
A VHDL Modeling Guide

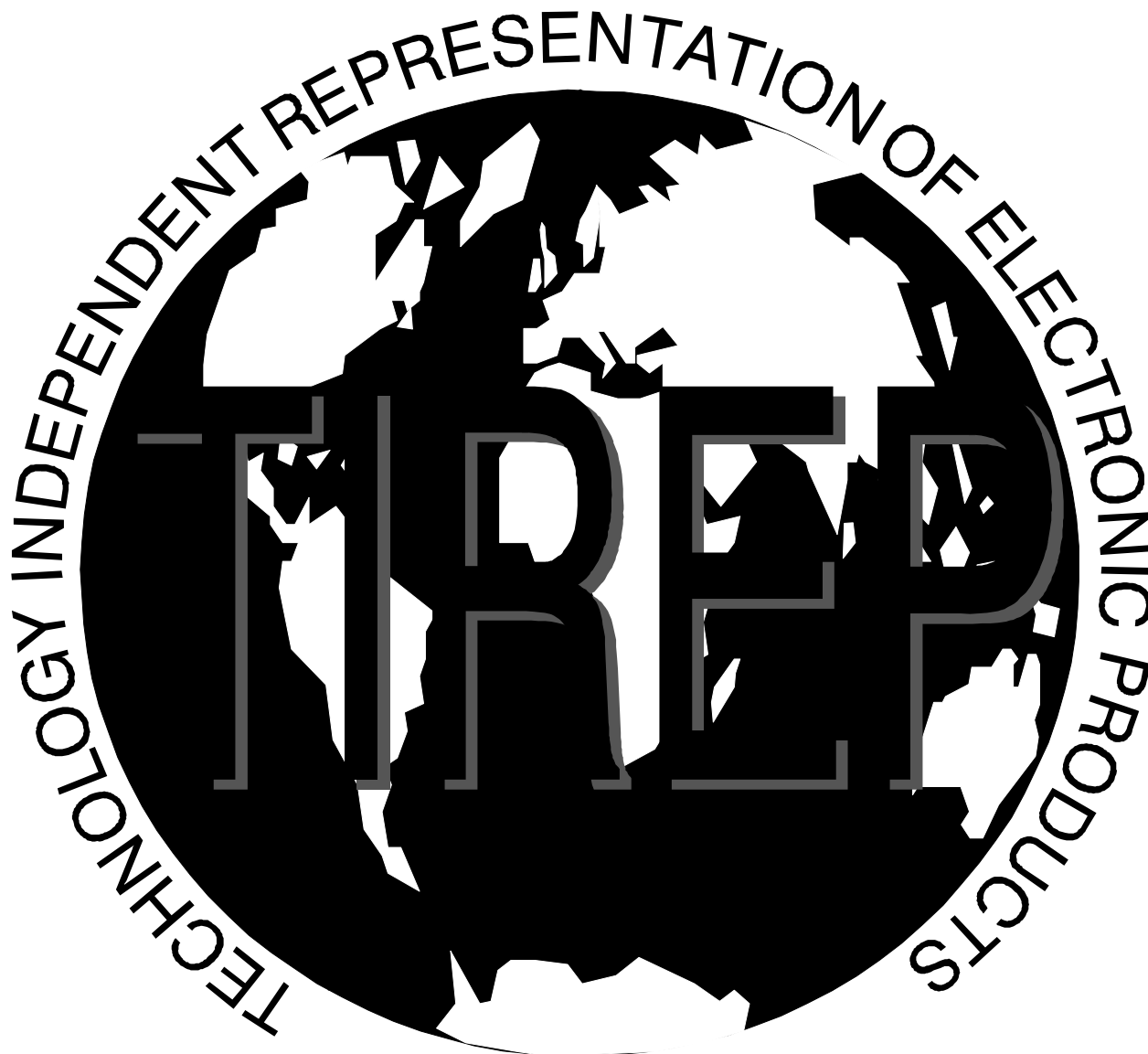
TP-804

Version 1.0

26 September 1994

A Product of the Standard Hardware And Reliability Program (SHARP),
Technology Independent Representation of Electronic Products (TIREP)
Project

A cooperative effort of the Naval Research Laboratory (NRL), the Naval
Surface Warfare Center (NSWC) and the Naval Air Warfare Center,
Aircraft Division (NAWC-AD).



A VHDL Modeling Guide

EXECUTIVESUMMARY

This document was developed under the Standard Hardware and Reliability Program (SHARP) Technology Independent Representation of Electronic Products (TIREP) project. It is intended for use by VHSIC Hardware Description Language (VHDL) design engineers and is offered as guidance for the development of VHDL models which are compliant with the VHDL Data Item Description (DID DI-EGDS-80811) and which can be provided to manufacturing engineering personnel for the development of production data and the subsequent production of hardware. Most VHDL modeling performed to date has been concentrated at either the component level or at the conceptual system level. The assembly and sub-assembly levels have been largely disregarded. Under the SHARP TIREP project, an attempt has been made to help close this gap. The TIREP models are based upon low complexity Standard Electronic Modules (SEM) of the format A configuration. Although these modules are quite simple, it is felt that the lessons learned offer guidance which can readily be applied to a wide range of assembly types and complexities.

Sponsor: **Mr. Dave Fisher**
 SHARP Program Manager

SHARP Project Leaders

NSWC: **Mr. Joe Toy**
NAWC: **Mr. Tony Hawkins**

TIREP Task Leaders

NSWC: **Mr. John Miles**
NAWC: **Mr. Louie Kitcoff**

TIREP Team Members

NSWC: **Mr. Gary Hout**
 Mr. Darin York
 Mr. Ed Woods
NRL: **Ms. Lisa Ceder**
NAWC: **Mr. Dave Broadhead**
 Mr. Chuck Rogers - Author

Comments, questions and concerns regarding this document may be addressed to:

Mr. Charles K. Rogers
NAWC-ADI
Code 306, MS-42
6000 E. 21st St.
Indianapolis, IN 46219-2189
Phone: 317-353-3579
Fax: 317-353-3690
E-Mail: rogersc1@po2.nawc-ad-indy.navy.mil



