

# Simulatable Specifications and VHDL in Top-Down Design

Mark Thullen and Lynn Driver  
TRW Dayton Engineering Laboratory, Dayton, Ohio  
Mark.Thullen@TRW.com  
Lynn.Driver@TRW.com

## ABSTRACT

*Simulatable Specifications (SimSpecs) have been identified as a key means for rapidly developing high quality electronic product designs. The Air Force's Continuous Electronics Enhancements Using Simulatable Specifications (CEENSS) program has been initiated to explore the application of Simulatable Specification technology to the design and development of Printed Circuit Board/Line Replaceable Module mixed signal electronic products. The CEENSS program is focused upon the following objectives:*

- *The ability to design products which have a very high probability of correctness,*
- *Achieving a radical reduction in the time and cost of product design,*
- *The ability to design products with significantly reduced design obsolescence, and*
- *Achieving a radical reduction in the time and cost for product redesign/retrofit.*

*A major contributor to the time and cost of original product design and design modification/retrofit is the excessive integration and test effort required to fix problems caused by faulty requirements. This integration and test delay is caused by detection of problems, determination of problem causes, and design modifications to correct the problems. The use of formal SimSpecs makes possible the development of complete, consistent, unambiguous requirements specifications which will significantly reduce integration and test problems which stem from requirements problems.*

*The utilization of SimSpecs in product design verification at each level of design will catch a large percentage of design errors in the step where they are introduced rather than deferring them all until integration and test where they are more complex and expensive to correct. SimSpecs are written in VHDL, which allows them to be formal, unambiguous representations, and enables verification through execution.*

*The use of SimSpecs as a primary design capture mechanism enforces the development of technology independent design representations which will significantly enhance users' ability to incorporate new technology into a design without reinventing the whole design. A key approach to producing high confidence designs is to reflect manufacturing and test aspects of the system in the SimSpec and to verify manufacturability and testability of the design as core elements of the product design verification process.*

# **1 INTRODUCTION**

## ***1.1 Statement of the problem***

Several factors are causing a crisis in the development of electronic systems for military and commercial uses:

1. The complexity of electronic systems is increasing much faster than the capabilities of the processes to develop the systems.
2. Economic and competitive pressures do not allow development times to increase as systems become more complex.
3. In the commercial world, Time-to-Market is often more important than development cost.
4. The quality of electronic systems is required to be higher than ever before.
5. Design teams are distributed geographically and in expertise.

These factors are combining to force dramatic improvements in development processes.

## ***1.2 CEENSS as a potential solution***

The CEENSS project is attempting to address the problems described above.

The key strategy of CEENSS is to define a development methodology which is centered on the use of SimSpecs in VHDL. Figure 1 shows a block diagram of the CEENSS environment, whose key elements are the Design Shelf, the Product Data Set (PDS), Design Libraries, and SimSpec tools.

