High-Speed Communications: A Modeling Approach

Foresight Systems M&S

Attention: Please Read This Page

This paper was originally prepared in August of 1997. While we believe that the material is important and will be interesting to current users of Foresight Systems software, we have been through some changes since the paper was written.

Nu Thena Systems, Inc. was a predecessor company to Foresight Systems M&S. All references to Nu Thena are obsolete. Readers should assume that Foresight Systems M&S replaces all instances of Nu Thena Systems, Inc.

The current contact information for Foresight Systems M&S is:

Foresight Systems M&S
12550 N. 89th St.
Scottsdale, AZ 85260

http://www.foresightsystems-mands.com
fs_marketing@foresightsystems-mands.com

The current production versions of the software do not support the direct synthesis to conventional hardware description languages. **FS/CoderC++** does automatically translate graphical Foresight executable system models into C++ code, which can be compiled and linked with a run-time library to form stand alone executables. CoderC, FS/Doc and FS Altia are no longer supported in production releases of the Foresight software.

The diagrams and illustrations included here were prepared with versions of the software current at the time that the paper was prepared. Some of them may not be identical to equivalent images from the current version of the software. We apologize for any potential confusion.

Please excuse the use of any out of date expressions and terminology. The authors believed that their application of technical terms was consistent with standard practice at the time that this document was written.

If you have questions, concerns, or issues, please contact:

Dean Stevens
President, Commercial Systems Group
Foresight Systems M&S
deans@foresightsystems-mands.com
(408) 278-3983

High-Speed Communications: A Modeling Approach

by
Mike Vertal
Nu Thena Systems, Inc.
August 1997

foresight
NU THENA SYSTEMS, INC.™

Introduction

High-speed communications permeate nearly every type of advanced consumer and computer-based product. Numerous high-speed communication standards exist which aim to satisfy this ever-growing appetite for communications bandwidth among and between people, processors, storage devices, displays, and other peripheral computing equipment. These complex and powerful data, voice, and video communications standards accommodate various topologies, a wide variety of transfer speeds, different levels of service, and numerous protocols. Moreover, existing standards continue to evolve and expand to address the continuous stream of challenges faced by high-speed communication system developers.

The communications subsystem design community - consisting of standards bearers, embedded system developers, peripheral equipment developers, chip-set designers, etc. - must keep pace with complex, evolving standards. Standards committee members need to rapidly evaluate new proposals to the standard, and must clearly specify and validate all revisions and extensions. Meanwhile, chip and equipment developers need to ensure that their design teams correctly understand a communication standard's operation, and must validate that their designs comply with its specifications. Moreover, the communication subsystem designers must embed their designs within complex electronic systems and deliver products within shorter time-to-market windows.

This paper discusses how system-level modeling and simulation tools - such as Foresight from Nu Thena Systems, Inc. - can help high-speed communication subsystem designers better meet these challenges.

Foresight™

Foresight provides the design community with the ability to capture specifications and designs of advanced communication systems as executable models. These Foresight models enable engineers to perform extensive analysis to ensure the functional correctness of specifications and designs, to optimize system architectural performance, and to facilitate the downstream (hardware or software) design engineering processes [1].

The fundamental enabling technology upon which Foresight relies for providing these capabilities is its systems-level design language. Foresight combines a collection of comprehensive system modeling constructs for capturing and analyzing system designs at multiple levels of abstraction - ranging from highly abstract performance models to highly detailed functional design models.

Its extensive system modeling language distinguishes Foresight from other modeling tools, and provides the communications subsystem designer with the ability to

1. easily capture communication architectures at a high-level of abstraction, and then analyze and optimize their performance,
2. model and verify functional behavior at multiple levels of detail, and
3. support the downstream hardware/software design activities.

Each of these system design activities, and how Foresight supports them, is described below.

High-level (Performance) Modeling and Analysis

This system design activity allows engineers to rapidly perform high-level design trade-offs and evaluate performance issues. Also known as performance modeling and analysis, this capability enables designers to rapidly capture architectural models, where the nodes within the model are characterized at relatively high levels of abstraction using queues, resources, and statistical sources; see Figure 1. High-level modeling and simulation enables very rapid trade-off analysis at the expense of model fidelity; nevertheless, there are numerous engineering design challenges for which high-level models are of sufficient fidelity.