This page intentionally left blank

Table of Contents

Chapter 1	1-1
1.0 A VHDL Modeling Guide	1-1
1.1 Background	1-1
1.1.1 SHARP TIREP	1-1
1.1.2 Approach	1-1
1.1.3 Model Development	1-1
1.1.4 Project Objectives	1-2
1.2 Reference Documents	1-2
1.2.1 Baseline References	1-2
1.2.2 Supporting Documents	1-2
1.3 Definitions	1-3
1.3.1 Analysis	1-3
1.3.2 Application Specific Integrated Circuit (ASIC)	1-3
1.3.3 Architecture Body	1-3
1.3.4 Behavioral Architecture	1-3
1.3.5 Complete Model	1-3
1.3.6 Compliant Model	1-3
1.3.7 Construct	1-3
1.3.8 Data Flow Architecture	1-3
1.3.9 Design Configuration	1-3
1.3.10 Design Library	1-3
1.3.11 Design Unit	1-3
1.3.12 Electronic Data Sheet (EDS)	1-4
1.3.13 Entity Declaration	1-4
1.3.14 Leaf Level Module	1-4
1.3.15 Library Unit	1-4
1.3.16 Operating Point	1-4
1.3.17 Package Body	1-4
1.3.18 Package Declaration	1-4
1.3.19 Propagation Delay	1-4
1.3.20 Register Transfer Level (RTL) Architecture	1-4
1.3.21 Structural Architecture	1-4
1.3.22 Synthesis	1-4
1.3.23 Synthesizable	1-4
1.3.24 Testbench	1-4
1.3.25 Transition Time	1-4
1.3.26 Very High Speed Integrated Circuit (VHSIC)	1-5
1.3.27 VHDL Module	1-5
1.3.28 VHSIC Hardware Description Language (VHDL)	1-5
1.3.29 Waveform and Vector Exchange Specification (WAVES)	1-5
1.3.30 WAVES Header File	1-5
1.4 Acronyms	1-5
1.5 Chapter Overview	1-5
1.5.1 Chapter 1	1-5
1.5.2 Chapter 2	1-5
1.5.3 Chapter 3	1-5
1.5.4 Chapter 4	1-5
1.5.5 Chapter 5	1-6
1.5.6 Chapter 6	1-6
1.5.7 Chapter 7	1-6
Chapter 2	2-1
2.0 Modeling Conventions	2-1

Table of Contents - continued

2.1 Introduction	2-1
2.1.1 Overview	2-1
2.1.2 Selected Modules	2-1
2.2 File Conventions	2-1
2.2.1 VHDL File Naming Conventions	2-1
2.2.2 Supplemental Recommendations.	2-2
2.3 A VHDL File Header	2-2
2.3.1 An Example VHDL Header	2-2
2.4 Coding Conventions	2-5
2.4.1 Design Unit Naming Conventions	2-5
2.4.2 Other Naming Conventions	2-6
2.4.3 Code Readability	2-7
2.4.4 Code Walk-Throughs	2-7
2.5 The TIREP Model	2-8
2.5.1 The TIREP Model Structure	2-8
2.5.2 TIREP Component Model Files	2-10
2.5.3 TIREP Assembly Model Files	2-10
2.6 Tricks and Gimmicks	2-11
2.6.1 General	2-11
2.6.2 A SN5476 J-K Flip-Flop Example	2-11
2.7 Reference Information	2-14
2.7.1 Predefined VHDL Data Types	2-14
2.7.2 Predefined VHDL Operators	2-14
2.7.3 Reserved Words	2-15
Chapter 3	3-1
3.0 A Modeling Template	3-1
3.1 Introduction	3-1
3.1.1 Overview	3-1
3.1.2 Scope	3-1
3.2 A Design Example	3-1
3.2.1 The CD4029B Counter	3-1
3.2.2 Fixed Files	3-1
3.3 Files with Context Clause Modifications	3-2
3.3.1 The EIA Design Specification Package	3-2
3.3.2 The WAVES Objects File	3-3
3.4 Design Files	3-3
3.4.1 The Core Component	3-3
3.4.2 The Design Specification (DS) File	3-3
3.4.3 The EDS Model	3-25
3.5 WAVES Dataset Files and Testbench	3-32
3.5.1 The WAVES Header File	3-32
3.5.2 The Wav_Frame Package	3-32
3.5.3 The Device Under Test Package	3-34
3.5.4 The Waveform Generator Package	3-34
3.5.5 The Vector File	3-38
3.5.6 The Testbench	3-38
3.6 Order of Analysis	3-51
3.6.1 The EDS Model	3-51
3.6.2 The WAVES Dataset	3-51
3.6.3 The Testbench	3-51
Chapter 4	4-1
4.0 Modeling with EIA-567	4-1

Table of Contents - continued

4.1	Introduction	4-1
4.1.1	Background	4-1
4.1.2	The Electronic Data Sheet (EDS)	4-1
4.1.3	The EIA-567 Approach	4-1
4.1.4	Interdependencies	4-1
4.1.5	Scope	4-1
4.2	Len Finegold's EIA-567 Approach	4-1
4.2.1	Generic Drivers and Checkers	4-1
4.2.2	The EIA-567 Packages	4-1
4.2.3	The Electronic Data Sheet	4-2
4.2.4	Pin Characteristics	4-2
4.3	A Core Model	4-2
4.3.1	A 12 Bit Population Counter	4-2
4.3.2	STD_LOGIC_1164	4-2
4.3.3	The Uninitialized State	4-2
4.3.4	Power	4-5
4.3.5	Busses or Vectors	4-5
4.4	The EIA-567 Electrical View Package	4-5
4.4.1	Subprogram and Data Type Declarations	4-5
4.4.2	Interdependencies	4-10
4.5	The FBG Electrical View Package	4-10
4.5.1	Data Type Declarations	4-10
4.5.2	Requirements	4-14
4.5.3	Limitations	4-14
4.5.4	Interdependencies	4-14
4.6	The EIA-567 Timing View Package	4-14
4.6.1	Functions and Data Type Declarations	4-14
4.6.2	Interdependencies	4-22
4.7	The FBG Timing View Package	4-22
4.7.1	Data Type Declarations	4-22
4.7.2	Requirements	4-23
4.7.3	Interdependencies	4-23
4.8	The EIA-567 Physical View Package	4-23
4.8.1	Data Type Declarations	4-23
4.8.2	Interdependencies	4-28
4.8.3	Package Body	4-28
4.9	The FBG Physical View Package	4-28
4.9.1	Data Type Declarations	4-28
4.9.2	Interdependencies	4-35
4.10	The EIA-567 Toolbox Package	4-36
4.10.1	Subprogram Declarations	4-36
4.10.2	Interdependencies	4-36
4.11	The Input Driver	4-36
4.11.1	Operation	4-36
4.11.2	Interdependencies	4-41
4.12	The Output Driver	4-41
4.12.1	Operation	4-41
4.12.2	Interdependencies	4-41
4.13	The Bidirectional Driver	4-41
4.13.1	Operation	4-41
4.13.2	Interdependencies	4-41
4.14	Simulation Control Variables	4-43

