



IEEE



IEEE

VHDL-200x

VHDL-200x DAC 2003

VHDL-200x Steering Committee Members

Stephen Bailey, VASG Chair
John Willis, ISAC and Fast-Track
Jim Lewis, Modelling & Productivity
J. Bhasker, Testbench & Verification

John Ries, Performance
Rob Anderson, Assertions
Paul Menchini, Data Types & Abstraction
Joshua Johnson, Asynchronous Design

Agenda

- IEEE information
- What is VHDL 200x?
- What is the Schedule?
- How does VHPI fit in?
- What is the organization?
- Enhancements (by functional team)
- Contact information

Instructions for the WG Chair



(Not necessary to be shown)

- **At Each Meeting, the Working Group Chair shall:**
- **Show slides #1 and #2 of this presentation**
- **Advise the WG membership that:**
 - The IEEE's Patent Policy is consistent with the ANSI patent policy and is described in Clause 6 of the *IEEE SA Standards Board Bylaws*;
 - Early disclosure of patents which may be essential for the use of standards under development is encouraged;
 - Disclosures made of such patents may not be exhaustive of all patents that may be essential for the use of standards under development, and that neither the IEEE, the WG nor the WG Chairman ensure the accuracy or completeness of any disclosure or whether any disclosure is of a patent that in fact may be essential for the use of standards under development.
- **Instruct the WG Secretary to record in the minutes of the relevant WG meeting:**
 - that the foregoing advice was provided and the two slides were shown;
 - that an opportunity was provided for WG members to identify or disclose patents that the WG member believes may be essential for the use of that standard;
 - any responses that were given, specifically the patents and patent applications that were identified (if any) and by whom.

IEEE-SA Standards Board Bylaws on Patents in Standards

6. Patents

IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard. This assurance shall be provided without coercion and prior to approval of the standard (or reaffirmation when a patent becomes known after initial approval of the standard). This assurance shall be a letter that is in the form of either

- a) A general disclaimer to the effect that the patentee will not enforce any of its present or future patent(s) whose use would be required to implement the proposed IEEE standard against any person or entity using the patent(s) to comply with the standard or
- b) A statement that a license will be made available without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination

This assurance shall apply, at a minimum, from the date of the standard's approval to the date of the standard's withdrawal and is irrevocable during that period.

Approved by IEEE-SA Standards Board – December 2002

Inappropriate Topics for IEEE WG Meetings

Slide #2

- Don't discuss licensing terms or conditions
- Don't discuss product pricing, territorial restrictions or market share
- Don't discuss ongoing litigation or threatened litigation
- Don't be silent if inappropriate topics are discussed... do formally object.