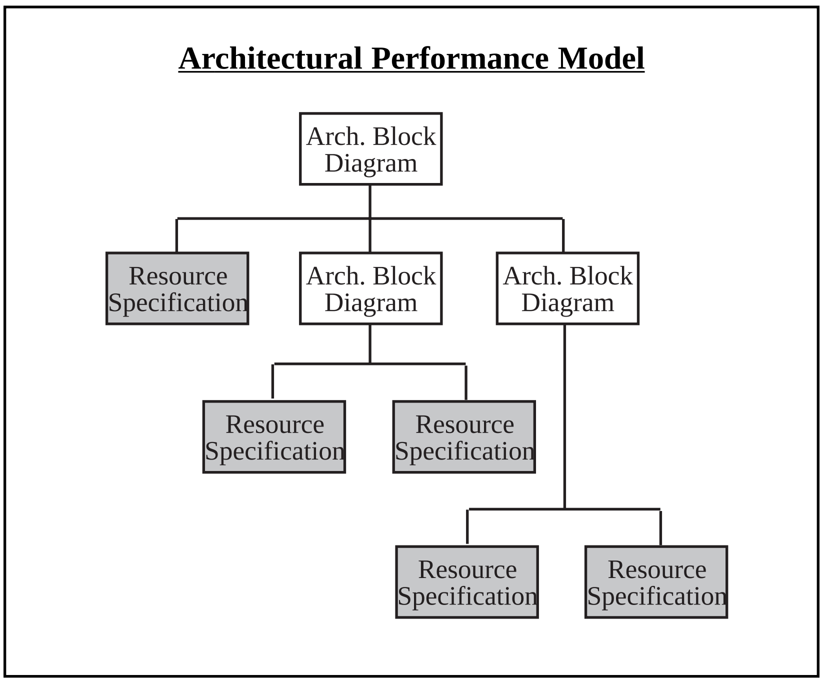

Figure 1. A Performance Model Hierarchy

Communication subsystem designers benefit from performance analysis in various ways. At this level of fidelity, protocols would not necessarily be modeled in any detail. Instead, frames would be modeled as data-less tokens and the fabric would be modeled as a delay node with a stochastically-determined time delay. With the ability to rapidly construct these high-level models, designers can, for example, quickly determine the impact of different data rates and service levels upon an application employing a particular data communications standard, such as Fibre Channel[2]. As another example, an engineer could rapidly evaluate the merits of different topologies for a particular end-user application.

Foresight supports high-level performance modeling with numerous built-in modeling constructs and libraries tailored for computer-based architectural performance analysis. Similar to other dedicated performance modeling tools and simulation languages, Foresight incorporates constructs such as queues (with various scheduling mechanisms), resources, and random number generators to easily capture high-level system behavior. And once a model is built (an example is shown in Figure 2), Foresight's interactive and batch simulation capabilities enable rapid model execution and analysis. Additionally, Foresight's data collection and visualization capabilities allow the user to easily evaluate simulation results, in the form of graphical plots such as histograms, bar graphs, and x-y plots.

