IEEE

VHDL-200x FDL 2003
23 Sep 03

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
Peter Ashenden, 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.

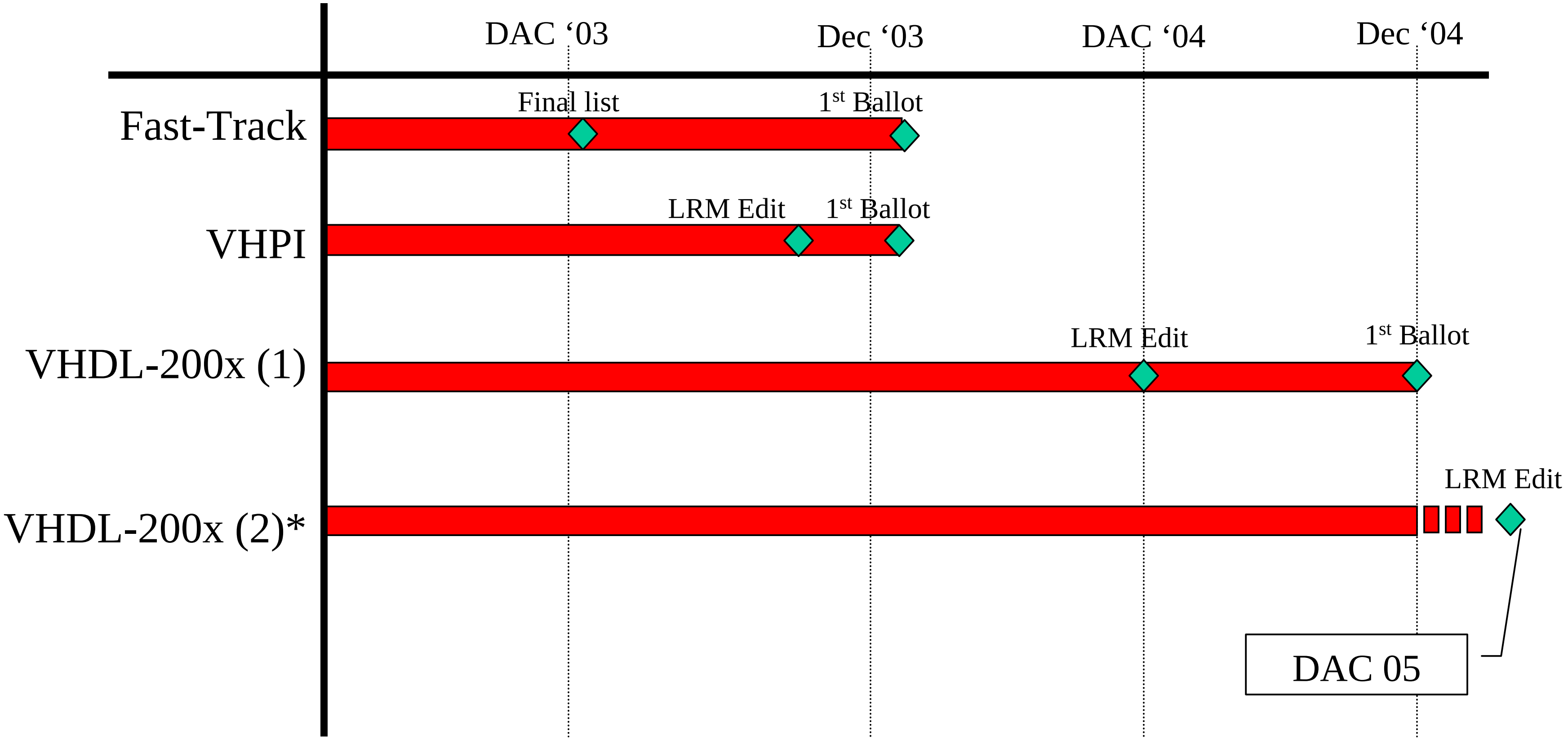
If you have questions,
contact the IEEE Patent Committee Administrator
at patcom@ieee.org

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

How Does VHPI Fit In?

- VHPI LRM editing is funded
  - Thanks Accellera!
  - Paul Menchini dropped out as editor
    - Francoise Martinolle and Peter Ashenden are completing
- 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).

VHDL-200x-FT Requests

Index	Issue	Status
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
FT3	Array/scalar logic operations	Analyzed
FT4	Min/max operations for types with defined "<" and ">" operations	Proposed
FT5	to_string, to_hstring, to_ostring	Proposed
FT6	'image (for all types)	Analyzed, Rejected

VHDL-200x-FT Requests

Index	Issue	Status
FT7*	Standard Signal Spy / XMR capability (package/procedural based)	Proposed
FT8	hwrite, owrite, dwrite, bwrite, hread, oread, dread, bread	Analyzed by 1164
FT9	Sized Bit String Literals	Analyzed
FT10	Sequential Conditional and Selected Signal Assignment	Proposed

* Status of FT7

Have donation from MTI and Cadence.