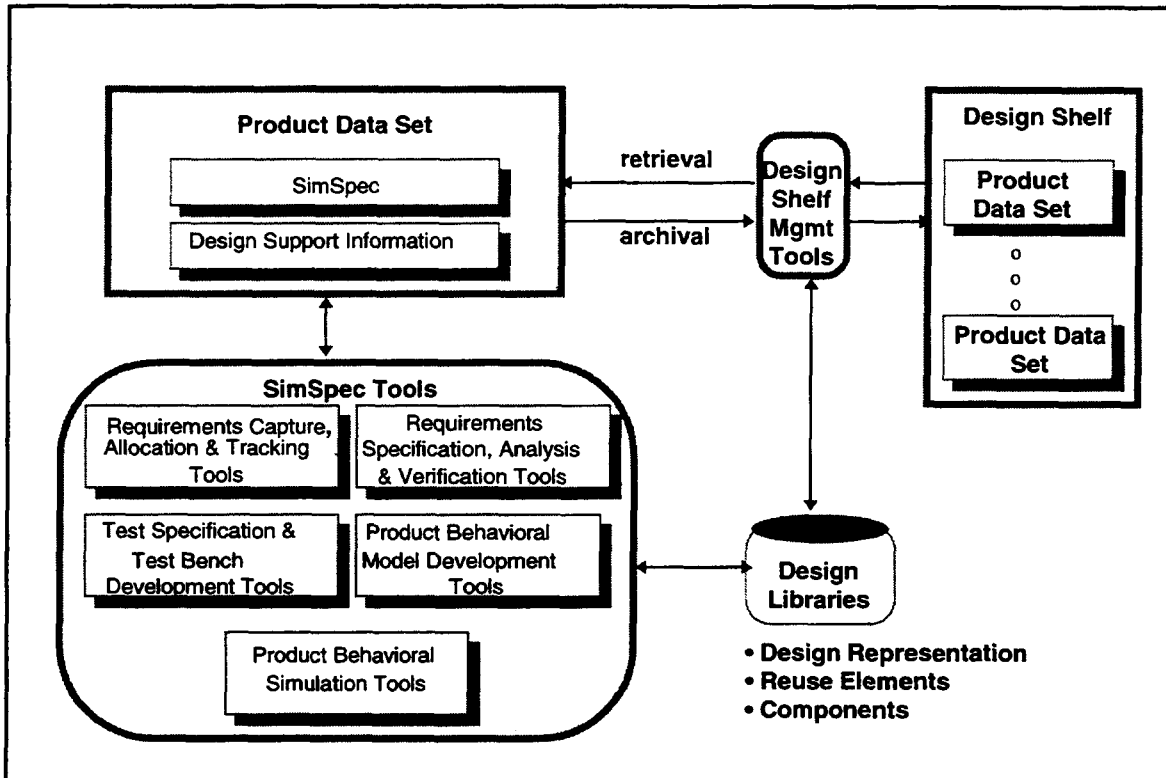


Figure 1: The CEENSS Environment

## 2 OVERVIEW

### 2.1 The need for SimSpecs

#### 2.1.1 Modeling guidelines for multiple abstraction levels

It is highly desirable to model different portions of a design at different levels of abstraction. This allows the designer to scrutinize a component of interest by modeling it in great detail while modeling the remainder of the system at a much higher level. However there are no standards which guide the designer in creating multi-level models. It is an objective of CEENSS to develop guidelines for technology independent models which may form the bases of such standards.

#### 2.1.2 Continuing verification

Because of the high complexity of today's electronic systems, it is usually not feasible to confirm design correctness after the design has been completed. Instead, correctness must be built-in at every step of the design process through continuing verification. However, verification requires two elements: 1) a test, and 2) a standard of correctness against which test results are compared. When a design is still at a very high level, consisting of mostly abstractions with few details, an appropriate test and standards are required, but they are not usually available in the conventional design methodology. Figure 2 shows the conventional development process and highlights problems inherent in it. The shaded problem areas (ovals) will be addressed by CEENSS.

