

Strategies for ASIC Board-Level Validation

White Paper

Issue: 01, 16th January 2002

Gus Paolone, VP Embedded Systems

Abstract

With the advent of multi-million gate System-On-a-Chip (SOC) ASIC devices, the validation effort required has become a significant challenge, and can amount to more person-months of effort than the development of the devices themselves.

This paper describes strategies for ASIC board-level validation. It presents the advantages of undertaking an open systems approach to the development of validation strategies, and highlights the opportunities that exist to maximize circuit and embedded software reuse from the validation effort for eventual deployment in the ASIC device end-user context.

A number of validation board architectures are presented, along with a discussion of how these architectures can be extended into a validation system to facilitate chipset testing.

1 Introduction

ASIC development requires extensive validation effort during both the design and prototype phases of the project. In addition to simulation of an ASIC during design, board level validation of a prototype ASIC provides optimum test coverage, and permits verification in a real user design context.

The validation board is used by the ASIC validation team to test and qualify the design, and can be substantially reused as a reference design by the ASIC device end user in their developments. Furthermore, as part of the development of the validation board, the software effort can produce a device driver that can be made available as a “value-add” component to the ASIC device end user, facilitating rapid deployment of the device into their products.

The validation environment chosen may be one that can be extendable and re-usable across complementary ASIC devices that belong to a particular product family. This can help achieve savings when using previously developed validation boards and can also create the opportunity to build a system using ASIC devices from a chipset family.

2 Board-level ASIC Validation

Board level validation can allow the various design contexts envisaged in the target applications to be verified. However, a concerted effort from the design team is required as there is usually a short window of opportunity between ASIC device prototype and final silicon. To this end, an independent validation team can be used to work in parallel in order to have the validation hardware and software ready prior to the prototype ASIC devices becoming available.

The ASIC design process includes a number of checkpoints that enable validation at various stages of the design as shown in Figure 1. The final validation of the device design is on a board that represents the intended target environment.

Board level validation, enables the validation of all timing and loading parameters found during gate level simulation and permits functional validation against RTL simulation results.

