VHDL-200X & The Future of VHDL

by
Jim Lewis
Director of VHDL Training, SynthWorks Design Inc
Jim@SynthWorks.com

http://www.SynthWorks.com

The Future of VHDL

SynthWorks

With all the media hype about languages such as Verilog/SystemVerilog, Vera, and specman e where does the future of VHDL lie? 2003 has heard many claims about VHDL being "The New Latin" and it is dead. Fortunately these claims were made by people or companies who have very little interest (or market share) in VHDL and are looking to push the market in their direction.

The VHDL designer and vendor community are actively working on revisions to both VHDL and the packages that support the language. These revisions are integrating the newest features of verification languages and assertion languages as well as adding features such as a programming language interface (VHPI) and a simulation control interface. In addition, changes are being made to improve performance and ease of use.

To ensure wide support of the new features, EDA vendors are being actively engaged to participate in the standardization efforts and only features that have both designer and vendor support will be integrated into the language.

The Future of VHDL

SynthWorks

Significant enhancements are currently being made to VHDL.

This paper summarizes the efforts being made by the following IEEE working groups:

Group	Website
VHDL-200X effort	http://www.eda.org/vhdl-200x
IEEE 1164	http://www.eda.org/vhdl-std-logic
IEEE 1076.3/numeric std	http://www.eda.org/vhdlsynth
IEEE 1076.3/floating point	http://www.eda.org/fphdl
IEEE 1076.6	http://www.eda.org/siwg

Caution:
All activities here are work in progress.
Some items presented may change

VHDL-200X

SynthWorks

Very little has changed in VHDL since the first standardization in 1987 and its first revision in 1993. During this time, design and verification complexity has grown. New methods and languages have been introduced to help in the short term.

The VHDL 200X effort kicked off last February at DVCon 2003 with the intent of bringing the new language features needed into a single language supported by numerous vendors.

VHDL-200X, Goals

SynthWorks

Enhance/update VHDL to improve performance, modeling capability, ease of use, verification features, simulation control, and the type system.

Maintain VHDL style, nature, and backward compatibility

Leverage industry efforts

- Spring board off of efforts by PSL assertions, Verisity E, Vera, and SystemVerilog

Focus on features sponsored by both users and vendors to ensure quick adoption.

VHDL-200X

SynthWorks

VHDL-200X is being developed in a time phased effort.

- The first phase is called Fast Track and is intended to be completed in early 2004.
- The remainder of the work will be sorted in a priority basis and will be developed in one or more following revisions.

Each proposal will be prototyped to some degree:

- Minimally users to provide tests and examples
- EDA vendor(s) prototype the change (preferred)

Ensures that the features are both cool and useful.

VHDL-200X, Subgroups

SynthWorks

Work on VHDL has been tasked to the following subgroups:

- Fast Track
- Performance
- Modeling and Productivity
- Testbench/Verification
- Assertions
- Data Types and Abstraction
- Environment

Each group is supported by its own separate reflector. If you want to keep up with all activities, sign-up for the individual email reflectors as well as the vhdl-200x general reflector.

VHDL-200X, Fast Track

SynthWorks

Goals: Make critical updates to the language to support other standards groups (such as IEEE P1164, IEEE P1076.3, and Assertions).

A few items on the list are:

- Unary Reduction Operators
- Array/Scalar Logic Operators
- Hierarchical signal access through a package (like Signal Spy from MTI)
- Formatted IO: hwrite, hread, owrite, oread …
- String Conversions: to_string, to_hstring, to_ostring …
- Permit conditional and selected signal assignment to be used in sequential code (processes & subprograms)

All items on the list are required to be non-controversial and are easy to scope the effect of the change.

VHDL-200x, Performance

SynthWorks

Goals: Make language changes that facilitate enhanced tool performance, primarily for, but not only for simulation.

VHDL-200x, Modeling and Productivity

SynthWorks

Goals: Improve designer productivity through enhancing conciseness, simplifying common occurrences of code, and improving capture of intent. Facilitate modeling of functionality that is currently difficult or impossible.

A few items on the list are:

- Process sensitivity list abbreviation via keyword "all" or combinational
- Case expressions.
- Don't care in case targets
- Read output ports
- Permit expressions to be mapped to signal ports of entities and subprograms.
- Case/If Generate

VHDL-200x, Testbench and Verification

SynthWorks

Goals: Language enhancements that ease the job of the verification engineer. Give VHDL similar functionality to Vera and Verisity E.

A few items on the list are:

- Random value generation w/ optional & dynamic weighting
- Associative arrays
- Queues/FIFOs
- Memories implementation and loading & dumping

VHDL-200x, Assertions

SynthWorks

Goals: Define support for temporal expressions and assertion-based verification in VHDL. Consider formal, synthesis and coverage implications.

Approach: Exploit work of others.

Likely to integrate PSL (from Accellera and SystemVerilog) into VHDL.

VHDL-200x, Data Types and Abstractions

SynthWorks

Goals: Enhancements centered on the type system. Higher abstraction level constructs (interfaces)

A few items on the list are:

- Object-orientation
- Greater than 32-bit range for integers (infinite range)
- Sparse / Associative Arrays
- Interface Objects
- Arrays of unconstrained arrays.
- Record templates with unconstrained fields
- Variant records / unions with bit-level mapping
- User-defined floating point mantissa/exponent
- User-defined positional values of enum literals

VHDL-200x, Environment

SynthWorks

Goals: Simulation control environment. Standard interfaces to other languages. Additional support packages.