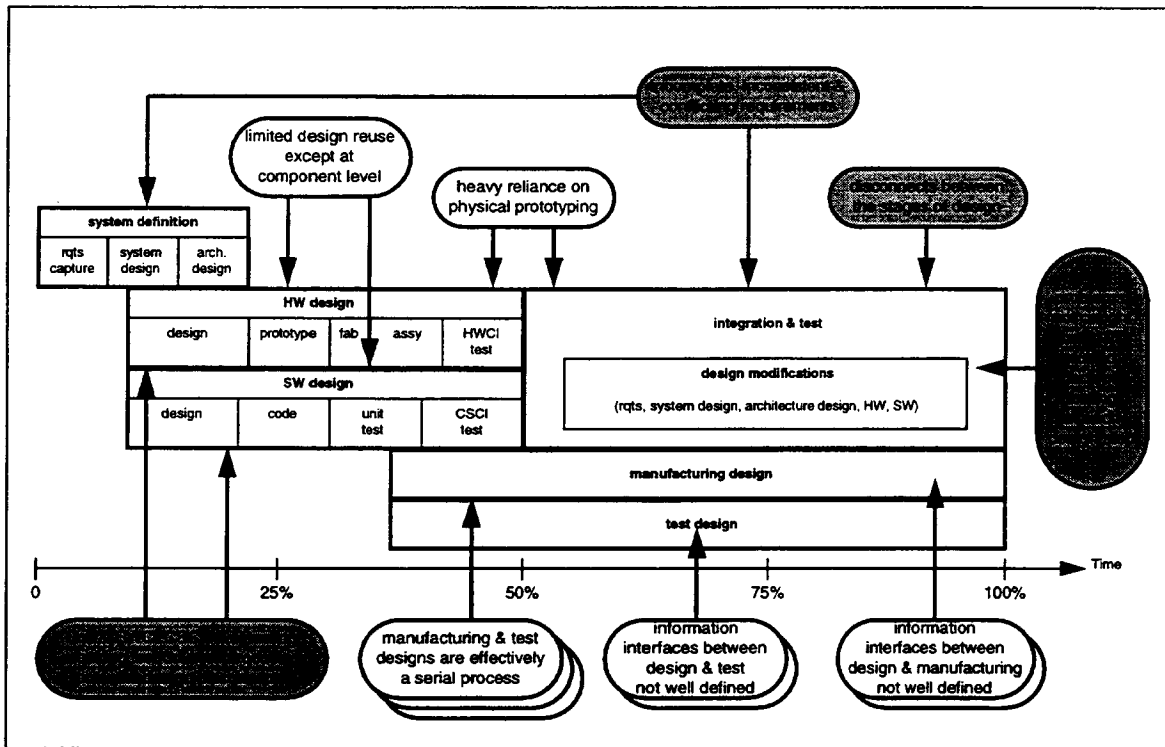


Figure 2: The Conventional Development Process

### 2.1.3 Unambiguous specifications

Few electronic systems are developed by just one person, so communication is an ever-present concern. The common system engineering practice of developing and using written specifications to record and convey requirement and design information was developed to address the need for effective communication. But because specifications are written in natural language they are subject to interpretation errors, ambiguity, inconsistency, and incompleteness. Clearly what is needed is a precise, unambiguous, language for expressing specifications.

### 2.1.4 Technology independence

Technology is changing so rapidly that it is not unusual for advances (new capabilities, or new solutions) to become available before a system is fielded. To exploit this fact, if technology decisions could be deferred as long as possible, the latest and best technology could be applied "at the last minute". This would reduce the likelihood of deploying systems with outdated technology.

For a similar reason, if a partially detailed existing design is used as a starting point for a new system, the latest technology could be inserted into the existing design, provided that it had been expressed in a technology independent manner. Deferring technology decisions as long as practical (and expressing designs in a technology independent form) helps make possible the benefits of flexibility and generality.

## ***2.2 Preliminary definition of SimSpecs***

A SimSpec is a specification in an electronic form which has these attributes:

- It contains the requirements for the product,
- It includes a simulatable, technology independent VHDL behavioral model of the product, and
- It includes a test bench for verification of the product model and validation of product designs at all stages of development.

## ***2.3 SimSpecs are contained in the Product Data Set***

To realize the full benefits of SimSpecs, additional information is required. For example it would be very helpful to provide a general purpose repository of "design intent" information. Such information could be in the form of memos, diagrams, schematics, spreadsheets, databases, etc. Also, supporting specifications such as a design spec, test spec, manufacturing spec, and support spec should be archived and maintained in sync with the SimSpec, to provide helpful related material for all developers.

The need for development information in addition to the SimSpec itself suggests the concept of a Product Data Set, which, as the name implies, is a repository of all data relevant to the product being developed.

## ***2.4 SimSpecs are compatible with Top-Down Design and Design Shelving***

A SimSpec is intended to be dynamic, to be elaborated and refined with the design for the product it describes. In this way a SimSpec's development parallels the development of the product in a systematic, general-to-specific, abstract-to-concrete process. This is a high-level definition of Top-Down Design, one with which the SimSpec is certainly compatible.

SimSpecs are also compatible with and complementary to a concept called Prototyping Plus [1], which can be summarized as a methodology which stresses prototyping as a means of enabling more efficient development of high quality electronic systems. One of the key elements of Prototyping Plus is Design Shelving, which is the practice of collecting and storing all design-related information at appropriate times during the development cycle. The SimSpec, or more properly the PDS, is obviously close in heritage to the Design Shelving concept.