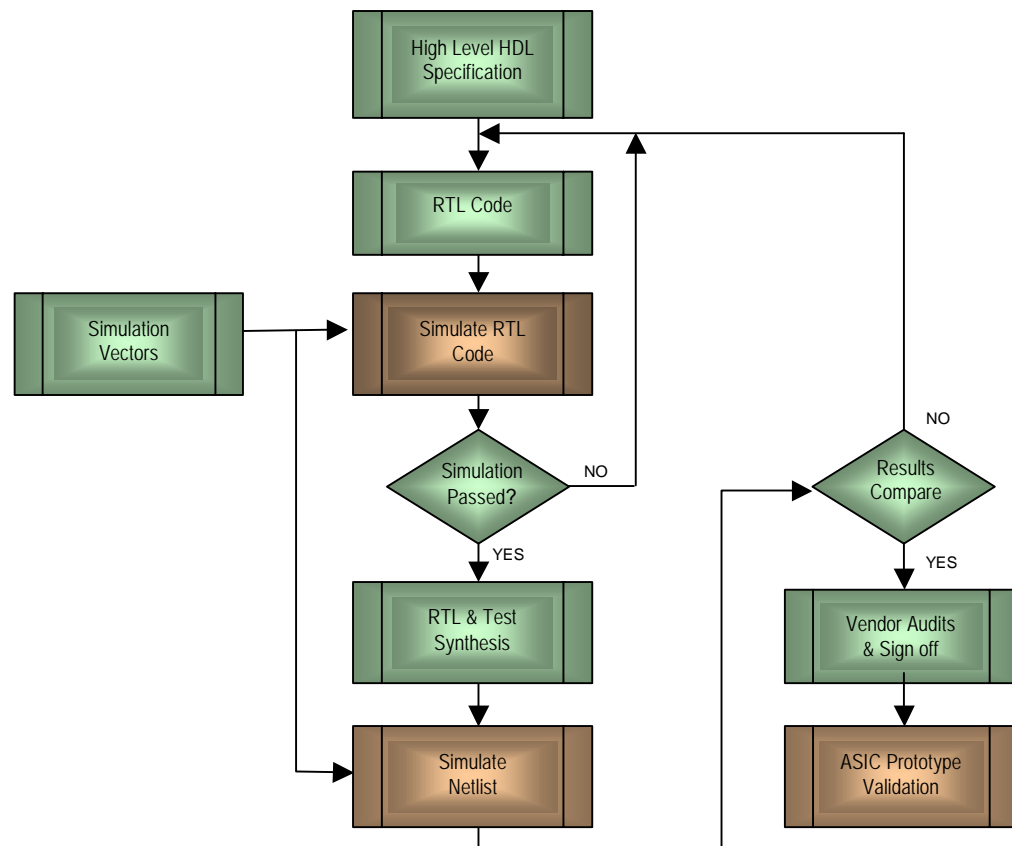


Figure 1 Typical ASIC Development Cycle [1]

3 ASIC Validation Board Architecture

The ASIC validation board enables the resolution of critical issues such as PCB layout, power supply requirements and interconnectivity to other components.

The board must be designed with controllability and observability in mind. This generally means the provision of access points to which standard test equipment may be connected. These access points can then be used to characterize and validate the timing specifications of particular signals on the ASIC device.

In more complex boards, additional hardware such as custom FPGA devices are used to generate stimulus, alter externally injected data and capture and analyze data from the ASIC device.

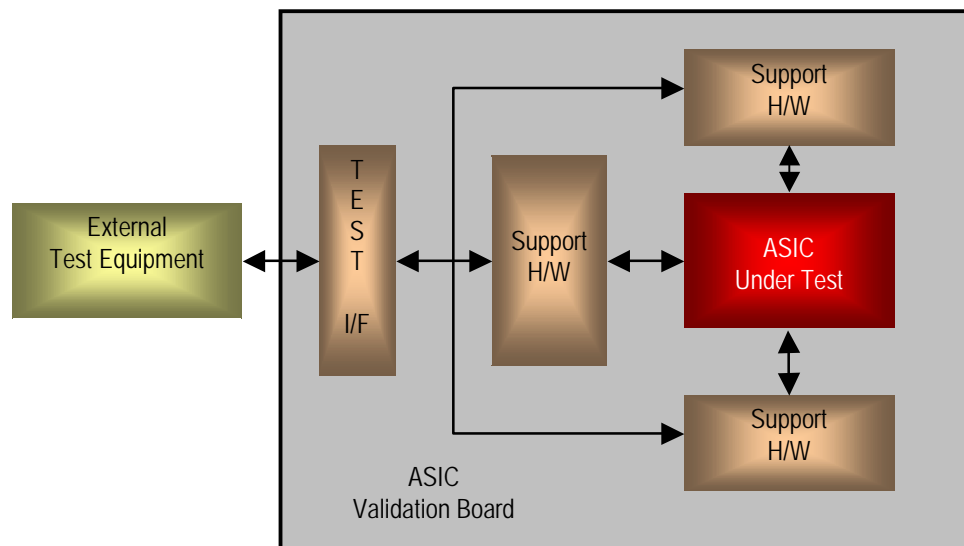


Figure 2 ASIC Validation - Board Architecture

The diagram in Figure 2, illustrates a Validation Board topology with the ASIC under test, and associated support hardware. The associated support hardware may incorporate interface FPGA devices, standard IC devices, power supply, microprocessor, etc.

4 ASIC Validation System Architecture

Occasionally, it may be required that the ASIC device be tested in conjunction with other devices, for example, when developing a chipset. Under these circumstances, it may be necessary to connect a number of validation boards together to build a “validation system”.

The interconnected boards may require buses to connect the data signals that are shared between the device under test and other boards. In addition, a controller bus may be required depending on the control architecture.