Table of Contents - continued

4.14.1 "X_Generation"	4-43
4.14.2 "M_Generation"	4-43
4.15 The Asynchronous Checker	4-43
4.15.1 Operation	4-43
4.15.2 Interdependencies	4-45
4.16 The Synchronous Checker	4-45
4.16.1 Operation	4-45
4.16.2 Interdependencies	4-47
4.17 The Electronic Data Sheet Model	4-50
4.17.1 Operation	4-50
4.17.2 Interdependencies	4-50
4.18 Disclaimer	4-50
4.18.1 Model Validation	4-50
4.18.2 EIA-567A	4-50
Chapter 5	5-1
5.0 Variations on EIA-567	5-1
5.1 Introduction	5-1
5.1.1 Approach	5-1
5.1.2 Drivers and Checkers	5-1
5.2 Transition Time Modeling	5-1
5.2.1 The Output Driver and Transition Time Modeling	5-1
5.2.2 Output Driver Comparison	5-4
5.3 Electrical View Considerations	5-4
5.3.1 Linking Electrical View Classes to Pins	5-4
5.3.1 Load Circuit Considerations	5-5
5.4 Removing State Limitations on the Input Driver	5-5
5.4.1 The Original Driver	5-5
5.4.2 An Alternative Driver	5-5
5.5 Dealing with the "m_gen" Variable	5-6
5.5.1 Severity Levels	5-6
5.5.2 Signal Transitions	5-12
5.5.3 Output Limitations	5-12
5.5.4 Cycle Checking	5-12
5.6 The GET_TIMING Function	5-12
5.6.1 The Original Function	5-12
5.6.2 An Alternative Approach	5-12
5.6.3 Adding "x_gen" Flexibility	5-13
5.7 Pin and Signal Corrolation	5-13
5.8 Modeling Hierarchy	5-14
5.8.1 Limitations of the EIA-567 Approach	5-14
5.8.2 An Approach to Hierarchy	5-14
5.8.3 The Design Specification Package	5-15
5.8.4 The EIA_567_DS Package	5-35
5.8.5 The Electronic Data Sheet Model	5-35
5.9 Dealing with Multiple Timing Constraints per Pin	5-35
5.9.1 Constraint Selection through Generics	5-35
5.9.2 Utilization of Multiple Output Drivers	5-61
Chapter 6	6-1
6.0 Modeling Electronic Assemblies	6-1
6.1 Introduction	6-1
6.1.1 Background	6-1
6.1.2 Objective	6-1

Table of Contents - continued

6.2	General Purpose Parts and Components	6-1
6.2.1	Mechanical Parts	6-1
6.2.2	Electronic Components without Behavior	6-2
6.2.3	Electronic Components with Elementary Behavior	6-2
6.2.4	Connector Modeling	6-4
6.2.5	Modeling Printed Circuit Boards	6-11
6.3	EDS Components	6-17
6.3.1	Inherited Timing	6-17
6.3.2	Zero Timing	6-17
6.4	The Circuit Card Assembly	6-28
6.4.1	The CCA Model	6-28
6.4.2	Interface Connections	6-28
Chapter 7		7-1
7.0	WAVES Testbenches	7-1
7.1	Introduction	7-1
7.2	WAVES Standard Packages	7-1
7.2.1	The WAVES_SYSTEM Package	7-1
7.2.2	The WAVES_STANDARD Package	7-1
7.2.3	The WAVES_INTERFACE Package	7-1
7.2.4	The WAVES_OBJECTS Package	7-2
7.3	The WAVES Header File.	7-3
7.4	The WAVES Logic Value Package	7-4
7.4.1	Background	7-4
7.4.2	The LOGIC_VALUE Data Type	7-7
7.4.3	The VALUE_DICTIONARY Function	7-7
7.4.4	The PIN_CODES String	7-9
7.4.5	Analysis	7-10
7.5	The WAVES Frame Package	7-10
7.5.1	Event Timing	7-10
7.5.2	Analysis	7-10
7.6	The Device Under Test (DUT) Package.	7-10
7.6.1	Pin Order	7-10
7.6.2	Analysis	7-14
7.7	The Waveform Generator Package	7-14
7.7.1	Input and Output Pin Sets	7-14
7.7.2	The DQDWAV Procedure	7-14
7.7.3	Analysis	7-14
7.8	The Vector File	7-17
7.8.1	Comments	7-17
7.8.2	Vector Symbols	7-17
7.8.3	Vector Timing	7-17
7.8.4	Potential Problem Areas	7-17
7.9	The Testbench	7-17
7.9.1	Generics	7-17
7.9.2	The Device Under Test	7-17
7.9.3	Internal Signals	7-26
7.9.1.4	The "to_1164" Function	7-26
7.9.1.5	Reading Data	7-26
7.9.1.6	Data Translation	7-26
7.9.1.7	The Data Monitor	7-26
7.9.1.8	Errors	7-26
7.9.2	Analysis	7-27

List of Figures

Figure 2.3.1-1. A Sample VHDL Header.	2-4
Figure 2.5.1-1. The TIREP Model Structure.	2-9
Figure 2.6.2-1. A Model for the SN5476 J-K Flip-Flop.	2-12
Figure 3.4.1-1. The CD4029B Counter Core Model.	3-4
Figure 3.4.2-1. The CD4029 Design Specification File.	3-7
Figure 3.4.2.4.2-1. A Test Load Circuit.	3-20
Figure 3.4.3-1. The CD4029 EDS Model.	3-25
Figure 3.5.1-1. The CD4029 WAVES Header File.	3-33
Figure 3.5.2-1. The Wav_Frame Package.	3-35
Figure 3.5.3-1. The CD4029 DUT Package.	3-37
Figure 3.5.4-1. The CD4029 Waveform Generator Package.	3-38
Figure 3.5.5-1. An Error-Free Vector File.	3-41
Figure 3.5.5-2. An Error Seeded Vector File.	3-43
Figure 3.5.6-1. The CD4029 Testbench.	3-45
Figure 4.2.1-1. The Electronic Data Sheet Model.	4-2
Figure 4.3.1-1. A 12-Bit Population Counter.	4-3
Figure 4.4.1-1. The EIA_567_EV Package.	4-6
Figure 4.5.1-1. The FBG_EV Package.	4-11
Figure 4.6.1.2.2-1. Definition of T0 and T1.	4-15
Figure 4.6.1-1. The EIA_567_TV Package.	4-16
Figure 4.6.1.2.4-1. Definition of Setup and Hold Times.	4-21
Figure 4.7.1-1. The FBG_TV Package.	4-24
Figure 4.8.1-1. The EIA_567_PV Package.	4-27
Figure 4.9.1-1. The FBG_PV Package.	4-30
Figure 4.10.1-1. The EIA_567_TB Toolbox Package.	4-37
Figure 4.11.1-1. A Generic Input Driver.	4-40
Figure 4.12.1-1. A Generic Output Driver.	4-42
Figure 4.13.1-1. A Generic Bidirectional Driver.	4-44
Figure 4.15.1-1. A Generic Asynchronous Checker.	4-46
Figure 4.16.1-1. A Generic Synchronous Checker.	4-48
Figure 4.17.1-1. The EDS Model of the FBG Module.	4-51
Figure 5.2.1.4-1. An Enhanced Output Driver	5-2
Figure 5.4.2-1. A Generic Input Driver.	5-5
Figure 5.5-1. A Synchronous Checker.	5-7
Figure 5.5-2. An Asynchronous Checker.	5-9
Figure 5.8.2-1. An EIA-567 Library.	5-16
Figure 5.8.2.2-1. A Generic Bidirectional Driver.	5-24
Figure 5.8.3.1-1. The DQD Core Model.	5-27
Figure 5.8.3.2-1. The DQD Design Specification Package.	5-37
Figure 5.8.4-1. The EIA_567_DS Package.	5-49
Figure 5.8.5-1. The DQD EDS Model.	5-52
Figure 6.2.1-1. A Generic Part Model.	6-1
Figure 6.2.2-1. A Filter Capacitor Model.	6-3
Figure 6.2.3-1. A Resistor Network Model.	6-4
Figure 6.2.4.1-1. An Input Only Connector Model.	6-7
Figure 6.2.4.2-1. An Input/Output Connector Model.	6-8
Figure 6.2.5.1-1. A Graphics Package.	6-11
Figure 6.2.5.2-1. A Printed Circuit Board Model.	6-14
Figure 6.2.5.2-2. A Printed Circuit Board Outline.	6-16
Figure 6.3.3-1. An EDS Component Model.	6-19
Figure 6.4.1-1. A DQD CCA Model.	6-29
Figure 7.3-1. A WAVES Header File for the DQD Module.	7-6