# **3 SUMMARY OF TOP-DOWN DESIGN**

## ***3.1 Definition of Top-Down Design***

Top-Down Design can be loosely defined as design through repeated, gradual decomposition, elaboration and refinement, starting from and driven only by system level requirements (and requirements derived from them).

### **3.2 The case for Top-Down Design**

#### **3.2.1 Top-Down Design is requirement-driven design**

A primary advantage of Top-Down Design is that every function and characteristic of the system can be traced back to a requirement -- a function is not implemented if it can't be so traced. In the downward direction, every requirement is reflected and satisfied in the final system. In other words the resulting system is everything that was specified but nothing more. A more subtle but equally important advantage is that properly performed Top-Down Design is inherently systematic, helping to insure that development is disciplined and reasonably predictable. However Top-Down Design does not guarantee that a correct system will result. To insure correctness more attention must be paid to verification.

The traditional design process departs from Top-Down Design in that the starting point is not always at "the top"; i.e., the new design starts with an old design and the designer chooses which functions to retain and which new functions must be provided. Also, there is often a "bottom-up" flavor, wherein existing low level functions are patched in by making interface changes in a "cut and fit" procedure. Although economy is usually cited to justify this, the real result is that unnecessary functions are often put into the new system because it is faster (and safer) to leave them in than to worry about the side effects of taking them out. The result can be a patchwork of functions of three kinds:

1. Those which can be traced to requirements,
2. Functions which can't be traced to requirements, and
3. Functions which can be traced but which differ slightly in functionality.

It is not hard to imagine the verification difficulties inherent in a mixed development process of this kind.

## **4 DESIGN SHELVING AND THE PRODUCT DATA SET**

### **4.1 Definition of Design Shelving**

#### **4.1.1 The traditional development cycle is iterative and discontinuous**

In the military world major weapon systems often take ten or more years from start of development until the system is deployed. When combined with the accelerating pace of technology advances, it is no mystery why such systems are no sooner deployed than upgrades and modifications are begun.

There are situations in the design process where it would be very convenient to return to a previous design state. Also, there are times when development is interrupted, and it would be convenient to save a snapshot of the current design state. These factors suggest the desirability of Design Shelving.

It is natural then to provide for occasional shelving of intermediate designs, like doing periodic backups in a computer system, that can be retrieved later and used as a point

of departure for inserting newly available technology, or for re-considering questionable design decisions.

It is important to note that Design Shelving is inherently compatible with the spiral development process, wherein development advances incrementally, and usually across the entire “design front.” It is also noteworthy that developers are accustomed to working in an iterative, discontinuous mode, a style which can exploit Design Shelving fully.

#### **4.1.2 Development is elaboration of concepts**

The development process can be thought of as repeated elaboration of concepts, or stepwise refinement [2], moving from “what” (requirements) to “how” (design) until all details have been determined. In this context, the availability of periodic archivals helps to satisfy the need for repeatedly revisiting a previous design state.

#### **4.1.3 The result of development is embodied in documents**

Until the system being developed is actually manufactured, the product of the development effort exists solely in documents. This strengthens the case for Design Shelving, which provides a mechanism for efficiently archiving and controlling documents and related supporting information.

### **4.2 Definition of Product Data Set**

As stated earlier, developers create and need access to various supporting information, not just specifications and design documents. The PDS (section 2.3), including the SimSpec and other design-related material, suggests itself as the natural package of information to be shelved.

An informal definition of a PDS is: the set of all information relevant to the development at a point in time. As shown in Figure 3, this includes a SimSpec, design specifications, design support information, manufacturing specifications, test specifications, and support specifications.

A PDS can also be said to be:

- The collection of design information necessary to enable manufacturing, test and support of the product and future design reuse, modification and retrofit activities,
- Captured in a tool neutral form, and
- Supportive of the Design Shelving concept.

