ESA Coding Standards

Prof. K. J. Hintz

Department of Electrical
and
Computer Engineering
George Mason University

ESA Modeling Guidelines

- European Space Research and Technology Center
- September 1994
- Based on VHDL-93
- Applicability
  - Documentation
  - Simulation
  - Not logic synthesis

Basic Language

- English
  - Documentation
  - Identifiers
  - Comments
  - Messages
  - File names
  - *etc.*

Readability Standards

- Consistent Writing Style
- Consistent Naming Conventions
- Reserved Words
  - Uniform casing (all lc or uc)
- Identifiers
  - Mixed case
  - Recommended < 15 chars, max of 28

Readability Standards

- Concise
- No Unused Code
- No More Than One Statement per Line
- Maximum of 80 chars/line
- Consistent Indentation
  - No **`Tab`**
  - Fixed number of spaces (*e.g.*, 3)

Readability Standards

- Group Related Constructs
  - Separate groups using
    - » blank lines
    - » dashed lines
- Vertically Align
  - Comments
  - Identifiers

Naming Conventions

- Documented in Each File
  - Meaningful
  - Non-cryptic
  - No Extended Identifiers
- Same as Actual Hardware
- Active Low Clearly Indicated in Name, *e.g.*,
  - Reset_N
  - Enable_Bar

Naming Conventions

- Indicate Purpose, Not Type
  - **`PC_Counter`**

  not
  - **`Eight_Bit_Reg`**

Comments

- Explain Function of Module to Other Designers
- Explanatory, Not Just Restatement of Code
- Locate Close to Code Described
  - Put near executable code, not just in header

Required File Header

- Name of Design Unit(s) in File
- File Name
- Purpose of Code
- Description of Hardware Modeled
- Limitations of Model
- Known Errors
- Intended Design Library of Compiled Code

Required File Header

- List of Analysis Dependencies
- Author and address
- Environment
  - Simulator
  - Version
  - Platform
- Change List

Subprogram Description

- Description of Function Immediately Precedes Declaration
- Include
  - Assumptions
  - Limitations
  - Parameters
  - Results

Port and Generic Clauses

- One Signal Declaration per Line
- Immediately Follow Each With Descriptive Comment
  - Do not group comments

Types

- MSB = = Leftmost Bit
- Allow for Type Change w/o Changing Behavior of Simulation by avoiding
  - Default initialization
  - Relying on the number of type values in a type declaration
  - Dependencies on the order in a type declaration

Literals

- Real
  - Only use decimal format
- Radix
  - 2, 8, 10, 16 Only
  - No exponent
  - No underscores in radix 10
- Hexadecimal
  - Uppercase only

Files

- Only One Allowed File Type
  - **`Std.Textio.Txt`**
- When Used as Input to VHDL, Fully Document Format in VHDL Code
- Line Length
  - 80 recommended
  - 255 max

Signals

- If Possible
  - Same signal name throughout all levels of hierarchy
- If Not
  - Derive signal names from base name
- Ordering of Vectors Matches Device Data Sheet

Ports

- Logical Ordering
  - Order by mode recommended
    - » in, bidirectional, out
  - Order by function allowed
    - » order by mode within each function
- Port Maps
  - Only named association

Assertion Statements

- Avoid Insignificant Events
- Should Provide
  - Clear description of reason for assertion
  - Hierarchical path to instance or package
  - Identify relevant signals

Assertions Severity Levels

- Failure
  - Errors in the model itself
- Error
  - Timing violations affecting state of model
  - Invalid data affecting state of model

Assertions Severity Levels

- Warning
  - Timing violations not affecting state of model
  - Invalid data not affecting state of model
- Note
  - Other Essential information

Declarations

- Descriptive Label for All Processes
- Sensitivity List
  - Use if only one wait statement
- Use Matching Identifier at End
- Avoid Using a Procedure Which Modifies Parameters Not Passed to It

Declarations

- Top-level Entity Should Have Same Name as Device Modeled

Configurations

- No Configuration Specifications within Architectures
- Explicit Configurations Should be Used in Testbenches

Packages

- Use IEEE Approved Packages if Available
- Package Modules by Functionality, *e.g.*,
  - Timing parameters
  - Timing subprograms
- Same Order of Declarations in Both Package Declaration and Package Body
- Non-IEEE Packages Must be Verified

Design Libraries

- Separate Design Library for Each Model
- Named After Device with **`_Lib`** appended
- Contains All Design Units Used by the Model except
  - IEEE std
  - Common packages
- Testbenches in Their Own Library
  - Contains all components except DUT

Constructs to Avoid

- No Model Intercommunication Through Files
- All Resolution Functions Must Be
  - Commutative [a*b = b*a]
  - Associate [a*(b*c) = (a*b)*c)]
- Shared Variables
  - If used, must be proved to be deterministic

