IEEE DASC P1076.1 Working Group

http://www.eda.org/vhdl-ams/

Working Group Meeting
September 14, 2010

Agenda

- ♦ **Call to order**
- ♦ **Approval of agenda**
- ♦ **Administrative issues**
  - Minutes of last meeting
  - Publicity
  - Review of IEEE patent policy
  - 1076.1.1 and 1076.1 updates
- ♦ **Project overview and status**
- ♦ **Next meeting**
- ♦ **AOB**
- ♦ **Adjourn**

Administrative Issues 1

- ♦ **Approval of minutes of WG meeting held August 10, 2010**
- ♦ **Publicity**
  - Publication in Simulation News in Europe
    - Missed deadline August 2010
    - Looking for next available slot
  - DASC column in IEEE Design and Test magazine expected to start late 2010 or early 2011
    - WG chairs will be asked to contribute
- ♦ **Review of IEEE patent policy**
  - http://standards.ieee.org/board/pat/pat-slideset.pdf

Administrative Issues 2: Standard Revisions

- ♦ **1076.1.1 revision**
  - Before ballot, LRM must go through Mandatory Editorial Coordination
    - May require some updates to LRM
    - Plan is to get this done before the end of this month
- ♦ **1076.1 revision**
  - 1076.1 PAR draft
  - PAR held up by questions regarding copyrights
    - Do we have to get Copyright permission from organizations that gave it to 1076?
    - Issue is being debated between IEEE and DASC

Revision of IEEE Std 1076.1

- **Scope**
  - Synchronize with IEEE Std 1076-2008. This will bring, among other features:
    - VHPI-AMS
    - IP protection, encryption
    - Genericity, i.e. the ability to write a model that is independent of the energy domain (e.g. a generic resistance model)
  - Integration of IEEE Std 1076.1.1 Packages for Multiple Energy Domain Support
  - Selected language enhancements prioritized by survey among users and other interested parties
    - Frequency domain modeling
    - Package supporting table-driven modeling
    - Package supporting vector/matrix operations
    - Other projects not yet started
      - Mixed Netlists
      - PDE Support
      - Uniform use of SPICE Models

VHPI-AMS: Overview

- **The VHDL Procedural Interface is a application programming interface to VHDL tools.**
- **VHPI allows an external application to access information about a VHDL model during analysis, elaboration and execution.**
- **This enables:**
  - the creation of tools such as debuggers or profilers for VHDL models
  - support foreign models, which are models written in part (foreign subprograms) or in their entirety using a standard programming language like C.

VHPI-AMS: Scope

- **The VHPI-AMS is an extension of the VHPI defined in IEEE 1076-2008.**
  - Extension will include modification/addition to machine readable package.
- **The VHPI-AMS will not extend the digital information model to IEEE 1076-2008.**
- **Derived requirements that will be supported:**
  - Standard utilities provided in VHPI extended for VHDL-AMS objects. Examples include printing, comparing values, error handling.
  - Static design data provided in VHPI extended for VHDL-AMS objects.
  - Object access to query new objects such as free quantities and through/across quantities on branches.
  - Extend VHPI foreign interface to support VHDL-AMS objects.
  - Simulation interaction and control for stop/start
  - Extension of VHPI Tool Execution specification (Section 20 in IEEE 1076-2008) to consider the analog/mixed-signal simulation cycle

VHPI-AMS: Scope Restrictions

- **Derived requirements that will be considered**
  - Modifications of the value of a quantity values (setting values but not forcing) may be supported in this version. This is similar to Section 22.5 Updating object values in IEEE 1076-2008.
- **Derived requirements that will not be supported:**
  - Modification of generics on elaborated design (which may cause re-elaboration) during restart. This could be done by a tool in an implementation specific fashion and does not require support in VHPI.
  - Modification of topology during simulation (forcing values, shorting and opening nodes) and adding/deleting elements.
  - No special semantics for meta simulation (monte carlo, worst case, etc…) will be added. These may still be implementable with the core VHPI and above extensions.

VHPI-AMS: Resources

♦ **Resources**

- M. Vlach: Programming Interface Requirements for an AMS Simulator, ISCAS’2004
    - This paper provides an overview of requirements for VHDL-AMS VHPI.