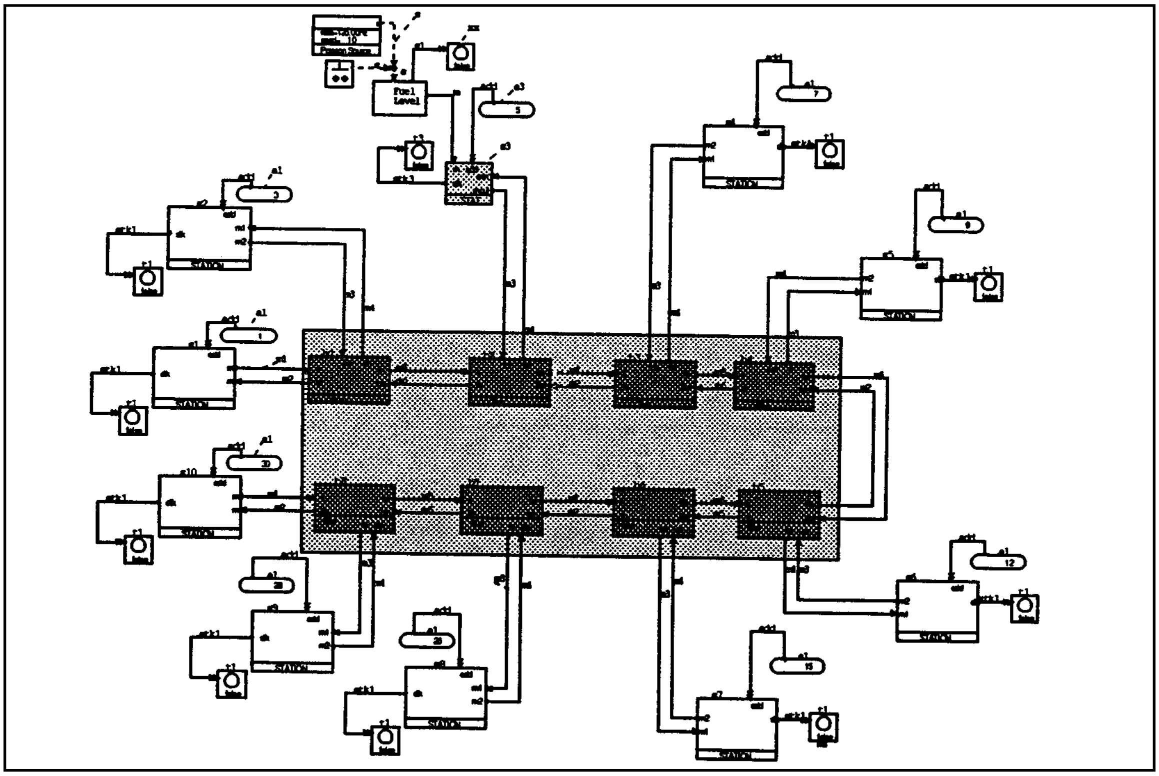

Figure 2. A Foresight Architectural Performance Model of a Linear Token Passing Network

Foresight's system design capabilities do not stop here. In contrast to dedicated performance modeling tools, Foresight extends far beyond the scope of high-level design analysis to include strong support for detailed functional modeling and verification.

Functional Verification

Functional modeling and design verification enables engineers to model system characteristics with more fidelity in order to address more detailed engineering problems. At this level of detail, actual system behavior and data is modeled, although not necessarily to the finest levels of granularity. Different types of specification and design problems demand different levels of modeling detail. As a result, a functional modeling language must support multiple levels of abstraction thereby allowing system designers to refine models to fine levels of detail where necessary, while abstracting away those characteristics that are unessential to a particular problem.

Functional models can serve as excellent reference models for communication standardization efforts. Executable models can be readily used to evaluate alternatives, verify the correctness of specifications, serve as prototypes during the review process, and facilitate clear and unambiguous communication among all standards members. As will be discussed later, these same functional models can accelerate product development cycles as well.

Functional modeling allows the characterization of a communications interface at (and below) the frame level, incorporating actual signaling protocols. Functional models at this level would enable detailed evaluation and verification of the interoperability of newly proposed protocols. Another, more detailed, functional model could characterize the frames down to the bit level, and incorporate low level functionality such as error control. Models at this level of detail could also be used for (more detailed) protocol verification, in addition to facilitating the validation of product designs. Numerous engineering problems can be addressed through functional modeling and analysis, provided the designer has the ability to characterize the system and analyze behavior at various levels of abstraction.

Foresight incorporates a full functional modeling language supporting design analysis at all levels of abstraction. As shown in Figure 3, Foresight functional models are built hierarchically from a collection of modeling constructs. Foresight's primary modeling construct is a hierarchical block diagram. These block diagrams allow systems to be described as a network of communicating processes, which are well suited for modeling high-speed communications systems functionality and their inherent concurrency.