A few items on the list are:

- Direct C and Verilog calls with well defined mapping of data objects (VHDL integer to C int)
- Simulation control (like $stop, ... in Verilog)
- Conditional compilation
- VCD for VHDL
- TEE functionality to STD.OUTPUT
- Verilog and C Foreign interfaces

VHDL-200x, Summary

SynthWorks

With the new features of VHDL-200X, VHDL will transition from being a simple HDL (hardware to description language) to a HDVL (hardware design and verification language). This will be a huge benefit for the users. Not only will we benefit from improved ease of use, but with the new features, there will no longer be any argument to learn an additional language such as Vera, SystemC, or specman e to help with verification.

IEEE 1164, Std Logic

SynthWorks

Goals: Enhance current std_logic_1164 package

A few items on the list are:

- Uncomment xnor operators
- Add shift operators for vector types
- Add logical reduction operators
- Add array/scalar logical operators
- Provide text I/O package for standard logic (similar to Synopsys' std_logic_textio)

Website: http://www.eda.org/vhdl-std-logic

See also DVCon 2003 paper, "Enhancements to VHDL's Packages" which is available at:

http://www.synthworks.com/papers

IEEE 1076.3, Numeric Std

SynthWorks

Goals: Enhance current numeric_std package.
Add a package for unsigned operations with std_logic_vector.
Add a package for floating point

A few items on the numeric_std list are:

- Logic reduction operators
- Array / scalar logic operators
- Array / scalar addition operators
- TO_X01, IS_X for unsigned and signed
- Unsigned arithmetic for std_logic_vector & std_ulogic_vector
- TextIO for numeric_std
- Floating point arithmetic

Website: http://www.eda.org/vhdlsynth
http://www.eda.org/fphdl -- floating point

IEEE 1076.6, RTL Synthesis

SynthWorks

Goals: Enhance synthesis coding styles to accept a wider set of synthesizable objects

A few items on the list are:

- Broader register coding styles
- Multiple clocked and multiple edged registers
- Support synthesis of registers in subprograms
- Support registers and latches in concurrent assignments

Website: http://www.eda.org/siwg

See also HDLCon 2002 paper, "Extensions to the VHDL RTL Synthesis Standard", which is available at:
http://www.synthworks.com/papers

Who Sponsors VHDL Standards?

SynthWorks

VHDL standards are IEEE standards.

IEEE Standards Association (IEEE-SA) coordinates balloting on all IEEE Standards.

The Design Automation Standards Committee is responsible for all design automation related standards.

VHDL development is coordinated by the VHDL Analysis and Standardization Group (VASG).

The VHDL-200x working group is part of VASG

Some of VHDL work is supported by Accellera

Websites:

IEEE:	http://www.ieee.org
DASC:	http://www.dasc.org
VASG:	http://www.eda.org/vasg
Accellera:	http://www.accellera.org

Participating in VHDL Standards

SynthWorks

Participation is open and all are encouraged to get involved

- It is your right and responsibility to participate

How can I participate?

- Read the reflector to catch up on the discussion.
- Join the reflector
- Discuss issues on reflector
- To have a vote on issues, you must be DASC member and participate in voting on a regular basis (per group rules)
- To ballot on a standard, you must be an IEEE SA member.

These are not static efforts. Your input can make a difference.

Vendor Support of Standards

SynthWorks

EDA vendor support of standards is not as simple as it may seem. For EDA vendors, supporting a standard is an investment. Hence, feature support is market driven. They don't support new features based on merit, they support them based on user requests.

As a result, if you see new features in a standard that you would like to use, make sure to request that your EDA vendor support the feature.

Author Biography

SynthWorks

Jim Lewis, Director of Training, SynthWorks Design Inc.

Jim Lewis, the founder of SynthWorks, has seventeen years of design, teaching, and problem solving experience. In addition to working as a Principal Trainer for SynthWorks, Mr. Lewis does ASIC and FPGA design, custom model development, and consulting. Mr. Lewis is an active member of IEEE Standards groups including, VHDL (IEEE 1076), RTL Synthesis (IEEE 1076.6), Std_Logic (IEEE 1164), and Numeric_Std (IEEE 1076.3). Mr. Lewis can be reached at jim@SynthWorks.com, (503) 590-4787, or http://www.SynthWorks.com

SynthWorks VHDL Training

SynthWorks

Comprehensive VHDL Introduction 4 Days
http://www.synthworks.com/comprehensive_vhdl_introduction.htm
A design and verification engineers introduction to VHDL syntax, RTL coding, and testbenches.
Our designer focus ensures that your engineers will be productive in a VHDL design environment.

VHDL Coding Styles for Synthesis 4 Days
http://www.synthworks.com/vhdl_rtl_synthesis.htm
Engineers learn RTL (hardware) coding styles that produce better, faster, and smaller logic.

VHDL Testbenches and Verification 3 days
http://www.synthworks.com/vhdl_testbench_verification.htm
Engineers learn how create a transaction-based verification environment based on bus functional models.

For additional courses see: http://www.synthworks.com