Constructs to Avoid

- Floating Point Values
  - Conversions
  - Comparisons
- Operating System Dependent Features
- Absolute Path Names for Files
- Absolute Addressing with Implicitly Declared Index, *e.g.*, string
  - Use predefined attributes

Constructs to Avoid

- Renaming Subprograms by Encapsulating Them in Subprograms with Different Names
- Hiding Signals, Variables, Constants, Subprograms, or Components by Declaring Another Object with the Same Name
  - Overloading OK

Constructs to Avoid

- Redefine Anything Which is Predefined
- References to the **`Work`** Library

Verification

- Performed by Someone Not Involved in the Creation of the Module
- Compare with Results of Other Models if Available
- Performed Completely in VHDL Using No Simulator Specific Features
- Execute Every Line of the Model

Verification

- Exercise All Assertion and Report Statements
- Verification of Subcomponents Does Not Obviate the Need to Test Them in Complete Model
- Verify All Boundaries and Singularities

Deliverables

- Electronic Format
  - Separate file for each design unit
  - File name same as design unit
  - ASCII
  - QIC-150 tape cartridges
  - Restore on Sun Sparc using **`tar`**
  - Fiducial information and how to restore written on tape label

Specific Model Requirements

- Component Level Simulation
- Board Level Simulation
- System Level Simulation

Component Simulation Model

- Purpose
  - Verify by simulation before manufacture
- Must have Correct Timing Characteristics
- Not Necessarily Synthesizable
- Either Gate Level or RTL
- Suggested to Not Mix Structural and Behavioral Descriptions in Same Architecture

Allowed Types

- Prefer
  - **`Bit`**
  - **`Bit_Vector`**
  - **`Boolean`**
  - **`Integer`**
  - **`IEEE.Std_Logic_1164 package`**

Component Model Interface

- Digital Signals for Model Interface
  - Preferred
    - **`Std_Logic`**
    - **`Std_Logic_Vector`**
  - Allowed (no others)
    - **`Bit`**
    - **`Bit_Vector`**
- Analog: **Real** type suggested

Component Model Signals

- No Global Signals
- All Signals in Port Clause, Even for Unmodeled Functions
- Power Pins Not Required
- No User-defined Types in Port Clause

Board Level Simulation Models

- Purpose
  - Verification of a Board Using One or More Components
- RTL or Higher Level
  - Gate-level netlist not acceptable
- Full functionality
  - Manufacturing test need not be implemented

Board Level Simulation Models

- Interface Signals
  - Correct Digital Waveforms
  - Correct Timing
- Coded for Simulation Speed
  - Minimum number of processes, signals, and signal assignment statements
  - Maximum use of variables
  - Use more abstract types

Board Level Simulation Models

- Avoid Reading Files
- Accompanying User's Manual with
  - Block diagram
  - Interconnecting signals
  - Sufficient details to allow another VHDL designer to develop models to perform board level simulations without needing original VHDL code

Board Level Model Interface

- Digital Signals for Model Interface
  - Preferred
    - **`Std_Logic`**
    - **`Std_Logic_Vector`**
- Analog: **Real** type suggested
- Pull-up/Pull-down Must be Modeled
- No Global Signals

Board Level Simulation Models

- Naming
  - Top-level entity name should be **`BoardLevel`**
- All Signals in Port Clause, Even for Unmodeled Functions
- Power Pins Not Required
- Unknown Values Only Need Be Reported Using Assertion Statements

Board Level Model Timing

- All Inputs Must be Checked w.r.t.,
  - Period
  - Pulse width
  - Setup Time
  - Hold Time
- Only Timing Violations Affecting Simulation Behavior Should be Asserted

Board Level Model Timing

- Outputs
  - All outputs assigned delays
  - Timing correctly modeled w.r.t. input signals and internal delays

Board Level Timing Parameters

- Timing Parameters Selectable Among
  - Worst case: lowest voltage, highest temperature, slowest execution speed
  - Typical case: nominal voltage, temperature, execution speed
  - Best case: highest voltage, lowest temperature, fastest execution speed
- Specified in Separate Package

Board Level Timing Parameters

- Parameters Expressed in Integer Number of ns
- Model Must Allow Timing Disabling Using Generics
  - **`TimingChecksOn`** : Boolean := False
- Timing Paramers Should Use Names Compliant with Vital Model Development Specification

Board Level Verification

- Automatically Performed Using Testbench
- Verify Using All Values of **`Std_Logic`** Type

System Level Simulation

- Purpose
  - Provide the functionality of a board, subsystem, algorithm, or protocol with a simulation speed allowing trade-offs to be performed
- No Similarity to Hardware is Required
- Coded for Efficient Simulation
- Interface
  - Use most suitable types

System Level Simulation

- Verification
  - Automated using testbench

End of Lecture