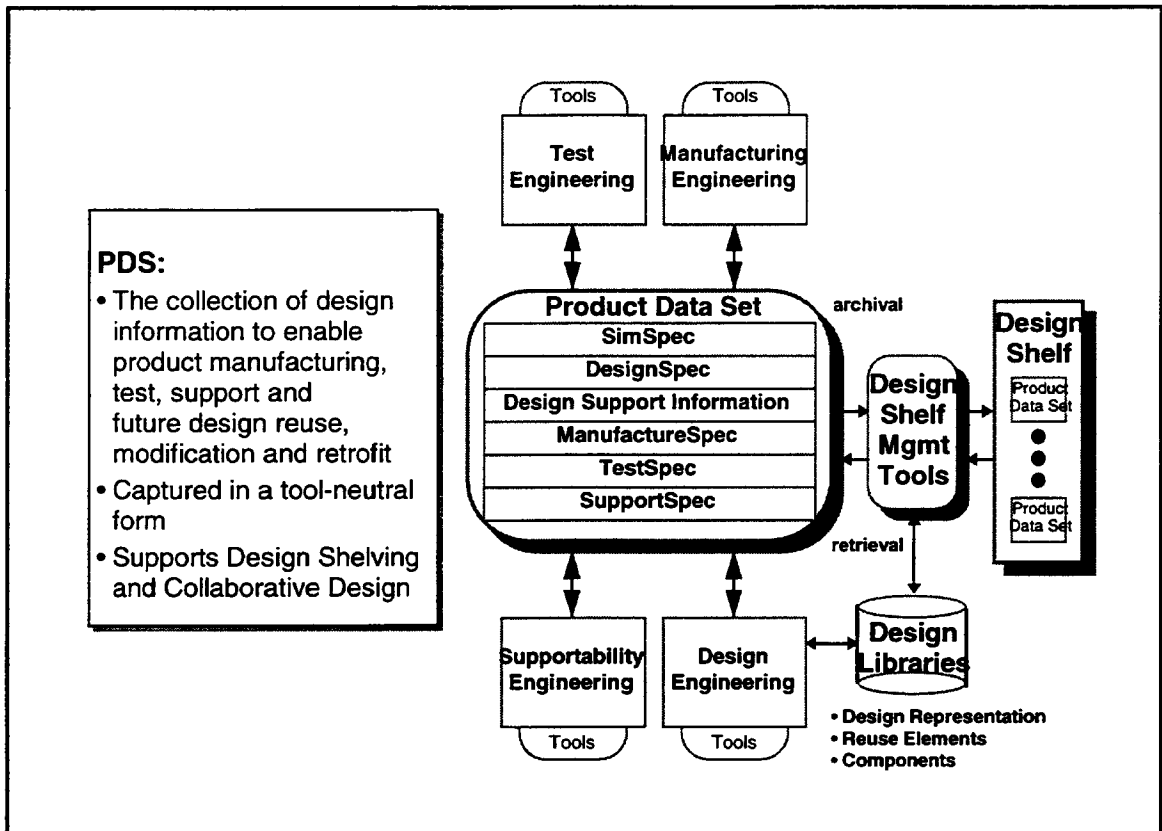


Figure 3: The CEENSS Product Data Set

#### 4.3 Benefits of Design Shelving

Now that Design Shelving and the PDS have been defined, we offer the following benefits of their combined use:

1. They are compatible with spiral development
2. A PDS is a unified repository for the design and design intent
3. A PDS is an efficient package for communication of a design
  - In purchasing/procurement situations
  - Among members of geographically dispersed development teams
4. A PDS helps to enable reuse
  - Having all design related information organized and available in one "package" makes finding design elements easier
5. The electronic format of a PDS allows improved control of design changes
  - Automated configuration control is feasible because of the PDS's electronic format
6. Tool neutrality encourages tool innovation
  - Tool vendors should be more likely to develop new tools because of the PDS's open, standardized format

#### **4.4 Obstacles/problems with Design Shelving**

As with any new process or methodology, there are barriers to adoption and exploitation of Design Shelving and the application of PDS. For example, Design Shelving requires specific tools to handle the detailed mechanics of shelving and unshelving; and because a PDS is ideally tool-neutral, standards and/or conventions for PDS structure and format are needed before tools appear. To encourage development and adoption of PDS-oriented tools, the CEENSS project may develop format conversion utilities that can be used by tools to handle details of PDS formats.