VHPI-AMS: Status

♦ **Status**

- Review of Verilog-AMS VPI extensions
- Initial analysis of extensions to information model
- Investigating possible donation

♦ **Next Steps**

- Review existing VHPI-AMS functions and identify modifications or additions
- Investigating technical feasibility of writing to quantities and related use cases
- Define detailed requirements based on these investigations

1076.1.1 Integration: Overview

♦ **Purpose**

- IEEE Std 1076.1.1 is a companion to IEEE Std 1076.1 that provides a collection of standard definitions to support the creation of models in multiple energy domains: electrical, translational, rotational, thermal, and others.
- Just as IEEE 1076-2008 integrated the previously separate standards defining the packages std_logic_1164, math_real, math_complex, numeric_bit, numeric_std, the plan is to make the 8 packages defined by IEEE Std 1076.1.1 an integral part of IEEE Std 1076.1.
- Existing 1076.1.1 will then be discontinued.

1076.1.1 Integration: Scope

♦ **Integrate 1076.1.1 text into new 1076.1 LRM**

♦ **Add 1076.1 packages to machine readable packages (library IEEE)**

- FUNDAMENTAL_CONSTANTS, MATERIAL_CONSTANTS
- ENERGY_SYSTEMS, ELECTRICAL_SYSTEMS, MECHANICAL_SYSTEMS, RADIANT_SYSTEMS, THERMAL_SYSTEMS, FLUIDIC_SYSTEMS

♦ **Review all descriptions/definitions against IEEE SI 10-2002 and IEEE PSI-10/D1.May 2010.**

♦ **Investigate adding support for dimensions and consistency of dimensions on connections. This could provide support for external dimensional consistency but not part of standard.**

- A dimension refers to the physical value being modeled and not the unit used to represent it. Example of length being a dimension but foot or meter being a unit.

1076.1.1 Integration: Status

- **Status**
  - Waiting on new LRM creation to integrate existing 1076.1.1 text and packages and to do the SI review.
  - Investigated early white papers on dimensional analysis for 1076.1.1.
  - Reviewed Modelica to understand its use of units and dimensions.
  - Defined Scope and Rational for Dimension support.
- **Next Steps**
  - Complete existing 1076.1.1 integration when new LRM format is reworked.
  - Define detailed requirements for Dimension support.

Frequency Domain Modeling: Overview

- **Purpose**
  - Allow a user to write a model whose frequency domain behavior depends on the simulation frequency
  - Scope is support for small-signal behavior
    - No intent to support harmonic balance or other algorithms that are sometimes called large signal AC.
- **Restrictions**
  - A large signal model for DC and time domain must still be defined, as there is no closed form transformation of a frequency domain model into time domain.
- **Resources**
  - J. Haase, E. Hessel, H.-T. Mammen: Proposal to Extend Frequency Domain Analysis in VHDL-AMS, FDL'09
    - This paper provides some of the mathematical background. Emphasis is on frequency domain behavior described by a table.

Frequency Domain Modeling: Status

- **Status**
  - Properties/limitations small-signal algorithms understood
    - Mathematical background
    - Implications of changing equations for frequency domain
    - Algorithmic changes in an implementation are minor
  - Changes to LRM minor
  - Guidelines for writing models with frequency-dependent behavior understood
    - How to write a model to get the expected behavior
    - Modeling style is imposed by small-signal algorithms
  - Findings documented mostly in meeting slides
- **Next steps**
  - Write-up of all aspects of frequency domain modeling support
  - Expose to WG for approval

Table Driven Modeling

- **Subcommittee members**
  - Ernst Christen
  - Joachim Haase
  - Arpad Muranyi
  - Alain Vachoux
- **Purpose**
  - Measured and simulated data are gathered into tables.
  - Tables include, possibly multidimensional, data points.
  - They are indexed by a number of independent variables and all necessary information to properly interpret and use these data points.
  - Table using may be mandatory in the following cases:
    - No analytical model exists
    - To achieve faster simulation
- **Scope**
  - Definition of VHDL-AMS data structures and functions that support this functionality
  - Development of VHDL-AMS package(s) that will serve as reference implementation
  - The development of an optimized implementation is out of the scope.