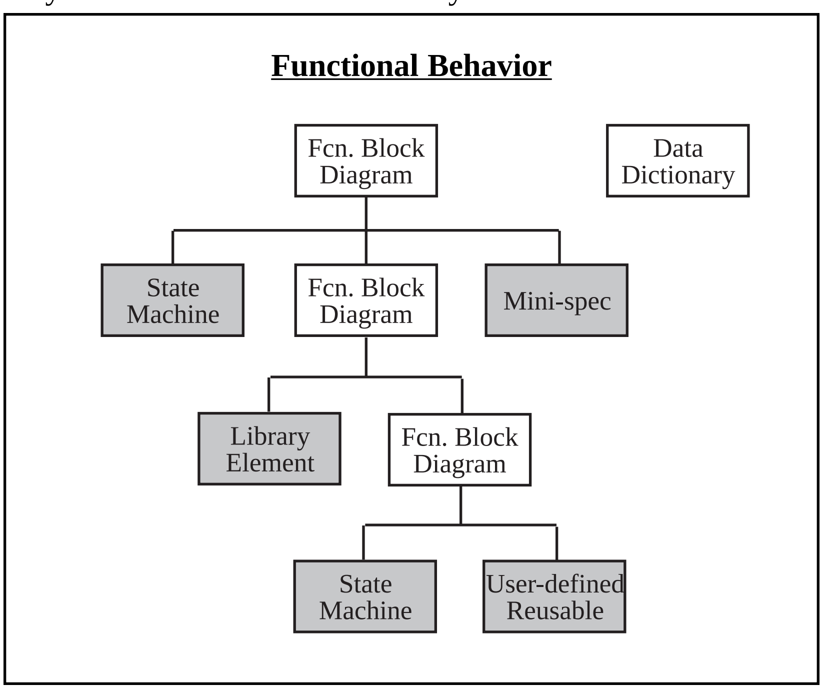

Figure 3. A Foresight Functional Model Hierarchy

Process blocks within a Foresight block diagram may be decomposed with more block diagrams, or with Foresight primitive, executable modules: state diagrams, mini-specifications, pre-defined library elements, or user-defined reusable elements. These executable elements have the expressive power to capture essentially any type of functional behavior, at any level of detail. All process blocks are connected with data flows, whose data types are stored and maintained in a data dictionary that supports a full range of data types, including integers, reals, enumerations, bits/bit vectors, records, and arrays.

Foresight's built-in mini-spec language, which is a procedural modeling language, has a VHDL-like syntax [3] and incorporates all the constructs found in any sophisticated procedural programming language. This, combined with Foresight's familiar block diagrams and state diagrams (see Figure 4), enables engineers to quickly learn Foresight's comprehensive modeling language, and to rapidly exploit its expressive power. Additionally, Foresight's strong support for model reusability enables rapid model development and encourages design reuse.