List of Figures - continued

Figure 7.4-1. A WAVES Logic Value Package.	7-8
Figure 7.5-1. A WAVES Frame Package.	7-11
Figure 7.6-1. The DQD_DUT Package.	7-13
Figure 7.7-1. A Waveform Generator Package.	7-15
Figure 7.8-1. The DQD Module Test Vector File, a Partial Listing.	7-18
Figure 7.9-1. The DQD Module Testbench.	7-20

List of Tables

Table 2.1.2-1. SHARP TIREP Module Selections.	2-1
Table 2.2.1.4-1. VHDL File Extensions.	2-2
Table 2.2.1.6-1. DID Documentation Files.	2-2
Table 2.2.2.2-1. File Suffix Conventions.	2-3
Table 2.3.1-1. VHDL Header Requirements.	2-3
Table 2.4.1.3-1. Architecture Naming Conventions.	2-5
Table 2.4.2.3-1. Internal Signal Naming Conventions.	2-6
Table 2.7.1-1. Predefined VHDL Data Types.	2-14
Table 2.7.2-1. Predefined VHDL Operators.	2-14
Table 2.7.3.1-1. VHDL Reserved Words.	2-15
Table 2.7.3.2-1. VHDL Standard Package Reserved Words.	2-15
Table 2.7.3.3-1. TEXTIO Package Reserved Words.	2-16
Table 2.7.3.4-1. VHDL WAVES Reserved Words.	2-16
Table 2.7.3.5-1. IEEE-STD-1164 Reserved Words.	2-17
Table 2.7.3.6-1. EIA-567 Reserved Words.	2-17
Table 3.2.1-1. Static Electrical Characteristics.	3-1
Table 3.2.1-2. Dynamic Electrical Characteristics.	3-2
Table 3.2.2-1. Design Files Requiring No Modification.	3-3
Table 3.3-1. Files Requiring Context Clause Changes.	3-3
Table 4.4.1-1. EIA_567_EV Package Declarations.	4-5
Table 4.5.1-1. Design_EV Package Declarations.	4-10
Table 4.6.1-1. EIA_567_TV Package Declarations.	4-15
Table 4.6.1.2.3-1. State Identifiers, "signal_edge" Versus "std_logic".	4-21
Table 4.7.1-1. Design_TV Package Declarations.	4-23
Table 4.8.1-1. EIA_567_PV Package Declarations.	4-26
Table 4.9.1-1. Design_PV Package Declarations.	4-29
Table 4.10.1-1. EIA_567_TB Package Declarations.	4-36
Table 5.6.1-1. EDS Operating Points.	5-12
Table 5.8.2-1. EIA-567 Package Declarations.	5-15
Table 5.8.3.2-1. Design_DS Package Declarations.	5-36
Table 5.8.4-1. EIA-567_DS Package Declarations.	5-36
Table 7.2.1-1. WAVES_SYSTEM Package Declarations.	7-2
Table 7.2.2-1. WAVES_STANDARD Package Declarations.	7-3
Table 7.2.3-1. WAVES_INTERFACE Package Declarations.	7-4
Table 7.2.4-1. WAVES_OBJECTS Package Declarations.	7-5
Table 7.4-1. WAVES Logic Value Package Declarations.	7-7
Table 7.5-1. WAVES Frame Package Declarations.	7-10
Table 7.6-1. Device Under Test Package Declarations.	7-13

This page intentionally left blank

Index

A		
ALL_PINS		
see WAVES_OBJECTS package		
analysis	1-3	
Application Specific Integrated Circuit	1-3	
APPLY		
see WAVES_OBJECTS package		
arc		
see graphics package		
architecture		
behavioral	1-3	
body	1-3	
data flow	1-3	
names	2-5	
register transfer level	1-4	
structural	1-4	
area		
see graphics package		
ASIC		
see Application Specific Integrated Circuit		
ASYNCR		
see EIA_567 package		
see EIA_567_TV package		
ASYNCR_ARRAY		
see design specification package		
see EIA_567_PV package		
asyn_checker	4-1, 4-43, 5-15	
m_generation	5-6	
operation	4-43	
output limitations	5-12	
ASYNCR_CONSTRAINTS		
see EIA_567 package		
see EIA_567_TV package		
ASYNCRS		
see design specification package		
see design timing view package		
B		
b_driver	4-1, 5-15	
EV_SIGNAL_LIMIT	5-4	
pin_data	5-4	
behavioral architecture		
see architecture		
busses		
use of	4-5	
C		
CAPACITANCE		
see EIA_567 package		
see EIA_567_EV package		
CAPACITIVE		
see also WAVES logic value package		
see WAVES_STANDARD package		
CCA		
see circuit card assembly		
circuit card assembly	6-28	
circuit card assembly model	6-28	
INTERFACE_CONNECTIONS	6-28	
see model, circuit card assembly		
code readability		
see readability, code		
code walk-through	2-7	
duration	2-8	
meeting guidelines	2-8	
preparation	2-7	
purpose	2-7	
review team	2-7	
Commercial Component Model Specification		
basic approach	4-1	
EIA-567	1-6	
EIA-567A	1-1, 5-1, 5-5	
header requirements	1-1, 3-22	
limitations	2-3	
operating point requirements	5-14	
recommended changes to	4-14	
timing requirements	1-6	
complete model	4-23	
see model, complete		
compliant model		
see model, compliant		
component		
capacitor	6-2	
connector	6-4	
diode	6-4	
electronic data sheet	6-17	
INTERFACE_CONNECTIONS	6-2	
mechanical	6-1	
resistor network	6-2	
transistor	6-4	
component model		
see model, component		
COMPOUND		
see WAVES_STANDARD package		
configuration	3-31	
design	1-3	
names	2-6	
connector		
input only	6-4	
input/output	6-4	
connector model		
see model, connector		
constant		
names	2-6	
construct	1-3	
CONVERT_TO_UX01Z		