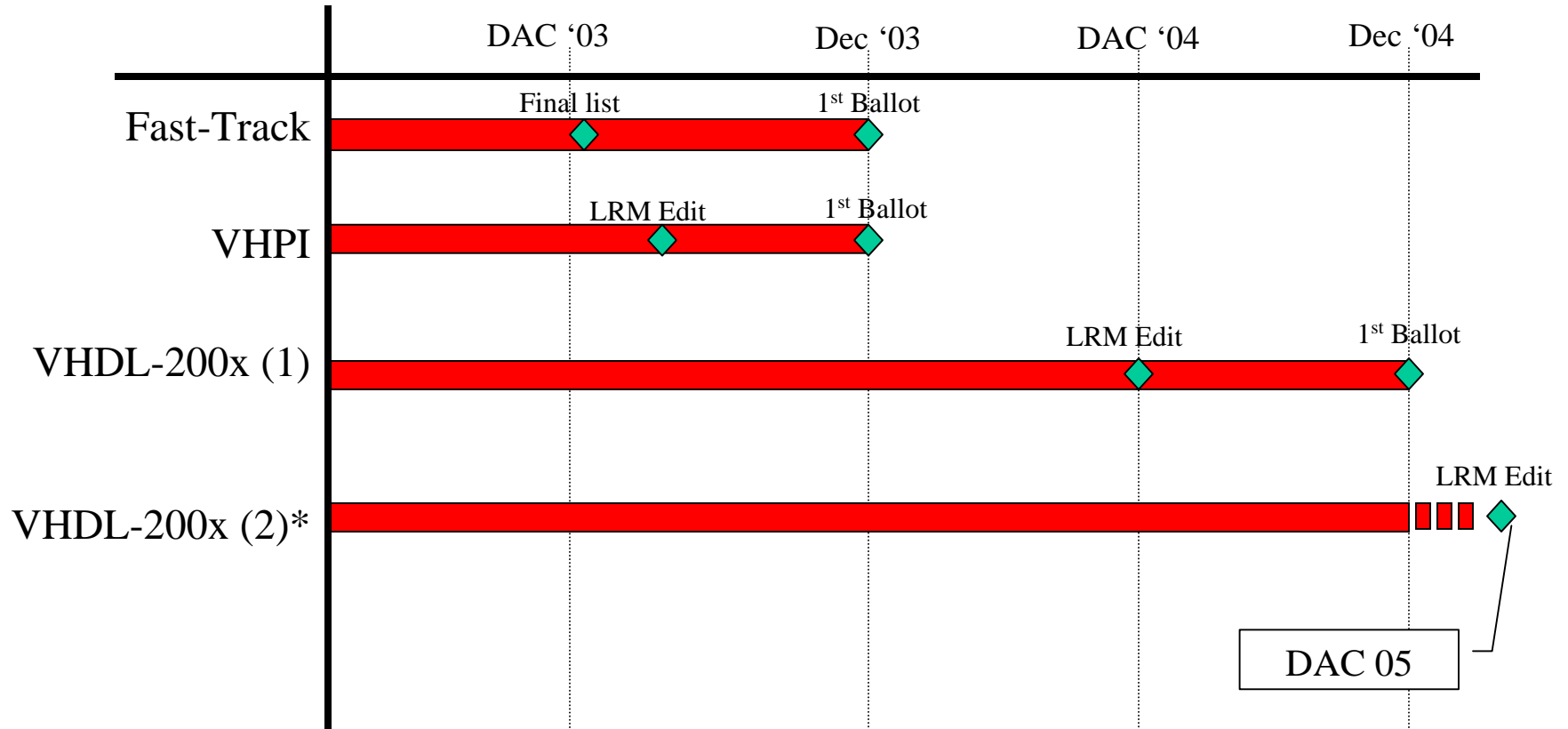
**If you have questions,
contact the IEEE Patent Committee Administrator
at patcom@ieee.org**

Approved by IEEE-SA Standards Board – December 2002

What is VHDL-200x?

- The next language revision (or two)
- Significant enhancements are planned
 - Not another “bug-fix release”
- Focus is on:
 - Performance & productivity
 - Testbench & verification
 - Assertion-based verification (ABV)
 - Modeling

Schedule



*If needed

Where are We Today

- Organized into teams
- Prioritized enhancement requests
- Some teams have published proposed language changes
- Others have started some analysis of requirements
- Others have not yet made significant progress
 - Post DAC it is important that they do begin making progress

How Does VHPI Fit In?

- VHPI LRM editing is funded
 - Thanks Accellera!
- FT and VHPI both ready to ballot in Dec?
 - Ballot 2 PARs individually
 - Ballot 2 PARs together
 - Fold into 1 PAR and ballot together
 - Recommendation: Fold into 1 PAR
 - Ensure FT & VHPI changes integrated

VHDL 200x Organization

- Fast-track
- Modeling & Productivity
- Performance
- Assertions
- Testbench & Verification
- Data Types & Abstraction
- Environment
- Asynchronous Modeling

Enhancements by Functional Team



Top Priorities

- Direct C & Verilog calls / std I/F
- Text IO
- to_string type capabilities
- Reading of out mode ports
- Performance
- Assertions
 - Apply Accellera assertions (PSL)
- Some modeling enhancements
- Environment (simulation control routines)
- Generate enhancements
- Testbench / verification
- Object-oriented enhancements