When adopting a system of interconnected validation boards, pre-existing validation boards can be reused to validate new complementary ASIC devices.

Here we introduce three extendable ASIC Validation System topologies; Compact PCI Bus, Universal Serial Bus, and Local Area Network.

4.1 Compact PCI Bus

The diagram in Figure 3 illustrates a validation system with two ASIC Validation boards connected to a controller with a Compact PCI (cPCI) Bus. This diagram illustrates the case where a single controller is used to manage the entire system, with a cPCI bridge device on each of the validation boards.

The common processor is connected to a Host Computer where test scripts are executed in batch mode or interactively.

In the figure, the signals shared by the ASIC devices under test are shown as the “Application Bus” and are separate from the microprocessor control bus.

This architecture requires that the boards be connected to a cPCI backplane housed in a suitable chassis. Flexibility is also improved as it allows additional boards to be connected to the validation system to extend its functionality.

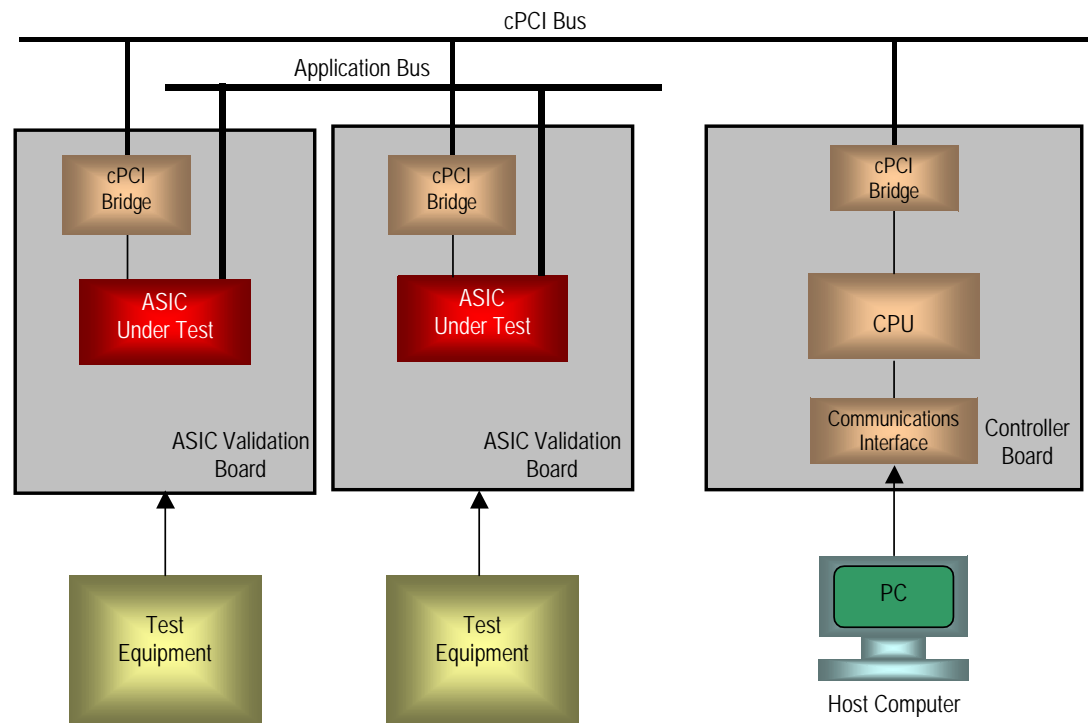


Figure 3 Validation System - cPCI Bus

4.2 Universal Serial Bus

The diagram in Figure 4 illustrates a validation system architecture in which a Universal Serial Bus (USB) is used for control. Two ASIC validation boards are connected to the Host Computer using USB.

USB offers a high-speed connection between the validation boards and the Host Computer. In addition, USB has the added flexibility of hot-swap functionality and is well supported on a variety of host operating systems such as Windows, MacOS and Linux.