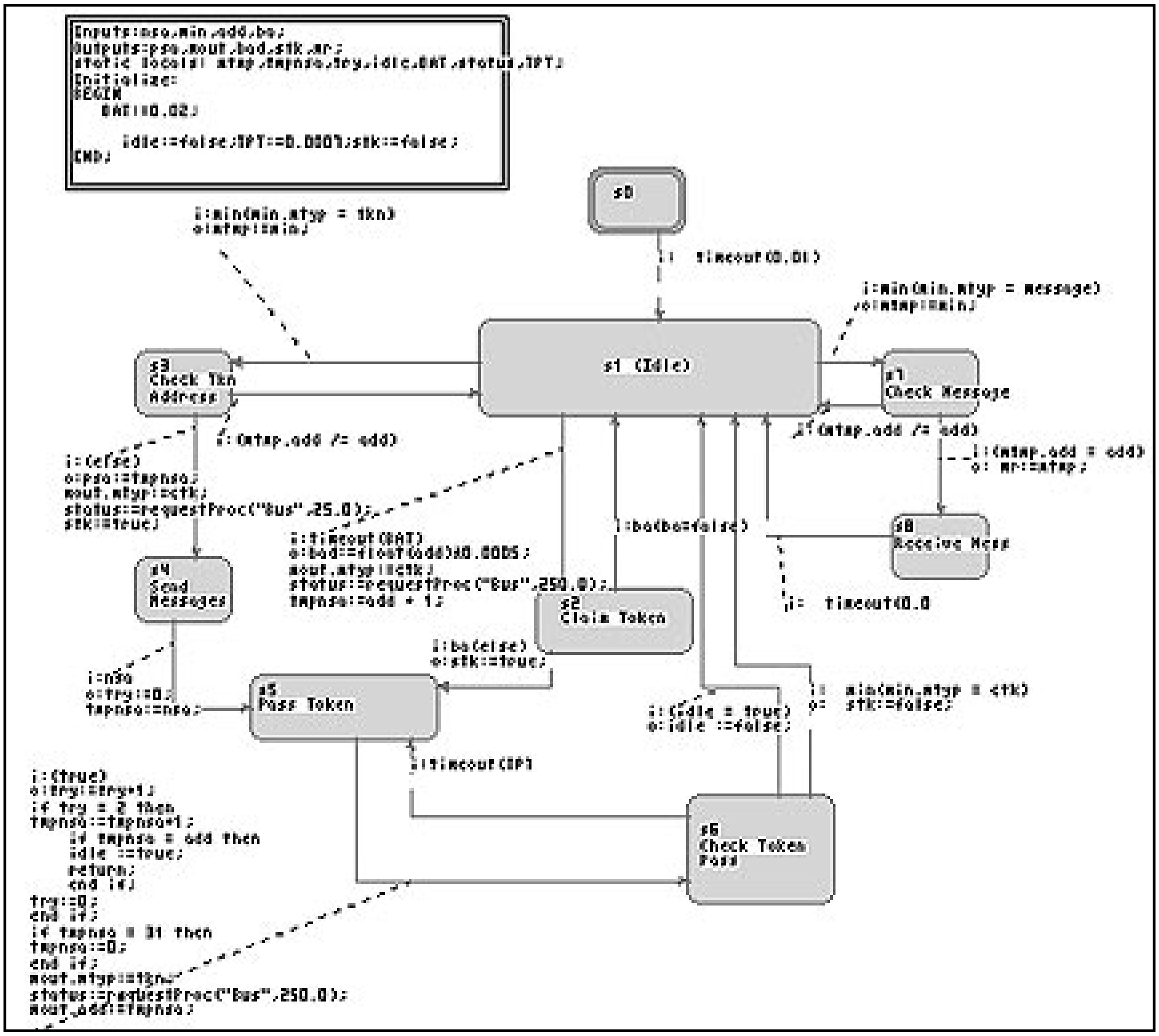

Figure 4. Foresight State Diagram Easily Captures Protocol Specifications

Foresight's functional modeling constructs are a straightforward extension of its high-level performance modeling constructs. This powerful capability (illustrated in Figure 5) enables engineers to mix levels of detail within the same model, and to address the wide range of system design problems which typically arise in the evolution of complex standards and the development of complex systems that incorporate high-speed communication interfaces.

Model-based System Design

Foresight's unique ability to integrate high-level performance analysis with functional design verification is further complemented by its ability to facilitate the downstream design engineering activities.

That is, Foresight's modeling and analysis capabilities can be leveraged during the design activities of high-speed communications based product development. By having an executable model of the standard (or numerous models each at different levels of detail), an engineering organization can reduce the risk associated with complex product development. A clear, unambiguous, and executable model of the communications interface serves as a reference point, enabling new engineers to quickly understand its operation, and facilitating project-wide communication.

