Language Design Requirements for VHDL-RF/MW™

John Willis, Ph.D. (jwillis@ftlsystems.com) and Joshua Johnson
FTL Systems, Inc., Rochester, MN 55902
FTL Systems UK Ltd., Chilworth, England SO16 7NP

Abstract **This paper describes design requirements for VHDL-RF/MW[1]. The resulting language design addresses distributed and full-wave interconnect models, frequency-domain modeling and parasitic interactions so as to maintain full conceptual and representational compatibility with VHDL, VHDL-AMS, Verilog, Verilog-AMS and SPICE.**

I. Introduction

Hardware description languages (HDL), such as VHDL and Verilog, have made enormous impact on the design community. Digital designers have benefited most, largely replacing schematic design views of complex systems. Analog HDL use is growing rapidly in markets such as telecommunications, avionics and even consumer products.

However, HDL use is extremely limited among designers developing inherently dense products with significant 1 GHz to 100 GHz spectral components. Lack of HDL impact in this regime increases the development cost, development time and product reliability. These needs motivated creation of VHDL-RF/MW.

As a secondary benefit, designers working at lower speeds will find subsets of VHDL-RF/MW convenient and valuable in their work. Frequency-domain component modeling is an important example, with applicability to designs such as filters or control loops running at modest frequencies and no significant parasitics.

This paper focuses on the design requirements for VHDL-RF/MW. Several companion papers summarize the resulting language design. John Willis (one of the authors) chairs the IEEE Design Automation Standards Sub-Committee on VHDL extensions for RF and MW.

II. Requirement Summary

Several years were invested to establish the practical requirements for a successful VHDL-RF/MW language design and implementation. The VHDL-RF/MW Requirements Phase identified ten primary needs which must be satisfied for VHDL-RF/MW to be successful:

1. Integrate distributed components
2. Integrate frequency-domain components
3. Integrate parasitic contributions
4. Maintain base language compatibility
5. Facilitate multiple interacting domains
6. Encapsulate design technology
7. Enable pre-processing to HDL-AMS
8. Exploit native implementations
9. Enable broad availability
10. Pursue eventual IEC standardization

III. Integrate Distributed Components

VHDL, VHDL-AMS, Verilog, Verilog-AMS and SPICE all utilize a lumped-parameter modeling abstraction. At any instant in time, signals, shared variables, quantities and terminals all have a unique, identifiable value (albeit having a complex type or nature).

While efficient to model and simulate, a lumped parameter abstraction does not completely capture behavior of a physical systems such as transmission delay, reflections from impedance mis-match or electro-magnetic interference. As physical density and operating frequency increase, approximations made to achieve an efficient lumped parameter model often mask deviations between the model behavior and behavior of the realized, actual system. These deviations commonly result in system failure, sub-optimal system reliability or sub-optimal power/cooling properties (resulting from un-intentional energy transfer). **VHDL-RF/MW must incrementally capture various non-lumped and full-wave component models.**

IV. Frequency-Domain Components

Components within a system are often more readily modeled (and simulated) in the frequency domain, such as a transfer function. However the simulation cycles for VHDL, VHDL-AMS, Verilog, Verilog-AMS and SPICE are fundamentally specified in the time domain (using a sequential imperative or constraint paradigm).

1. VHDL-RF/MW is a trademark of FTL Systems Inc. and FTL Systems UK Limited.

0-7803-7239-5/02/$10.00 © 2002 IEEE

VHDL-AMS introduces significant capabilities for transducing the time domain into the frequency domain, however it does not support integrated modeling where the component model is in the frequency domain. Harmonic balance techniques are a significant and well-accepted approach to frequency domain simulation.

VHDL-RF/MW needs to address both frequency domain modeling in the context of VHDL-AMS and Verilog-AMS as well as the relationship between frequency-domain models and the simulation cycle. Following the precedence of the under-specified VHDL-AMS analog solver, the VHDL-RF/MW language design does not specify the implementation of the coupling mechanism between the frequency and time domains.

V. Integrate Parasitic Contributions

Interconnects conventionally represent the primary, intentional couplings within a design such as a deposited trace or fluidic coupling. However, realization of a model into an actual design results in many un-intentional (parasitic) couplings between interconnects, between interconnects and components and between interconnects and the external environment (other systems).

Un-intentional couplings result from parasitic contributions to explicit or implicit terminals. While parasitic contributions are noise to the primary design flow, they can be very significant to the functionality and reliability of a design. **Therefore when such contributions are relevant to function and reliability, a VHDL-RF/MW design flow must recognize the appropriate level of modeling detail, then simulate the appropriate parasitic interactions.**

VI. Base Language Compatibility

VHDL, VHDL-AMS, Verilog, Verilog-AMS and SPICE represent the backbone of contemporary hardware description language design. Several hundred thousand designers have each spent one or more years developing expertise and skills in the use of these base languages. Even beyond the billions of dollars invested in tool implementations, the investment in designer expertise *must* be preserved in order for VHDL-RF/MW to succeed. Departure from this installed base was a major factor in the failure of a previous microwave hardware description language effort, MHDL.

