well as processors and communication interfaces to grow the system functionality.
6. **Reusability** is inherent in off-the-shelf FPGA prototyping systems. A system's reusability in subsequent designs is determined by the quantity and diversity of its resources — gates, memory and processing power. These resources must grow to remain up to date with changing functionality needs.
7. **Analysis and debug** must not be limited to one FPGA at a time, which makes whole-system debug slow and tedious. Signal probing should work easily with designs partitioned across multiple FPGAs; a deep debug trace capability must be provided with probing schemes that maximize the use of FPGA pin I/O.
8. Support for a **mixed-level prototype**. Often during development not all blocks are available in RTL. A complete prototyping system should support behavioral blocks running on a host computer to interface and communicate with RTL blocks mapped to FPGAs.

Complete Prototyping Platform — The Complete Solution

Given the attributes and shortcomings of existing FPGA prototyping systems, what should be the attributes of the next generations of systems? As noted, the coming generations of off-the-shelf FPGA prototyping solutions must offer greater choice and flexibility in the deployment of resources. Consequently, the coming generations must take a "complete prototyping platform" approach. They must:

1. Provide sufficient **performance** to operate as software development platforms. This requires PCB's with superior signal characteristics, clocking strategy, and connector structure.
2. Provide **capacity, scalability, extensibility, and reusability** – modern FPGA-based prototypes can support designs from 20M to 500M gates. Add-on DDR2/3 modules can quickly create systems with multi-gigabyte memories. Peripherals and processors available on daughter cards make it easy to add various IP functionality.
3. Increase **scalability, extensibility and reusability** by enabling the deployment of an ever-increasing range of IP — both silicon and software — and pre-designed board-level subsystems. This requires the availability of copious IP and board-level reference design options.
4. Support the latest **high-speed interfaces** such as PCIe, HDMI, and 10Gig Ethernet.
5. Include an easy **compile and build** environment to accelerate the process of prototype bring up, as well as ease the processing of design ECOs. Automatic partitioning tools should accept user input that can result in faster-running prototypes.
6. Offer global **remote access** to centralized prototyping resources, enabling access by multiple, geographically-distributed teams. This can be achieved by cloud-based control and storage.
7. Support **behavioral/transaction-level interface** and standard bus protocols such as AXI to ease system integration and provide a platform for software development.
8. Perform **analysis and debug** of the whole SoC design at full system speed, with a focus on "deep" hard-to-find bugs. Simultaneously, it must achieve at least an order of magnitude increase in signal and cycle observability, without consuming FPGA resources such as gates, memory and I/O connectivity. This requires the implementation of structures that offer "pervasive observability" of design functionality. Fast probe-swapping is essential.
9. Continue to team with **functional design and verification tools** to perform tasks at every stage of the functional design flow, such as cross-leveraging the strengths of FPGA prototyping and simulation. This requires tight, well-integrated bridges between the diverse design environments.

Conclusion

A modern FPGA-based prototyping system must meet a number of demanding criteria to help designers realize their latest system-on-chip designs. An extensible, scalable system must offer a variety of both hardware and software interfaces. High-performance and extensive debug capabilities are critical requirements. The ability to function as an enterprise-wide resource, with the easy access and configurability of a cloud service, multiplies the value of such a system. Meeting these criteria and combining features in a rich set of functionality qualifies a system as a truly complete prototyping platform.