Table Driven Modeling: Plan of Activities

- **Schedule**
  - Phase 1: Establish the "Requirements for Table Look-Up Modeling with VHDL-AMS"
  - Phase 2: Discussion of VHDL(-AMS) implementation aspects
  - Phase 3: Definition of VHDL(-AMS) function and package headers
  - Phase 4: Reference implementation of functions
  - Phase 5: Test of the reference implementation
  - Phase 6: Finishing of documentation - proposal for the standard revision
- **Timeline**
  - Begin in June 2010
  - Expected end in summer 2011
- **State of affairs**
  - Phase 1 finished
  - Requirement document to be submitted to the WG for discussion and vote by the end of the week

Table Driven Modeling: Requirements (1)

- **Description of functions that shall be supported**
  - Multidimensional real-valued functions $f: R^n \to R,\ \ y = f(x_1, x_2, \ldots, x_n)$
    - n ≤ 4 shall be supported.
    - n > 4 should be supported.
    - Data points are given on a grid.
    - It is not required that the grid is fully populated.
    - It is not required that the data points are ordered.
  - One-dimensional complex-valued functions $f: R \to C,\ \ y = f(x)$
- **Interpolation of given data points**
  - Interpolation methods:
closest data point, piecewise constant,
piecewise linear, quadratic spline, cubic spline
  - Extrapolation methods:
    - same (as interpolation), constant, linear, periodic
    - none (failure if data point outside table is accessed)

Table Driven Modeling: Requirements (2)

- **Table Data Sources**
  - Storage of data points using arrays shall be supported.
  - Storage of data points using files shall be supported.
  - The following file formats should/may be supported
    - Verilog-AMS
    - Touchstone
    - IBIS
    - Excel CSV

Vector/Matrix Support: Plan

- **Call for participation to form the subcommittee**
  - 4 – 5 people joined
- **Develop requirements (work-in-progress)**
  - Cooperated with other related subcommittees to define the basic requirements
  - Develop detailed requirements based on the basic requirements (will ready in September-October)
- **Present detailed requirements to working group for approval (October 2010)**
- **Develop Technical solution from detailed requirements (3-4 months)**
- **Present the solution to working group for approval (Jan-Feb 2011)**
- **Review LRM changes implemented from LCS for accuracy and clarity (2 months)**
- **Completion of work (May 2011)**

Vector/Matrix Support: Requirements

- ♦ **Matrix/vector operations are defined as reference in standard package [required]**
  - Implementation should produce the same result as definition but does not need to exactly follow the definition.
- ♦ **Matrix definition**
  - Real and complex type [required]
  - Non-floating data type [required]
  - Block matrix notation [required]

Vector/Matrix Support: Requirements (2)

- ♦ **Matrix/vector construction**
  - Identity matrix [required]
  - Zero matrix [required]
- ♦ **Matrix/vector slicing [required]**
  - v(i : j) -- a vector of ith to jth element in vector v
  - A(: , j) -- a vector of jth column in matrix A
  - A(r, i:j) -- a vector of ith to jth element in rth row in matrix A
- ♦ **Supported mathematical operations:**
  - +, -, *, /, ^ for matrix element [required]
  - Dot product for matrix/vector [required]
  - Transpose, conjugate transpose (complex), complex conjugate (complex) for matrix/vector [required]
  - Determinant for matrix/vector [required]

Vector/Matrix Support: Requirements (3)

- Matrix power [required]
- Matrix exponential [required]
- Norms (1, 2, infinite) for matrix/vector [required]
- Trace for matrix [recommended]
- Kronecker Product for matrix [recommended]
- Dimension check in all ops. [recommended]
- Inverse [optional]

- ♦ **Annotation of matrix/vector [recommended]**
  - Supply additional information such as matrix property (triangular, banded) or other user-defined property to be used along with the matrix.

Vector/Matrix Support: Requirements (4)

- ♦ **Matrix definition**
  - Supports matrix of (real and complex) [required]
  - Supports matrix of (non-floating data type) [recommended]
  - Supports block matrix notation (matrix in matrix)
- ♦ **Type checking for quantity vector/matrix [recommended]**
- ♦ **Matrix/vector operator definition in standard package [required]**
  - Implementation should produce the same result as definition but not exactly follow it.

Next Meeting

- **Meeting schedule**
  - Meeting schedule once per month, mostly web
  - Always announced at www.eda.org/vhdl-ams
  - Next meetings:
    - Tuesday, October 12, 2010, 8am PDT
    - Tuesday, November 9, 2010, 8am PDT (tentative)