The architecture illustrated in Figure 4 uses local processors (usually embedded into the USB interface device) on each validation board. These processors implement the USB protocol as well as the functionality required to support the Host Computer test scripts such as register access, memory test, data and address bus testing, etc.

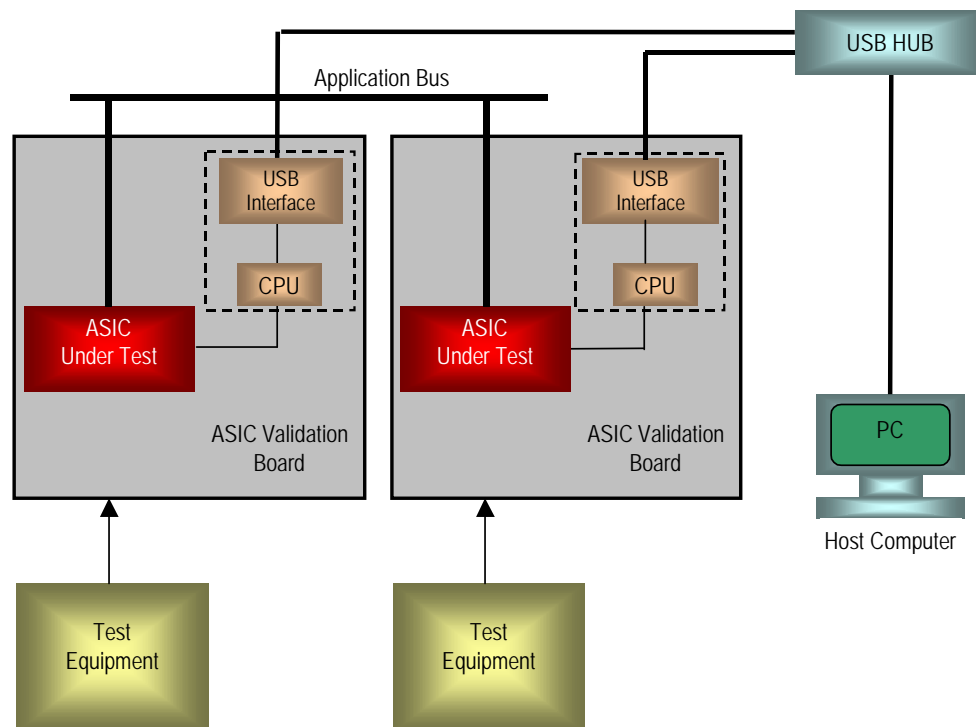


Figure 4 Validation System – USB

4.3 Local Area Network

The diagram in Figure 5 shows the ASIC validation boards interconnected using a 10/100Base-T Local Area Network (LAN). This LAN is primarily used for control and requires that each validation board include its own controller and associated network software.

Through the LAN interface the ASIC validation board can be accessed by multiple users from any network enabled Host Computer anywhere within a corporate LAN or across the Internet, removing any distance restrictions.

The validation system can make use of a range of applications available to networked devices, such as telnet or web servers. These applications are well established, easy to add-on and enhance the accessibility of the validation system.

Test equipment can also incorporate LAN connectivity, enabling the creation of a fully automated test environment controlled by test scripts from the Host Computer.

A further advantage of the LAN approach is one where the Validation Cards can be accessed by the end user, as part of their ASIC evaluation efforts. They may access the facilities of the new ASIC device via an integrated web server, allowing them to evaluate the feature set, in a validation environment that emulates their design context.