VHDL 200x Fast Track

John Willis and Jim Lewis

vhdl-200x-ft@vhdl.org

<http://www.vhdl.org/vhdl-200x/vhdl-200x-ft>

VHDL-200x-FT Charter

- Make critical updates to the language to support other standards groups (such as IEEE P1164, IEEE P1076.3, and Assertions).
- Select additional enhancements that:
 - Have low LRM impact
 - Non-controversial
 - Relatively high benefit



VHDL-200x-FT Requests

<u>Index</u>	<u>Issue</u>	<u>Status</u>
FT1	Allow explicit operators (ie: ">") to overload implicit operators	Analyzed by ISAC
FT2	Unary Reduction Operators (and, or, xor, ...). Also see 1164 & numeric_std.	Analyzed by 1164
FT3	Array/scalar logic operations	Analyzed by 1164
FT4	Min/max operations for types with defined "<" and ">" operations	Proposed
FT5	to_string, to_hstring, to_ostring	Proposed
FT6	'image (for all types)	Proposed

VHDL-200x-FT Requests

<u>Index</u>	<u>Issue</u>	<u>Status</u>
FT7*	Standard Signal Spy / XMR capability (package/procedural based)	Proposed
FT8	hwrite, owrite, dwrite, bwrite, hread, oread, dread, bread	Analyzed by 1164

* Status of FT7

Have donation from MTI.

Expecting donation from Cadence.

Are in early stages of discussion with Synopsys.

VHDL-200x-FT Status



IEEE

- Leverage off of work being done by the 1164 and 1076.3 working groups.
- We are actively looking for champions for proposals

VHDL 200x

Modeling and Productivity

Jim Lewis

vhdl-200x-mp@vhdl.org

<http://www.vhdl.org/vhdl-200x/vhdl-200x-mp>



VHDL-200x-MP Charter

- Improve designer productivity through enhancing conciseness, simplifying common occurrences of code, and improving capture of intent.
- Enhance VHDL to allow (easy) modeling of functionality which is currently difficult or impossible.

VHDL-200x-MP Requests



IEEE

General Proposals

- Bidirectional Connections (Switch, Jumper, Resistor)
- Regularized and minimized bracketing (end)
- Allow concurrent assignments (conditional and selected) in sequential code
- Create a one dimensional array aggregate. Permit it to be used on LHS of assignment
- *Permit expressions to be mapped to signal ports of entities and subprograms.
- More locally static expressions. Things like concatenation, indexing, and slicing of static objects/values
- Allow attribute declarations in code regions (not just decl)

* Candidate for Fast Track (Simplifies use of OVL)



VHDL-200x-MP Requests

Entity

- Allow ";" to terminate as well as separate an interface list
- Eliminate passive statement restriction on entities

Package

- Allow subprogram bodies in package declaration region



VHDL-200x-MP Requests

Types, Operators, and Overloading

- Give `bit_vector` an unsigned interpretation.
- Create `rising_edge` and `falling_edge` for type `bit`.
- Integrate 1164, 1076.2, 1076.3 into 1076
- Create comparison operators that return `std_ulogic`.
?EQ, NE, GT, LT, GE, LE?
- Boolean equivalence (of `sl` and `boolean`). if `sl` then
- Ability to apply register kind semantics to `std_logic`.
Retain last resolved value when all drivers are off.
- * Max function (also listed in fast track)
- Remove white space requirement in physical literals.
- Short alias name for `std_logic_vector`.
- Value folding of `std_ulogic` (2 state/4 state).

VHDL-200x-MP Requests

Generate

- Add else/elsif clause in if-generate
- Case generate.

Components and Instantiations

- Implicit generic/port map in component instance

Process

- Permit keyword "all" or alternately symbol "*" in sensitivity list to imply all signals read in the process are in the sensitivity list

Loop

- Longest static prefix issue with loops



VHDL-200x-MP Requests

Case Statements

- Case Statement expressions (index & choices)
- Don't cares in case statement targets and comparison operators
- Non-locally static expressions in case expressions.