Index - continued

see EIA_567 package			
see EIA_567_EV package			
cor_pnt			
see graphics package			
core model			
see model, core			
CURRENT			
see EIA_567 package			
see EIA_567_EV package			
CURRENT_VECTOR			
see design electrical view package			
see design specification package			
D			
data flow architecture			
see architecture, data flow			
Data Item Description			
DI-EGDS-80811	1-1 - 1-2		
header requirements	2-3		
number of files	2-2		
operating point requirements	4-14		
operating requirements	4-29		
text files	2-2		
timing requirements	4-23		
delay			
see EIA_567 package			
see EIA_567_TV package			
propagation	1-4, 5-1, 7-10		
routing	3-30, 4-50		
see WAVES_INTERFACE package			
see WAVES_OBJECTS package			
DELAY_ARRAY			
see design specification package			
see EIA_567_PV package			
DELAY_TIME			
see WAVES_INTERFACE package			
DELAY_TIME_BASIS			
see WAVES_INTERFACE package			
DELAYS			
see design specification package			
see design timing view package			
design configuration			
see configuration, design			
design electrical view package	4-2, 4-10		
CURRENT_VECTOR	4-10		
ELECTRICAL_PIN_SPEC	4-10		
IMAX	4-10, 4-14		
LOWER_TEMPERATURE_LIMIT	4-10, 4-14		
pin classes	5-4		
UPPER_TEMPERATURE_LIMIT	4-10, 4-14		
VMAX	4-10, 4-14		
VMIN	4-10, 4-14		
VNOM	4-10, 4-14		
VOLTAGE_VECTOR		4-10	
design library			
see library, design			
design physical view package		4-2	
EDS		4-29	
EIA_STRING		4-29	
ELECTRONIC_DATA_SHEET		4-29	
ESD_PROTECTION		4-29, 4-35	
INTERFACE_CONNECTIONS		4-29, 5-13	
MAXIMUM_POWER DISSIPATION		4-29, 4-35	
OPNT		4-29	
OUTLINE_DRAWING		4-29, 4-35	
PART_NAME		4-29, 4-35	
PIN_AND_SIGNAL_CORROLATION		4-29	
PIN_INDEX		4-28 - 4-29, 5-4	
POINT_SPECIFICATION		4-29	
design specification elements			
see header, design specification elements			
design specification package		5-15	
ASYNC_ARRAY		5-36	
ASYNC		3-21, 5-36	
CURRENT_VECTOR		3-3, 5-36	
DELAY_ARRAY		5-36	
DELAYS		3-22, 5-36	
EDS		3-23, 5-36	
ELECTRICAL_PIN_SPEC		3-19, 5-36	
electronic data sheet		3-23	
ELECTRONIC_DATA_SHEET		3-24, 5-36	
ESD_PROTECTION		3-6, 5-36	
IMAX		3-20, 5-36	
incomplete specifications		3-22	
LOWER_TEMPERATURE_LIMIT		3-20	
LOWER_TEMPERATURE_LIMIT		5-36	
MAX_ASYNC_CONSTRAINTS_PER_PIN		3-24, 5-36	
MAX_DELAY_CONSTRAINTS_PER_PIN		3-24, 5-36	
MAX_SYNC_CONSTRAINTS_PER_PIN		3-24, 5-36	
MAXIMUM_POWER DISSIPATION		3-6, 5-36	
operating points		3-23	
OPNT		5-36	
OUTLINE_DRAWING		3-6, 5-36	
PART_NAME		3-6, 5-36	
PIN_AND_SIGNAL_CORROLATION		5-36	
PIN_INDEX		3-3, 5-36	
PIN_POINTERS		5-36	
POINT		5-36	
POINT_SPECIFICATION		5-36	

Index - continued

SYNC_ARRAY	5-36	part	6-1
SYNCS	3-22, 5-36	resistor_network	6-2
timing limitations	3-21	sn5476	2-11
tzero	6-17	sync_checker	3-3, 4-45, 5-6
UPPER_TEMPERATURE_LIMIT	3-20, 5-36	t_driver	4-41
VMAX	3-20, 5-36	see also unit, design	
VMIN	3-20, 5-36	wav_frame	3-32, 7-10
VNOM	3-20, 5-36	wav_logic	3-3, 7-4
VOLTAGE_VECTOR	3-3, 5-36	WAVES_INTERFACE	3-3
design timing view package	4-2	WAVES_INTERFACE package	7-1
ASYNCS	4-22 - 4-23	WAVES_OBJECTS	3-3, 7-2
DELAYS	4-23	WAVES_STANDARD package	7-1
MAX_ASYNC_CONSTRAINTS_PER_PIN	4-23	WAVES_SYSTEM	7-1
MAX_DELAY_CONSTRAINTS_PER_PIN	4-23	design unit naming conventions	
MAX_SYNC_CONSTRAINTS_PER_PIN	4-23	see naming conventions, design unit	
SYNCS	4-23	DETECT_EDGE	
design unit		see EIA_567 package	
async_checker	3-3, 4-43, 5-6	see EIA_567_TV package	
b_driver	3-3, 5-15	device under test package	
capacitor	6-2	TEST_PINS	3-34, 7-13
cd4029_core	3-3	DID	
cd4029_ds	3-3	see Data Item Description	
cd4029_dut	3-34	DIRECTION	
cd4029_eds	3-25	see also WAVES logic value package	
cd4029_tst	3-38	see WAVES_STANDARD package	
cd4029_wav	3-3, 3-34	direction_keyword	
connector	6-4	see WAVES_SYSTEM package	
dqd_core	5-15	DIRECTION_LIST	
dqd_ds	5-24	see WAVES_STANDARD package	
dqd_dut	7-10	DIRECTION_SUMMARY	
dqd_eds	5-35	see WAVES_STANDARD package	
dqd_pcb	6-14	direction_type	
dqd_tst	7-17	see WAVES_SYSTEM package	
dqd_wav	7-14	DISCONNECTED	
eia_567	3-3, 5-14	see also WAVES logic value package	
EIA_567_DS	3-2, 5-35	see WAVES_STANDARD package	
EIA_567_EV	4-5	DRIVE	
EIA_567_PV	4-23	see also WAVES logic value package	
EIA_567_TB	4-36	see WAVES_STANDARD package	
EIA_567_TV	4-14	E	
eia_cd4029_ds	3-3	EDGE_CHECK	
fbg_core	4-2	see EIA_567 package	
fbg_eds	4-50	see EIA_567_TB package	
fbg_ev	4-10	EDS	
fbg_pv	4-28	see design physical view package	
fbg_tv	4-22	see design specification package	
graphics	6-11	see electronic data sheet	
i_driver	3-3, 4-36, 5-5	EIA-567	
o_driver	3-3, 5-1	see Commercial Component Model Specification	
		EIA_567 package	5-14
		ASYNC	5-15