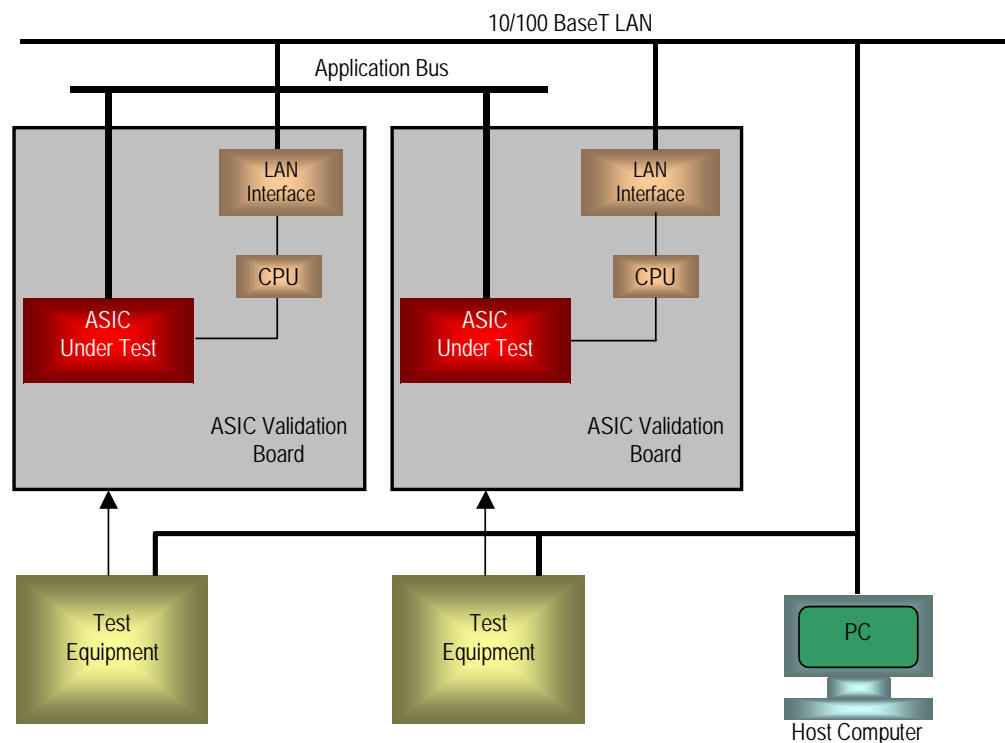


Figure 5 Validation System – 10/100BaseT LAN

5 Software

The validation architectures described in the previous section require an embedded application capable of configuring and controlling the devices on the board. In addition, a host application provides a user interface and implements various test functions required to validate the device under test, as shown in Figure 6. This approach has the added benefit that the ASIC is being subjected to the intended operating conditions of the device. These real world conditions include accessing the device at full bus speed and correctly handling real-time events such as interrupts, within an embedded real-time environment.

5.1 Device Configuration and Control

The embedded application uses a device driver to access the ASIC under test. This driver gives the embedded application access to the ASIC device registers, onboard memory (if appropriate) and provides interrupt service routines in response to interrupts.

The device driver can be made available for deployment to the ASIC device end user. The driver can be designed with portability in mind. It can be operating system independent and not be dependent on the processor architecture.

5.2 Test Functionality

The embedded application provides basic test functionality to support the validation of the ASIC under test. Some good candidate tests required to validate an ASIC that benefit from an embedded implementation include:

- Register Access Tests
- Data and Address Bus Tests
- Memory Tests
- Power Management Control
- Complex Configuration

Additional tests can be included depending on the functionality of the ASIC device.

This basic set of test functions can be extended into complex test scenarios by the test scripts executed at the Host Computer.

5.3 User Interface

The model presented in this paper is one where the real-time processing is done by the embedded application. To access the features provided by this application, a communications link is provided between the embedded application and a test script executed on a Host Computer.