If Statements

- Add Endif (like elsif). See also general stuff about regularizing syntax and end
- Orif, orels, etc for mutual exclusive branches in FSM

VHDL-200x-MP Status

- We are actively looking for proposals
- We are actively looking for champions for proposals

VHDL-200x Performance

Chairman: John Ries

johnr@model.com

E-mail: vhdl-200x-perf@eda.org

<http://www.eda.org/vhdl-200x/vhdl-200x-pref>

Scope and Responsibilities

- Language changes that increase the speed of tools.
- Mostly focused on simulator performance.

Performance Proposals

- Zero-delay ordering of signals
- Removal of simulation deltas
- Define 2 & 4 state semantics
- Atomic composite signals
- Expressions in sensitivity lists

Performance Proposals(2)

- Sensitivity to all signals read
- Light-weight signals
- Architecture level signal drivers
- Removal of deprecated constructs

VHDL-200x Assertions

Chairman: Rob Anderson

rob@reawebtech.com

E-mail: vhdl-200x-asr@eda.org

<http://www.eda.org/vhdl-200x/vhdl-200x-asr>

Scope and Responsibilities

- Enhance VHDL to provide support for Assertion-Based Verification
 - Primarily this means adding support for temporal expressions/sequences with assertions
 - Secondly also includes API for coverage data
 - Want to leverage work of PSL and SystemVerilog assertions

Current Status

- Currently analyzing and comparing the capabilities of PSL and SystemVerilog Assertions
- Next step: determine if anything is missing
 - Especially from a VHDL perspective
- Expect PSL or SVA adoption as initial draft
 - How much deviation or enhancement TBD

Assertions Schedule

- Finish PSL / SVA comparison in June
- Publish comparison
- Determine & publish any additions desired for VHDL by mid July
- Determine VHDL starting point by Sep
- Finish technical proposal by Mar '04

VHDL-200x Testbench & Verification

Chairman: J. Bhasker

jbhasker@esilicon.com

E-mail: vhdl-200x-tbv@eda.org

<http://www.eda.org/vhdl-200x/vhdl-200x-tbv>

1076-200x: TB V Group

- Testbench and verification group
- List of issues under consideration posted on our home page – For eg. sparse arrays, constrained random number generation, lists, events.
- A couple of proposals on the table
- WG is active - started teleconferences
- Associative arrays and fork-join features discussed
- <http://www.eda.org/vhdl-200x/vhdl-200x-tbv>
- Questions? Jbhasker@esilicon.com

Associative arrays

-- Type declarations:

```
type myaaT is associative (INTEGER) of BIT;  
type COLOR is {Red, Blue, Green, Yellow, Orange};  
type my2aaT is associative (COLOR, COLOR) of  
INTEGER;
```

-- Two associative arrays:

```
variable mem_aa: myaaT;  
signal matrix: my2aaT;
```

-- Implicit subprograms:

Delete, exists, size, first, last, next, prev

Fork & Join

```
<fj_label>: fork          -- Sequential statement
  <seq_blk1>: declare    -- Sequential block; allowed outside
                        -- of a fork-join context as well.
    <local_declarations>
  begin
    <sequential_statements>
  end declare <seq_blk1>;

  <seq_blk2>: declare
    <local_declarations>
  begin
    <sequential_statements>
  end declare <seq_blk2>;
join all <fj_label>;    -- none / first / condition_clause /
                        -- timeout_clause
```



VHDL 200x Data Types and Abstraction

Paul Menchini

vhdl-200x-dta@vhdl.org

<http://www.vhdl.org/vhdl-200x/vhdl-200x-dta>

VHDL-200x-dta Charter

- Enhance VHDL data types abstractions.
- Serve as a resource to other groups contemplating changes to the existing VHDL data types and abstraction mechanisms.