### **5 SIMULATABLE SPECIFICATIONS**

#### **5.1 Detailed definition**

A summary definition of a SimSpec was given above. Here is a more complete, detailed definition in terms of its components:

- The product requirements specification -- this specification is similar to the traditional paper-based requirements specification, but contains more information. The SimSpec is refined and elaborated as the design is refined and elaborated. In the design process top level functions are decomposed into more specialized functions, and top level requirements are then allocated to them. Requirements can also be decomposed in this process. Figure 4 shows the simultaneous functional decomposition and requirements allocation processes.

To satisfy the need for unambiguous requirements, product requirements must be expressed in a specialized, i.e., non-English, language. The CEENSS project is considering various requirements specification approaches for this purpose. One example of such an approach is VSPEC, which extends the VHDL entity declaration to include requirements. [3]

- The product behavioral model -- this is the VHDL behavioral model of the product/system being designed. As the development progresses this model is made more detailed in function to reflect the latest design. Through VHDL simulation a developer can execute this model to simulate the function of the desired system.
- The top-down representation of the design expressed by vertically interoperable VHDL models of various levels for each decomposed sub-function.
- Product design configurations -- these record the design as it is refined and elaborated. Configuration information consists of definitions of all functions (in terms of inputs, algorithms, and outputs) and their associated requirements. As functional decomposition and requirement allocation proceed, the relevant information becomes the latest configuration. (Note: The CEENSS project has not decided on the best method for representing this information.)
- Product test requirements -- these are the requirements for testing the system during development, manufacturing, and deployment. (CEENSS will do further work in this area also, to determine appropriate methods of specifying test

requirements. A good candidate is Test Requirements Specification Language (TRSL), which is being developed under another Air Force contract.)

- The product test bench -- this is another VHDL module; its purpose is to execute in conjunction with the product behavioral model, providing inputs to the product model and accepting outputs from it. Because of its direct interaction with the product model, the test bench interfaces must be maintained to remain complementary with the product model interfaces as they are elaborated and/or modified. Note that there is just one test bench, and not a separate test bench for each product function or sub-function. This reflects the philosophy of always testing the product model as a whole (rather than testing components to be later assembled and integrated into a system) to avoid problems at integration.
- Test vector sets -- these are sets of test data used as input for the test bench. CEENSS plans to use the Waveform and Vector Exchange Specification (WAVES, IEEE 1029.1) as the format/mechanism for test vectors.