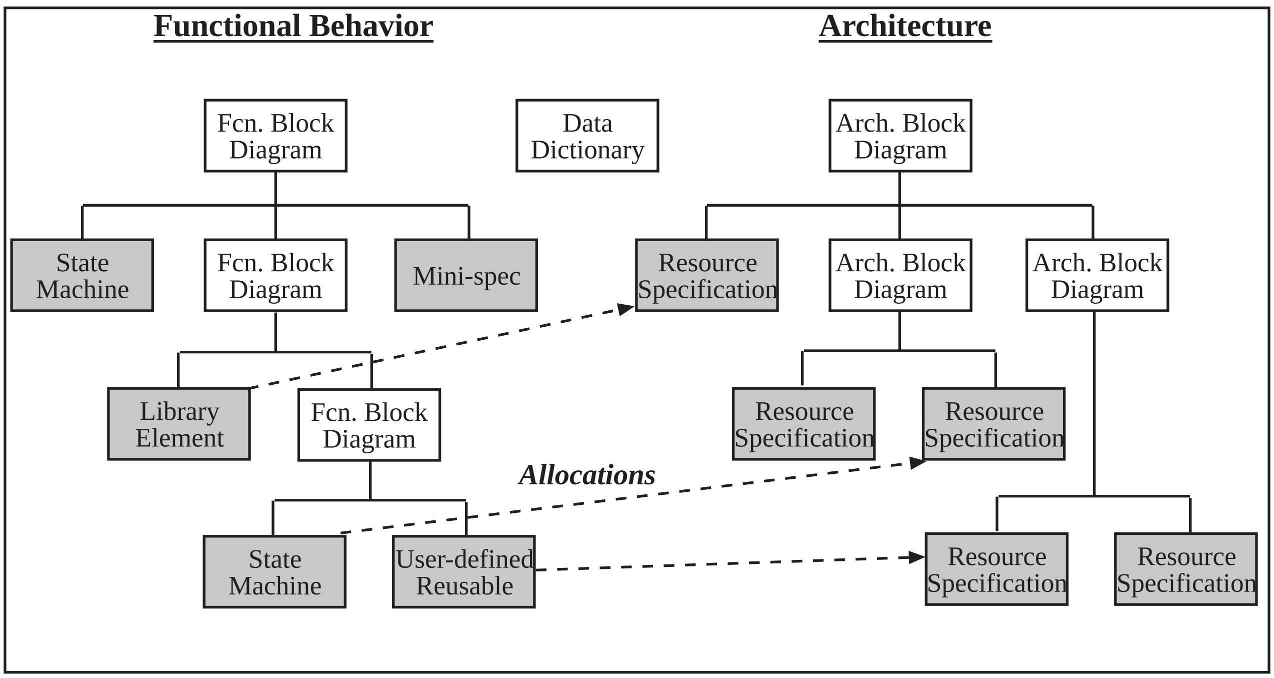

Figure 5. Foresight Integrates Functional Verification with Architectural Design

Foresight's system design facilities extend a communication subsystem model's usefulness beyond that of a reference model. The same functional design verification and architectural performance analysis capabilities that can be applied to interface specifications can equally be applied to product development efforts. Foresight allows product engineers to verify the correctness of a system's functional requirements, and explore various hardware / software architectural alternatives [4], including but not limited to the communications interface implementation.

Additional Foresight system design automation products can be utilized in system design as well. Foresight's cosimulation interface (FS/Bridgeway) allows other (e.g., legacy) models in other tools to be linked to the communication subsystem models in Foresight. Also, Foresight's file I/O and "C" call mechanisms facilitate the rapid integration of Foresight models with external data. Automatic production of “C” code with Nu Thena’s CoderC product supports the development of run-time prototypes. And Nu Thena’s VHDL code generator allows system designers to automatically produce ASIC specifications and testbenches directly from the system model for use in downstream ASIC design and verification, as illustrated in Figure 6.