Index - continued

ASYNCH_CONSTRAINTS	5-15	EIA_567_TV package	4-2
CAPACITANCE	5-15	ASYNC	4-15
CONVERT_TO_UX01Z	5-15	ASYNCH_CONSTRAINTS	4-15
CURRENT	5-15	DELAY	4-15, 5-1
DELAY	5-15	DETECT_EDGE	4-15, 4-22
DETECT_EDGE	5-15	OUTPUT_DELAYS	4-15
EDGE_CHECK	5-15	POINT	4-14 - 4-15
EIA_STRING	5-15	relational functions	4-15, 4-22
ESD_CLASS	5-15	SIGNAL_EDGE	4-15, 4-21
EV_SIGNAL_LIMIT	5-15	SYNC	4-15, 4-21
EV_SIGNAL_LIMITS	5-15	SYNCH_CONSTRAINTS	4-15
F	5-15	VALID_OPERATING_POINT	4-15, 4-22
MAX	5-15	VOLTAGE_VECTOR	4-22
OPERATING_SELECTION	5-15, 6-17	EIA_STRING	
OUTPUT_DELAYS	5-15	see EIA_567 package	
PIN_RECORD	5-15	see EIA_567_PV package	
POWER	5-15	ELECTRICAL_PIN_SPEC	
RESISTANCE	5-15	see design electrical view package	
SIGNAL_EDGE	5-15	see design specification package	
SYNC	5-15	electronic data sheet	1-4, 4-1
SYNCH_CONSTRAINTS	5-15	see also design specification package	
TEMPERATURE	5-15	electronic data sheet model	
VOLTAGE	5-15	GET_TIMING	4-50
EIA_567_DS package	5-35	INTERFACE_CONNECTIONS	5-13
GET_TIMING	5-12, 5-36	see model, electronic data sheet	
relational operators	5-36	operation	4-50
VALID_OPERATING_POINT	5-36	power pins	4-50
EIA_567_EV package	4-2	routing delays	4-50
CAPACITANCE	4-5	VALID_OPERATING_POINT	4-50
CONVERT_TO_UX01Z	4-5, 4-10	ELECTRONIC_DATA_SHEET	
CURRENT	4-5	see design physical view package	
EV_SIGNAL_LIMIT	4-5, 4-9	see design specification package	
EV_SIGNAL_LIMITS	4-5	entity	
OPERATING_SELECTION	4-5, 4-9,	declaration	1-4
	6-17	names	2-5
POWER	4-5	errchk process	
RESISTANCE	4-5	see testbench	
TEMPERATURE	4-5	ESD_CLASS	
VOLTAGE	4-5	see EIA_567 package	
EIA_567_PV package	4-2	see EIA_567_PV package	
ASYNCH_ARRAY	4-26, 4-28	ESD_PROTECTION	
DELAY_ARRAY	4-26, 4-28	see design physical view package	
EIA_STRING	4-26	see design specification package	
ESD_CLASS	4-26	ETIME	
PIN_POINTERS	4-26, 4-28	see WAVES_INTERFACE package	
PIN_RECORD	4-26	see WAVES_OBJECTS package	
SYNCH_ARRAY	4-26, 4-28	EV_SIGNAL_LIMIT	
EIA_567_TB		see EIA_567 package	
GET_TIMING	5-12	see EIA_567_EV package	
EIA_567_TB package	4-2	see also ELECTRICAL_PIN_SPEC	
EDGE_CHECK	4-36	see also pin_data	
F	4-36	EV_SIGNAL_LIMITS	
GET_TIMING	4-36	see EIA_567 package	
MAX	4-36	EVENT	

Index - continued

see WAVES_INTERFACE package			
EVENT_LIST			
see WAVES_INTERFACE package			
EVENT_TIME			
see WAVES_INTERFACE package			
EVENT_TIME_BASIS			
see WAVES_INTERFACE package			
EVENT_VALUE			
see WAVES_STANDARD package			
event_value_element			
see WAVES_SYSTEM package			
F			
F			
see EIA_567 package			
see EIA_567_TB package			
file naming conventions			
see naming conventions, file			
FILE_SLICE			
see WAVES_OBJECTS package			
FILE_SLICE_LIST			
see WAVES_OBJECTS package			
files			
model	2-10		
source code	2-2		
text	2-2		
FRAME			
see WAVES_INTERFACE package			
FRAME_ELIST			
see WAVES_INTERFACE package			
FRAME_EVENT			
see WAVES_INTERFACE package			
FRAME_LIST			
see WAVES_INTERFACE package			
FRAME_SET			
see WAVES_INTERFACE package			
FRAME_SET_ARRAY			
see WAVES_INTERFACE package			
G			
generic			
checkers	3-31,4-1		
drivers	3-31,4-1		
names	2-6		
GET_TIMING			
see EIA_567_DS package			
see EIA_567_TB package			
see electronic data sheet model			
graphics package			
arc	6-11		
area	6-11,6-14		
cor_pnt	6-11		
hole	6-11		
length	6-11		
line	6-11		
origin	6-11,6-14		
H			
header			
author information	2-3		
design specification elements	2-5		
file	2-2		
module description	2-5		
order of revision listing	2-5		
organization information	2-3		
reference information	2-3		
revision date	2-5		
revision history	2-3		
status information	2-3		
supporting files	2-5		
version number	2-5		
HIGH			
see also WAVES logic value package			
see WAVES_STANDARD package			
hole			
see graphics package			
I			
i_driver			
alternate version	4-1,4-36,		
EV_SIGNAL_LIMIT	5-15		
operation	5-5		
original version	4-36		
pin_data	5-5		
state limitations	5-4		
state limitations	5-5		
IEEE-STD-1029.1			
see Waveform and Vector Exchange Specification			
IEEE-STD-1164			
see Standard Multi-Value Logic System for VHDL			
iih			
see EV_SIGNAL_LIMIT			
iiil			
see EV_SIGNAL_LIMIT			
IMAX			
see design electrical view package			
see design specification package			
incomplete specifications	3-22		
INDEX_SLICE			
see WAVES_OBJECTS package			
INDEX_SLICE_LIST			
see WAVES_OBJECTS package			
input_pins			
see waveform generator package			
INTERFACE_CONNECTIONS			
see circuit card assembly model			
see component			
see design physical view package			