Designers of VHDL-RF/MW recognized that VHDL-RF/MW would need to maintain both conceptual and representational compatibility with VHDL, VHDL-AMS, Verilog, Verilog-AMS and SPICE to become successful. However VHDL-AMS, Verilog-AMS and SPICE enjoy radically different representation and significantly different conceptual approaches, presenting an essential and challenging obstacle to be overcome.

VII. Multiple Interacting Domains

Relative to earlier, structural mixed signal design languages, contemporary VHDL-AMS and Verilog-AMS languages are constructive in the sense that they support definition and interaction of models representing many different domains. An VHDL-AMS model may include electronic, thermal, mechanical, fluidic and other component models, providing an important capability for system simulation. Significantly, new domains and realization technologies may be readily added without change to the base HDL.

While domain and technology extensibility are important for AMS, such extensibility becomes more important with VHDL-RF/MW. VHDL RF/MW models may represent not only a wide range of realization topologies but also operating frequencies and operating environments. **VHDL-RF/MW language extensibility is critical.**

VIII. Encapsulate Design Technology

Expertise required to understand interconnect technology physics and appropriately model such physics in VHDL-RF/MW is often distinct in a design team from the expertise used for software, digital, analog or physical design. **Therefore it is important that VHDL-RF/MW modeling detail be effectively encapsulated so that designers with minimal VHDL-RF/MW technology background may semi-transparently utilize technology modeling setup by a technology specialist.**

IX. Pre-Processing to VHDL-AMS

Even good language extensions are difficult for design teams to adopt when tool implementations are scarce. Conversely it is difficult for tool developers to create tools when demand is uncertain. Success for VHDL-RF/MW requires cost-effectively breaking this loop.

Pre-compilation of the new language extensions, such as VHDL-RF/MW, into an existing language domain, such as

VHDL-AMS, can break such cycles. However the requirement to design a new language extension so that it can be completely pre-processed into an existing language usually constraints the language design, resulting in a less efficient extension. This inherent conflict has been explored in the C++ extensions from C and the OO-VHDL extensions from VHDL. **Following the resolution taken by C++; VHDL-RF/MW provides a useful subset which can be pre-compiled to yield VHDL-AMS source code.**

X. Exploit Native Implementations

Native implementations of VHDL-RF/MW enjoy additional constructs as well as opportunities for performance and capacity optimization. **Language functionality added to the VHDL-RF/MW subset must add significant designer capability.** Capacity and performance results from explicitly denoting constructs to the VHDL-RF/MW compiler to enable special case compilation more readily than if the compiler need discover the special case.

XI. Enable Broad Availability

Designers are most receptive to new design methodologies and languages when in an academic setting; the pressures of commercial design schedules do not intrinsically provide an incentive to adopt new practices. **Therefore a wide range of implementations and supporting texts must be available, ranging from university texts and implementations suitable for home use to high-end implementations supporting large scale design teams.**

XII. Pursue IEC standardization

Once a design language extension has popular support, it may be more readily and efficiently standardized. Standardization opens the language up to a larger body of helpful critics. Experience has shown that efforts to standardize a language before achieving broad availability and experience leads to an ineffective (such as MHDL) or excessively long (such as the ten years spent on VHDL-AMS) process. **VHDL-RF/MW employs the use before standardization paradigm.**

XIII. Conclusions

A two year requirements gathering effort distilled down to a set of ten basic design requirements summarized in this paper. The design requirements resulted in VHDL-RF/MW, described in companion papers.

XIV. References

IEEE Std. 1076-1 IEEE Standard VHDL Language Reference Manual with Analog and Mixed Signal Extensions, 1999.

OVI Verilog-AMS Draft Language Reference Manual.

Paulo J. C. Rodrigues, Computer Aided Design of Nonlinear Microwave Circuits, Artech House Publishers, 1998.

Turlington, Thomas R. Behavioral Modeling of Nonlinear RF and Microwave Devices, Artech House Publishers, 2000.

XV. Acknowledgments

This work was funded in part by the DARPA/MTO NeoCAD program under US Army Aviation and Missile Command Contract DAAH01-01-C-R163, by FTL Systems Inc. (US) and under separate funding by FTL Systems UK Ltd. (UK). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

We appreciate input from the design community during the requirements generation phase. Requirements input came from the IEEE Design Automation Technical Committee (DATC), the Design Automation Standards Committee (DASC) subgroups on VHDL-AMS and VHDL-MW.

Input from commercial and academic sources are especially appreciated including (but not limited to) the Mayo Foundation Special Purpose Processor Development Group (SPPDG), the Air Force Research Laboratory (Wright and Rome Sites), HP/Agilent, and Hal Carter at The University of Cincinnati. Input from FTL Systems teams in Rochester, Minnesota, US and at the Chilworth Research Centre, Hampshire, UK are appreciated.