Synopsys has not made a donation to date and the donation deadline has passed.

FT1: Explicit Overrides implicit

Allow explicit operators (ie: ">") to overload implicit operators

Purpose: Allow 1076.3 to create a numeric package for overloading numeric operators including comparison operators (">", …)

FT2: Unary Reduction Operators

Unary Reduction Operators (and, or, xor, ...).

Purpose: Facilitates reduction operations (such as a parity).

Simplifies parity calculation like Parity1 below to Parity2.

```
Signal Parity1, Parity2, ParityEnable : bit ;
Signal Data : bit_vector(7 downto 0) ;

Parity1 <=
   (Data(7) xor Data(6) xor Data(5) xor Data(4) xor
    Data(3) xor Data(2) xor Data(1) xor Data(0))
   and ParityEnable ;



Parity2 <= xor Data and ParityEnable ;
```

FT3: Array/Scalar Logic Operations

Purpose: Facilitates controlling/gating of array objects.

Simplifies code like DataOut1 to DataOut2.

```
signal ASel, BSel, CSel, DSel : std_logic ;
signal A, B, C, D : bit_vector(7 downto 0) ;
signal DataOut1, DataOut2 : bit_vector(7 downto 0) ;

GenLoop : for I in DataOut1’Range loop
begin
  DataOut1(I) <=
    (A(I) and ASel) or (B(I) and BSel) or
    (C(I) and CSel) or (D(I) and DSel) ;
end generate ;

DataOut2 <=
   (A and ASel) or (B and BSel) or
   (C and CSel) or (D and DSel) ;
```

FT5: to_string, to_hstring

Purpose: Enhance VHDL's string handling capability.

Permits useful messages to be printed with report.

```
assert (ExpectedVal = ReadVal)
  report "In ... Expected Value /= Read Value.  Expected = " &
    to_string(Expected) & "    Read = " & to_string(ReadVal))
  severity error ;
```

Also works well with VHDL-93 write(<file_handle>, <string>)

```
write(Output, "%%%ERROR data value miscompare in CpuModel." &
       LF & "  Actual data value = " & to_hstring(Data) &
       LF & "  Expected data value = " & to_hstring(ExpData) &
       LF & "  at time:  " to_string(now, right, 12) ) ;
```

FT9: Sized Bit String Literals

Purpose: Enhance VHDL's literal capability

- Prefixed with a number to indicate number of bits
- Prefixed with S if the number is signed
- Expand and replicate metavalues

Literal	Equivalent String	Comment
X"17"	"000010111"	Implicitly sized to 8 bits
5X"17"	"10111"	Explicitly sized to 5 bits
8X"F"	"00001111"	Default, 0 fill extra bits
8SX"F"	"11111111"	Signed, replicate left bit
7X"8F"	"0001111"	Warning due to truncation?
7SX"8F"	"1001111"	Sign bit handled correctly
X"ZZ"	"ZZZZZZZZ"	Expanding Z
O"01-"	"000001---"	Expanding -

FT10: Sequential Conditional & Select Signal Assignment

Purpose: Simplify coding. Simplifies State1Proc to State2Proc

```
State1Proc : process (StateReg, FP)
begin
  case StateReg is
    when IDLE =>
      if (FP = '1') then
        NextState  <= FLASH ;
      else
        NextState  <= IDLE ;
      end if ;
```

```
State2Proc : process (StateReg, FP)
begin
  case StateReg is
    when IDLE =>
      NextState <= FLASH when (FP = '1') else IDLE ;
```

VHDL-200x-FT Status

- 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

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

* Candidate for Fast Track (Simplifies use of OVL)

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
- 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-twiki.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

Status

- 18 people subscribed to vhdl-200x-perf
- Eight initial draft change proposals
- Looking for additional requests
- Looking for people to analyze requests and draft change proposals

Current Proposals

- PERF01: Removal of simulation deltas
- PERF02: Expressions in sensitivity lists
- PERF03: Define 2 & 4 state semantics
- PERF04: Light-weight signals
- PERF05: Atomic composite signals
- PERF06: Removal of deprecated constructs
- PERF07: Zero-delay ordering of signals
- PEFF08: Sensitivity to all signals read (MP overlap)

VHDL-200x Assertions

Chairman: Rob Anderson
rob@reawebtech.com
E-mail: vhdl-200x-asr@eda.org
http://www.eda-twiki.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
  - – Secondarily also includes API for coverage data
  - – Want to leverage work of PSL and SystemVerilog assertions