Index - continued

- | | | |
|------------------------------------|-------------------------|--|
| see electronic data sheet model | | |
| ioh | | |
| see EV_SIGNAL_LIMIT | | |
| iol | | |
| see EV_SIGNAL_LIMIT | | |
| L | | |
| label | | |
| names | 2-6 | |
| leaf level module | | |
| see module, leaf level | | |
| length | | |
| see graphics package | | |
| library | | |
| design | 1-3 | |
| library unit | | |
| see unit, library | | |
| line | | |
| see graphics package | | |
| load circuits | 5-5 | |
| LOGIC_LIST | | |
| see WAVES_INTERFACE package | | |
| LOGIC_MAP | | |
| see WAVES_INTERFACE package | | |
| LOGIC_SET | | |
| see WAVES_INTERFACE package | | |
| LOGIC_VALUE | | |
| see WAVES logic value package | | |
| WAVES_INTERFACE package | 7-2 | |
| LOW | | |
| see also WAVES logic value package | | |
| see WAVES_STANDARD package | | |
| LOWER_TEMPERATURE_LIMIT | | |
| see design electrical view package | | |
| see design specification package | | |
| M | | |
| m_generation | 4-43, 5-1,
5-4, 5-6 | |
| MATCH | | |
| see WAVES_OBJECTS package | | |
| MATCH_CONTROL_TYPE | | |
| see WAVES_OBJECTS package | | |
| MAX | | |
| see EIA_567 package | | |
| see EIA_567_TB package | | |
| MAX_ASYNC_CONSTRAINTS_PER_PIN | | |
| see design specification package | | |
| see design timing view package | | |
| MAX_CONSTRAINTS_PER_PIN | | |
| see design specification package | | |
| MAX_DELAY_CONSTRAINTS_PER_PIN | | |
| see design specification package | | |
| see design timing view package | | |
| MAX_SYNC_CONSTRAINTS_PER_PIN | | |
| see design specification package | | |
| see design timing view package | | |
| MAXIMUM_POWER DISSIPATION | | |
| see design physical view package | | |
| see design specification package | | |
| MERGE_ETIME | | |
| see WAVES_INTERFACE package | | |
| MERGE_STRING | | |
| see WAVES_OBJECTS package | | |
| MIDBAND | | |
| see also WAVES logic value package | | |
| see WAVES_STANDARD package | | |
| model | | |
| circuit card assembly | 2-9 | |
| complete | 1-3 | |
| compliant | 1-3 | |
| component | 2-9 | |
| connector | 2-9 | |
| core | 2-8 | |
| electronic data sheet | 2-9, 3-25,
4-2, 4-50 | |
| electronic data sheet structure | 3-32 | |
| files | 2-10 | |
| miscellaneous component | 2-9 | |
| structure | 2-8 | |
| template | 3-1 | |
| module | | |
| leaf level | 1-4 | |
| VHDL | 1-5 | |
| module description | | |
| see header, module description | | |
| N | | |
| naming conventions | | |
| design unit | 2-5 | |
| file | 2-1 | |
| file name extensions | 2-1 | |
| filename | 2-1 | |
| source code | 2-2 | |
| UNIX versus PC-DOS | 2-2 | |
| NEW_FILE_SLICE | | |
| see WAVES_OBJECTS package | | |
| NEW_FRAME_SET | | |
| see WAVES_INTERFACE package | | |
| NEW_FRAME_SET_ARRAY | | |
| see WAVES_OBJECTS package | | |
| NEW_LOGIC_MAP | | |
| see WAVES_INTERFACE package | | |
| NEW_LOGIC_SET | | |
| see WAVES_INTERFACE package | | |
| NEW_PINSET | | |
| see WAVES_OBJECTS package | | |
| NEW_TIME_DATA | | |

Index - continued

see WAVES_OBJECTS package			
NO_PINS			
see WAVES_OBJECTS package			
null_event			
see WAVES_SYSTEM package			
null_temp			
see WAVES_SYSTEM package			
O			
o_driver	5-15		
EV_SIGNAL_LIMIT	5-4		
pin_data	5-4		
tri-stating	5-4		
OBSERVED			
see also WAVES logic value package			
see WAVES_STANDARD package			
operating point	1-4		
operating points	3-30		
see also design specification package			
OPERATING_SELECTION			
see EIA_567 package			
see EIA_567_EV package			
operator_kind			
see WAVES_SYSTEM package			
operator_type			
see WAVES_SYSTEM package			
OPNT			
see design physical view package			
see design specification package			
order of analysis	3-51		
electronic data sheet model	3-51		
testbench	3-51		
WAVES dataset	3-51		
origin			
see graphics package			
OUTLINE_DRAWING			
see design physical view package			
see design specification package			
OUTPUT_DELAYS			
see EIA_567 package			
see EIA_567_TV package			
output_pins			
see waveform generator package			
P			
package			
body	1-4		
body names	2-6		
declaration	1-4		
names	2-6		
PART_NAME			
see design physical view package			
see design specification package			
PCB			
see printed circuit board			
PIN_AND_SIGNAL_CORROLATION			
see design physical view package			
see design specification package			
PIN_CODE_LIST			
see WAVES_OBJECTS package			
PIN_CODE_STRING			
see WAVES_OBJECTS package			
PIN_CODES			
see WAVES logic value package			
see also WAVES_INTERFACE package			
pin_data			
see b_driver			
see i_driver			
see o_driver			
PIN_INDEX			
see design physical view package			
see design specification package			
pin_load			
see EV_SIGNAL_LIMIT			
PIN_POINTERS			
see design specification package			
see EIA_567_PV package			
PIN_RECORD			
see EIA_567 package			
see EIA_567_PV package			
PINSET			
see WAVES_OBJECTS package			
POINT			
see design specification package			
see EIA_567_TV package			
POINT_SPECIFICATION			
see design physical view package			
see design specification package			
pointer			
names	2-6		
port			
names	2-6		
POWER			
see EIA_567 package			
see EIA_567_EV package			
power pins	3-31		
electronic data sheet model	4-50		
implementation of	4-5		
operation	3-32		
PREDICTED			
see also WAVES logic value package			
see WAVES_STANDARD package			
printed circuit board model	6-11		
promote			
see WAVES_SYSTEM package			
propagation delay			
see delay			
pull_value			

Index - continued

see t_driver		
R		
READ_FILE_SLICE		
see WAVES_OBJECTS package		
readability		
capitalization	2-7	
code	2-7	
comments	2-7	
indentation	2-7	
obscuring or limiting	2-7	
spaces	2-7	
tabs	2-7	
reference designators	2-6	
reference information		
reserved words	2-15	
VHDL data types	2-14	
VHDL operators	2-14	
register transfer level architecture		
see architecture, register transfer level		
relational functions		
see EIA_567_TV package		
RELEVANCE		
see also WAVES logic value package		
see WAVES_STANDARD package		
relevance_keyword		
see WAVES_SYSTEM package		
relevance_type		
see WAVES_SYSTEM package		
REQUIRED		
see also WAVES logic value package		
see WAVES_STANDARD package		
RESISTANCE		
see EIA_567 package		
see EIA_567_EV package		
RESISTIVE		
see also WAVES logic value package		
see WAVES_STANDARD package		
RESPONSE		
see also WAVES logic value package		
see WAVES_STANDARD package		
revision history		
see header, revision history		
routing delay		
see delay, routing		
RTL		
see architecture, register transfer level		
S		
SHARP		
see Standard Hardware and Reliability Program		
SHARP_TIREP		
see TIREP Project		
signal		
names	2-6	
SIGNAL_EDGE		
see EIA_567 package		
see EIA_567_TV package		
signals		
internal	3-31	
simfrm		
see WAVES frame package		
simin		
see WAVES frame package		
simout		
see WAVES frame package		
simulation control variables		
see m_generation		
see x_generation		
Standard Hardware and Reliability Program		
1-1		
Standard Multi-Value Logic System for VHDL		
IEEE-STD-1164	1-1	
std_logic	2-6, 4-2	
the uninitialized state	4-2	
STATE		
see also WAVES logic value package		
see WAVES_STANDARD package		
state_keyword		
see WAVES_SYSTEM package		
state_type		
see WAVES_SYSTEM package		
state_value		
see WAVES_SYSTEM package		
std_logic	2-6	
STIMULUS		
see also WAVES logic value package		
see WAVES_STANDARD package		
STRENGTH		
see also WAVES logic value package		
see WAVES_STANDARD package		
strength_keyword		
see WAVES_SYSTEM package		
strength_type		
see WAVES_SYSTEM package		
strength_value		
see WAVES_SYSTEM package		
structural architecture		
see architecture, structural		
SUPPLY		
see also WAVES logic value package		
see WAVES_STANDARD package		
supporting files		
see header, supporting files		
SYNC		
see EIA_567 package		
see EIA_567_TV package		
SYNC_ARRAY		