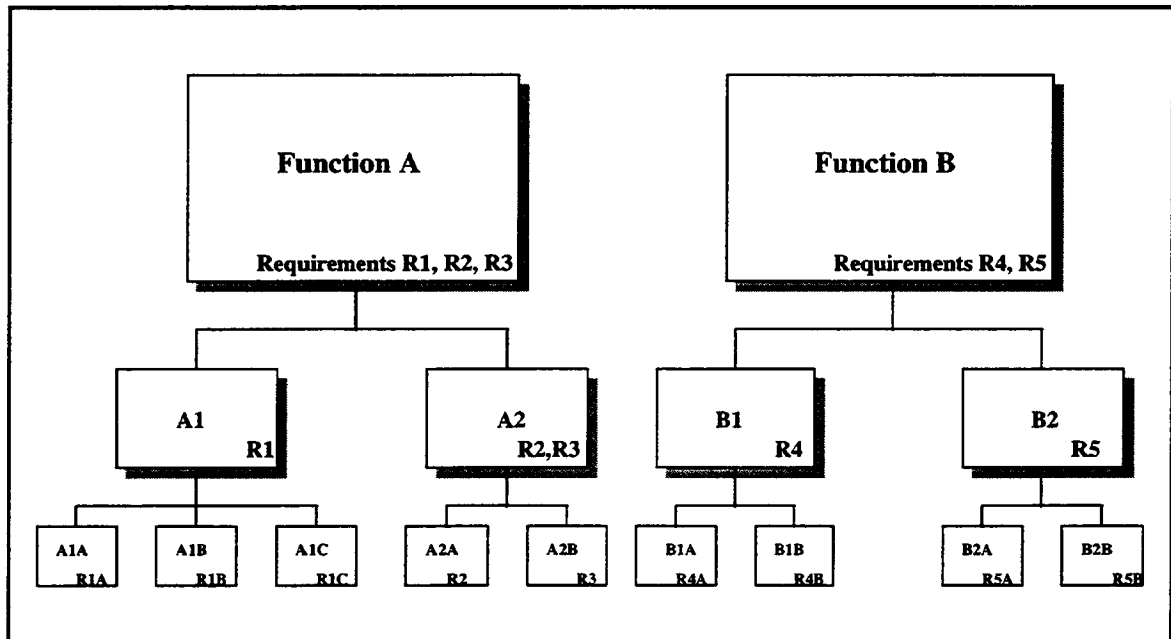


Figure 4: Example of Functional Decomposition and Requirements Allocation

## 5.2. SimSpecs are expressed in VHDL

As mentioned in conjunction with each SimSpec component, SimSpecs are written in VHDL. However, there are specific components of SimSpecs which may not be best represented in VHDL. For example, consider how to express in VHDL such requirements as “the complete system shall weigh not more than ten pounds”, and “the system shall have a Mean Time Between Failure of not less than 1000 hours.” Also at issue is how the emerging mixed signal VHDL standard will impact SimSpecs and their use.

### **5.3 Benefits of SimSpecs**

SimSpecs have the following benefits in the development of electronic systems:

- They enable true Top-Down Design (vertical model interoperability)
- They include verifiable requirements
- They help produce better designs in less time
- They support virtual prototyping and rapid prototyping
- They produce designs with reduced obsolescence
- They make possible more effective re-designs and retrofits
- They support concurrent engineering
- They support geographically distributed workplaces

### **5.4 Further details of SimSpecs are in progress**

The CEENSS project is approximately one year into a four year program. In the next twenty-four months we plan to develop and refine the SimSpec concept, supplying most of the details that are missing as of this writing.

The work done to date has been documented in several forms, most notable of which is the CEENSS Critical Review Package [4]. This and other documents may be requested from US Air Force, Materials Lab, Manufacturing Technology Directorate, WPAFB, OH. Selected materials will also be made available on the Internet.

## **6 CONCLUSIONS**

### **6.1 SimSpecs are the key concept in an improved development methodology**

This paper has presented an overview of a methodology intended to improve the design of electronic systems. Its drafters believe that the methodology offers considerable promise in attaining the desired improvements, primarily because of SimSpecs. SimSpecs attack the requirements problem, and in doing so should reduce Integration and Test time. SimSpecs also provide a practical means for continuous design verification, for enabling vertical model interoperability, for enabling collaborative design, and for supporting reduced technology obsolescence through technology independent design representation.

### **6.2 The CEENSS methodology is consistent with Top-Down Design and Design Shelving**

The methodology combines and applies elements of Top-Down Design, Design Shelving, and the spiral development process, and is therefore compatible with those techniques. When SimSpecs are added to this approach, it has the potential to help alleviate the current crisis in design of electronic systems.

## **7 ACKNOWLEDGMENTS**

The CEENSS project described in this paper is supported by the United States Air Force under contract number F33615-93-C-4304, administered by Wright Laboratory, Manufacturing Technology Directorate, Wright-Patterson AFB, Ohio

## **REFERENCES**

- [1] "Prototyping Plus", Chapter 3 in *Building Future Security Strategies for Restructuring Defense Technology and Industrial Base*; Office of Technology Assessment of the Congress of the US, June 1992.
- [2] Wirth, Nicklaus, "Program Development by Stepwise Refinement," *Communications of the ACM*, 14, No. 4, pp. 221-227.
- [3] Alexander, Perry, Philip Baraona, and John Penix, *VSPEC: A Declarative Specification Methodology for System Synthesis*; ICEHDL '95.
- [4] Thullen, Mark, et al., *CEENSS Critical Review Package*, CEENSS Contract Deliverable Requirement List item number A008, 15-Nov-1995.