Current Status

- Meeting at Mentor Graphics early July 2003
- Decided to incorporate PSL as VHDL’s ABV language foundation
- Began considering issues with integrating PSL with VHDL
- Conducted advisory vote with broader WG on using PSL as ABV basis.
  - Vote passed overwhelmingly

Assertions To Do

- Accellera is “merging” SystemVerilog Assertions and PSL
  - We will use the result
- Meanwhile:
  - Language changes to facilitate ABV with PSL
  - Define integration semantics
  - Define VHPI extensions for ABV/PSL capabilities

Schedule

- We hoped it could be included with VHPI and Fast-Track
  - Accellera PSL/SVA merge slowing things
- If reasonable, we could ballot addition separately when it is ready
  - Depends on user demand
  - Status of VHPI/Fast-Track and subsequent activities

VHDL-200x Testbench & Verification

Chairman: J. Bhasker
jbhasker@esilicon.com
E-mail: vhdl-200x-tbv@eda.org
http://www.eda-twiki.org/vhdl-200x/vhdl-200x-tbv

1076-200x: TBV 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 : Associative arrays, fork-join, queues/fifos, event objects, lists
- Proposals on other topics are welcome
- WG is active - started teleconferences
- http://www.eda.org/vhdl-200x/vhdl-200x-tbv
- Questions? Jbhasker@esilicon.com

Issues being considered

- Associative arrays
- Fork-join
- Queues / FIFOs
- Lists
- Synchronization and handshaking (event objects)
- Request and wait for action
- Expected value detectors
- Access to coverage data for reactive TB
- Sparse arrays
- Random value generation with optional and dynamic weighting
- Random object initialization
- Random 2-state value resolution
- Loading and dumping memories

Associative arrays

-- Type declarations:

```
type BitAssocArrayT is associative (INTEGER) of BIT;
type COLOR is (RED, BLUE, GREEN, YELLOW, ORANGE);
type ColorAssocArrayT is associative (COLOR, COLOR) of INTEGER;
```

-- Two associative arrays:

```
variable MemoryA1: BitAssocArrayT;
signal ScoreBoard: ColorAssocArrayT;
```

-- Implicit subprograms:

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

Fork & Join

```
[ fj_label :]  fork
   [ [sblk_labela:] declare – A sequential block
      { declarations } ]
   begin
      { sequential statements }
   end declare [ sblk_label_a ] ;
   [ [sblk_label_b:] declare – seq block is also a seq statement
      { declarations } ]
   begin
      { sequential statements }
   end declare [ sblk_label_b ];
   . . .
join [ all | none | first | [ condition_clause ]
     [ timeout_clause ] ]  [ fj_label ];
```

Lists

- Type declarations

```
type IntListT is list (<>) of INTEGER; -- unconstrained list
type BvListT is list (0 to 5) of BIT_VECTOR(0 to 3);
                                         -- constrained list
type StringListT is list (3 downto 0) of STRING (0 to 5);
```

- List objects:

```
variable usb_fan: IntListT; -- List is empty by default
signal usb_data: BvListT := (“001”, “000”, “000”);
-- List has three elements indexed from 0 to 2.
```

- Predefined attributes: ‘DELETE, ‘INSERT, ‘LENGTH, ‘SORT, ‘UNIQUE, ‘REVERSE, ‘EXISTS, and ‘INDEX.

Alternatives

- Suggestion to use SUAVE style dynamic processes
  - In addition to fork/join?
  - In place of fork/join?
  - Need some examples to better assess needs
- Type genericity proposal from SUAVE and FTL Systems work
  - General solution
  - Used to provide lists, FIFOs, associative array, etc. implementation?
  - Working with DTA team

VHDL 200x Data Types and Abstraction

Peter Ashenden

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 (work with TBV)
- Sparse arrays
- Enhanced object initialization
- Built-in lists

VHDL-200x-dta Status

- Preparing proposals for:
  - Type genericity to support TBV requirements for container, communication and abstract-data types
  - Object-oriented data modeling
- Monitoring proposals from other teams:
  - Ensure clean integration with existing language
  - Ensure clean integration between 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

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
  - Funding provided by Accellera
  - Paul Menchini finished first phase editing
    - 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
  - Change of career = Paul no longer doing editing work
    - Francoise, John and Peter are coordinating to complete editing

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)
  - IEEE allows provision of a companion CD

Working Group status

- The WG is:
  - Determining where Paul left off and how much editing work remains
  - Working with Peter as new tech editor
  - Reviews of Paul’s previous work were completed
    - Additional reviews will continue as editing work continues

Road map

- Goal was to have a ballotable LRM by end of October and go to ballot by 2003 end.
  - – Change of editor may delay by a few months

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 Related Standards

- Please attend DASC meeting on Thursday @ 14:00
  - 1164
  - 1076.6
  - Others
- VHDL-200x will coordinate and facilitate any enhancements requested by 1076.x & 1164 WGs.