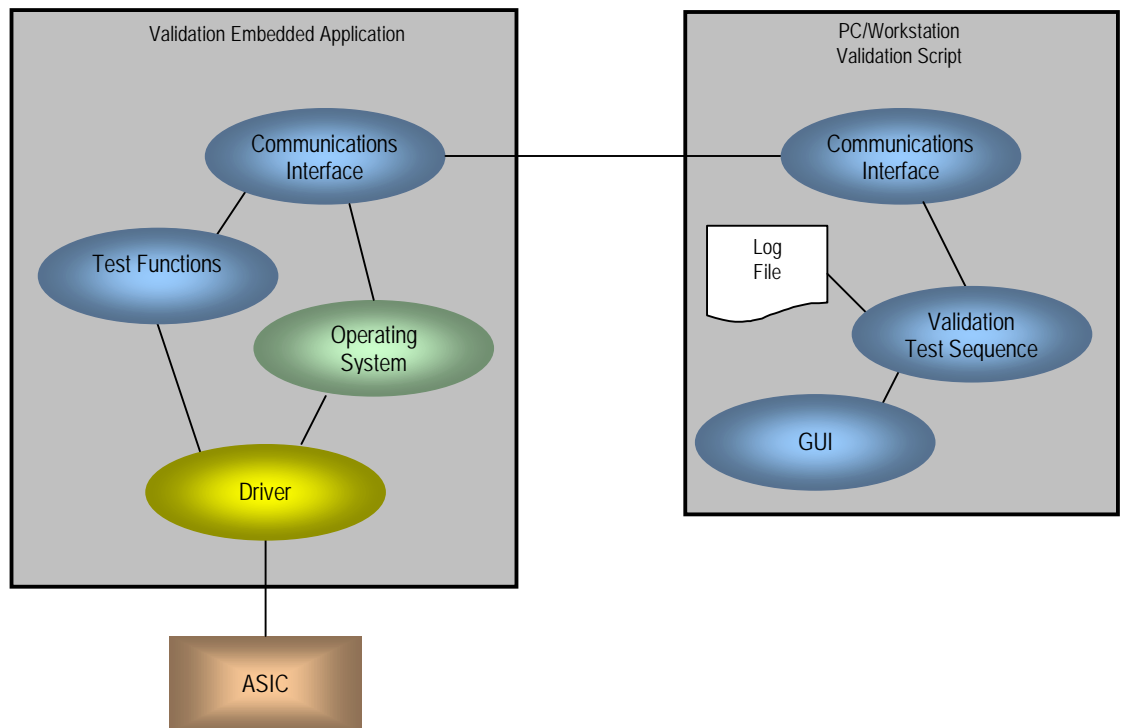


Figure 6 Validation Software Architecture – Graphical User Interface

Two main methods exist for controlling the embedded application from a Host Computer. The first is interactively through a user friendly GUI, the second is batch mode. Both methods are complementary; each having important uses and should be considered when designing such applications. Interactive control of the validation environment permits manual configuration of parameters that affect the state of the device under test. This is useful when making manual measurements and observations. However batch mode is consistent with a more automatic mode of testing and is more suitable for regression testing.

Either mode of testing can be implemented using scripting languages such as Perl or Tcl/Tk. These scripting languages are portable across the commonly used operating systems, are easy to write and extend, removing the need for expensive software development tools, and highly trained software engineers. Tcl/Tk is a good example of a scripting language that can implement a GUI and network connection in a relatively small script.

6 Conclusion

ASIC validation can benefit from the application of open system verification strategies that enhance reuse of key aspects of the validation effort.

This model creates an opportunity to develop device drivers and reference designs that can be made available to the ASIC device end user. In addition, the board permits designers to identify and correct device or board level issues ahead of device deployment.

Choosing the right validation board architecture can also improve reuse and help validate complementary ASIC devices, leading to greater confidence of their interoperability.

Glossary

ASIC	Application Specific Integrated Circuit
CPCI	Compact Peripheral Control Interface
CPU	Central Processing Unit
FPGA	Field Programmable Gate Array
GUI	Graphical User Interface
H/W	Hardware
I/F	Interface
IP	Internet protocol
LAN	Local Area Network
PC	Personal Computer
PCB	Printed Circuit Board
RTL	Register Transfer Level
SOC	System On a Chip
Tcl	Tool Command Language
Tk	Toolkit
USB	Universal Serial Bus

References

- [1] HDL Chip Design, Douglas J. Smith, 1996. Doone Publications, Madison, AL, USA
- [2] Block-Based Prototyping Methodology. Article from www.chipcenter.com/pld/pldf061.htm