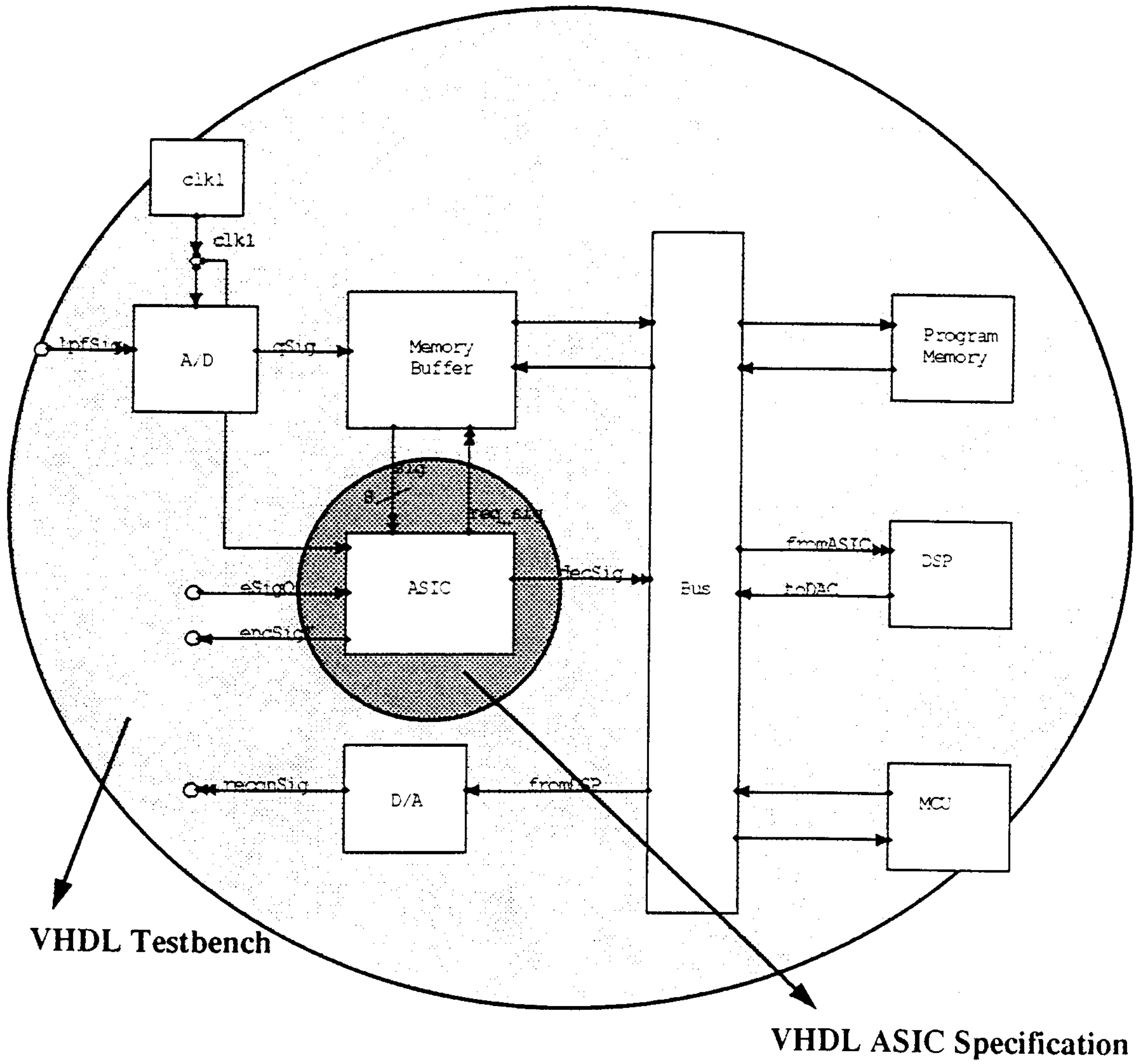

Figure 6. Foresight Design Refinement and automatic VHDL Code Generation

The Foresight system design automation product family can be easily integrated into existing product design work flows. Foresight incorporates interfaces to popular desktop publishing and word processing tools (FS/Doc), human interface prototyping tools (FS/Altia), and requirements traceability tools (FS/RQIF). Additionally, Foresight’s extensibility provides the opportunity to tightly integrate with other third party software tools.

Conclusion

Foresight provides an environment in which high-speed communication standards can be captured as executable models, at multiple levels of detail, to specify clearly and unambiguously product functional and performance specifications. In fact, numerous models already exist and are available to Foresight users. Upon this base of extensible, executable communication models:

- Proposals to modify and/or extend specifications can be rapidly evaluated and rigorously validated, no matter the level of detail required
- New users can quickly understand a communication standard's operation
- High-speed communication based product development cycles can be accelerated

The high-speed communication subsystem design community can benefit from a common modeling and simulation tool set, which enables the development of executable specification models at multiple levels of detail, and which is robust enough to cover the wide spectrum of system design issues that arise during management of evolving standards and their implementation in products. Foresight offers a comprehensive modeling language, where multiple levels of design abstraction can be used to address any type of specification or design challenge - all within an easy to use and extensible system design environment.

References

[1] M. Vertal, "Foresight: System Simulation for System Developers", Proceedings of the 27th Annual Simulation Symposium, IEEE Computer Society Press, La Jolla, CA, April 11-14, 1994.

[2] ANSI X3.230-1994, Fibre Channel Physical and Signalling Interface (FC-PH) Rev. 4.3.

[3] ANSI/IEEE Std 1076-1993, IEEE Standard VHDL Language Reference Manual.

[4] P. Solanti and M. Vertal, "A System Simulation Based Approach for Embedded ASIC Design", Proceedings of the On-Chip System Design Conference 1997, Design Supercon, Santa Clara, CA, January 21-23 1997.