VHDL-200x-dta Responsibilities

- Collect suggestions for change
- Analyze suggestions for change
- Analyze proposals and language designs from other subgroups
- Develop language designs for changes and proposals
- Recommend changes to language in dta

VHDL-200x-dta Requests

- Object Oriented Features
- Variant Records (“free unions”)
- New standard data types
 - Boolean, integer, real vectors
- Associative arrays
- Sparse arrays
- Enhanced object initialization
- Built-in lists

VHDL-200x-dta Status

- We wish to leverage off of previous work (e.g., in OO and variant records)
- We are actively looking for proposals
- We are actively looking for champions for proposals



VHDL 200x Environment

Dennis Soderberg

vhdl-200x-env@vhdl.org

<http://www.vhdl.org/vhdl-200x/vhdl-200x-env>



VHDL-200x-Env Charter

- Standardize handling of non-VHDL constructs.
- Enhance platform interoperability

VHDL-200x-Env Requests

General Proposals

- Add simulation control environment
- Simulation control subprograms
- Environment
- External Interface

VHDL-200x-Env Status

- We are looking for more proposals

Additional VHDL Enhancement Standardization Projects



IEEE

VHPI standard Status

John Shields, Françoise Martinolle



VHPI Technical status

- Technical specification is done
 - It covers: post-analysis, elaboration and runtime VHPI access
- Final draft is available on web site
<http://www.vhdl.org/vhdlpli>
- Editorial work
 - Initial funding (20%) provided by Accellera
 - Paul Menchini has begun incorporating VHPI in the VHDL 1076 LRM
 - First funded phase supposed to end June 15 (on track)
 - Integrate VHPI significant control points into elaboration and simulation.
 - Define common terminology
 - Organize and outline the additional chapter(s) and appendices of the VHDL LRM

VHDL 1076 deliverables

- VHPI integrated in the VHDL LRM 1076
- A CD containing the formal XML representation of the VHPI static and dynamic information model (binary readable)
 - Today the IEEE allows provision of a companion CD

Working Group status

- The WG is:
 - closing any remaining open issue
 - interacting with and reviewing Paul's first draft
 - organizing in parallel a comprehensive review of the current VHPI specification

Road map

- Goal is to have a ballotable LRM by end of October and go to ballot by 2003 end.
- Started to:
 - incorporate critical sections affecting VHDL semantics first
 - identify experts to early review draft versions

VHPI industry implementations

- Many companies have already provided some level of VHPI implementation to customers
 - Vendors include: Cadence, Synopsys, FTL systems...
 - Tools already using VHPI: Debussy, Verisity, Vera...
- Key vendors projected to aim at compliance by DAC 2004

1076.6 VHDL Synthesizable Subset

Chair: J. Bhasker
jbhasker@esilicon.com

1076.6

- Standard for VHDL RTL Synthesis
- First standard out in 1999 (IEEE Std1076.6-1999)
- WG diligently worked on next rev for past couple of years
- Next revision draft is ready for ballot
- Revision PAR just approved by IEEE
- Ballot and standardization expected by EOY
- <http://www.eda.org/siwig>, siwig@eda.org



VHDL-200x

P1164 Status

Peter Ashenden
Ashenden Designs

P1164 Chair

peter@ashenden.com.au

vhdl-std-logic@eda.org

www.eda.org/vhdl-std-logic

Standard Revision

- Changes to enhance usability
- Adding functions and operators
- Adding textio support
- No major conceptual changes
- To be balloted by end-2003
- www.eda.org/vhdl-std-logic

Accepted Changes To Date

- Uncomment "xnor" operators
- Shift operators for vector types
- Vector/scalar logic operators
 - scalar is replicated to vector's length
- Logical reduction functions
- Match functions

Rejected/Pending Changes

- Rejected
 - Subtype of vector results
 - Capacitive strength (deferred)
 - 'image (refer to VASG)
- Pending
 - `std_logic_textio` package (donated by Synopsys)
 - `to_string` functions

Articulation with VASG

- Vector/scalar operators for `bit_vector`
- Logical reduction functions for `bit_vector`
 - or unary logical reduction operators
- Generalize 'image to composite types
- Radix-formatted textio
- `To_string` functions