Index - continued

see design specification package		see WAVES_SYSTEM package	
see EIA_567_PV package		system_waves_port_list	
sync_checker	4-1, 4-45, 5-15	see WAVES_SYSTEM package	
m_generation	5-6	T	
operation	4-45	t_driver	4-1, 4-41
signal transitions	5-12	operation	4-41
SYNC_CONSTRAINTS		pull_value	5-4
see EIA_567 package		tri-stating	5-4
see EIA_567_TV package		TAG	
SYNCS		see WAVES_OBJECTS package	
see design specification package		TEMPERATURE	
see design timing view package		see EIA_567 package	
synthesis	1-4	see EIA_567_EV package	
synthesizable	1-4	test load circuits	3-20
system_delay_time_basis		test_load	
see WAVES_SYSTEM package		see EV_SIGNAL_LIMIT	
system_delay_time_type		TEST_PINS	
see WAVES_SYSTEM package		see device under test package	
system_direction_summary		TEST_PINS_LIST	
see WAVES_SYSTEM package		see WAVES_OBJECTS package	
system_event		testbench	1-4, 3-38
see WAVES_SYSTEM package		component instantiation	3-50
system_event_time		component under test	3-38
see WAVES_SYSTEM package		errchk process	3-50
system_event_time_basis		internal signals	3-50
see WAVES_SYSTEM package		name	3-38
system_event_value		translate process	3-50
see WAVES_SYSTEM package		waveform generator procedure	3-50
system_frame		text files	
see WAVES_SYSTEM package		see files, text	
system_frame_list		TIME_DATA	
see WAVES_SYSTEM package		see WAVES_OBJECTS package	
system_frame_set		TIME_DATA_LIST	
see WAVES_SYSTEM package		see WAVES_OBJECTS package	
system_frame_set_array		TIMED_SLICE	
see WAVES_SYSTEM package		see WAVES_OBJECTS package	
system_logic_map		TIMED_SLICE_LIST	
see WAVES_SYSTEM package		see WAVES_OBJECTS package	
system_match_control		timing	
see WAVES_SYSTEM package		event	7-10
system_tag_string		inherited	6-17
see WAVES_SYSTEM package		WAVES vector file	7-17
system_tagged_event		zero	6-17
see WAVES_SYSTEM package		timing constraints	
system_tagged_flag		generic selection	5-35
see WAVES_SYSTEM package		multiple	5-35
system_tagged_temp		multiple output drivers	5-61
see WAVES_SYSTEM package		timing limitations	3-21
system_tagged_time		TIREP Project	1-1
see WAVES_SYSTEM package		approach	1-1
system_waves_match_list		model development	1-1
see WAVES_SYSTEM package		objectives	1-2
system_waves_port		references	1-2

Index - continued

DIRECTION	7-4 - 7-5	NEW_LOGIC_SET	7-4
DISCONNECTED	7-4 - 7-5	PIN_CODES	7-2
DRIVE	7-4 - 7-5	private constants	7-1
HIGH	7-4 - 7-5	private function	7-1
LOGIC_VALUE	7-7	WAVES_MATCH_LIST	
LOW	7-4 - 7-5	see WAVES_OBJECTS package	
MIDBAND	7-4 - 7-5	WAVES_OBJECTS package	7-1
OBSERVED	7-4, 7-7	ALL_PINS	7-5
PIN_CODES	7-7, 7-9	APPLY	7-5
PREDICTED	7-4, 7-7	DELAY	7-5
RELEVANCE	7-4, 7-7	ETIME	7-5
REQUIRED	7-4, 7-7	FILE_SLICE	7-5
RESISTIVE	7-4 - 7-5	FILE_SLICE_LIST	7-5
RESPONSE	7-4 - 7-5	FRAME_SET_ARRAY add	7-5
STATE	7-4 - 7-5	INDEX_SLICE	7-5
STIMULUS	7-4 - 7-5	INDEX_SLICE_LIST	7-5
STRENGTH	7-4 - 7-5	MATCH	7-5
SUPPLY	7-4 - 7-5	MATCH_CONTROL_TYPE	7-5
VALUE_DICTIONARY	7-7	MERGE_STRING	7-5
WAVES testbench		NEW_FILE_SLICE	7-5
data translation	7-26	NEW_FRAME_SET_ARRAY	7-5
device under test	7-17	NEW_PINSET	7-5
error handling	7-26	NEW_TIME_DATA	7-5
functional verification	7-26	NO_PINS	7-5
generics	7-17	PIN_CODE_LIST	7-5
internal signals	7-26	PIN_CODE_STRING	7-5
to_1164	7-26	PINSET	7-5
WAVES vector file		READ_FILE_SLICE	7-5
cautions	7-17	TAG	7-5
comments	7-17	TEST_PINS_LIST	7-5
symbols	7-17	TIME_DATA	7-5
timing	7-17	TIME_DATA_LIST	7-5
WAVES_INTERFACE package	7-1	TIMED_SLICE	7-5
DELAY	7-4	TIMED_SLICE_LIST	7-5
DELAY_TIME	7-4	WAVE_TIMING	7-5
DELAY_TIME_BASIS	7-4	WAVE_TIMING_LIST	7-5
ETIME	7-4	WAVES_MATCH_LIST	7-5
EVENT	7-4	WAVES_PORT_LIST	7-5
EVENT_LIST	7-4	WTIME_SLICE	7-5
EVENT_TIME	7-4	WTIME_SLICE_LIST	7-5
EVENT_TIME_BASIS	7-4	WAVES_PORT_LIST	
FRAME	7-4	see WAVES_OBJECTS package	
FRAME_ELIST	7-4	WAVES_STANDARD package	7-1
FRAME_EVENT	7-4	CAPACITIVE	7-3
FRAME_LIST	7-4	COMPOUND	7-3
FRAME_SET	7-4	DIRECTION	7-3
FRAME_SET_ARRAY	7-4	DIRECTION_LIST	7-3
LOGIC_LIST	7-4	DIRECTION_SUMMARY	7-3
LOGIC_MAP	7-4	DISCONNECTED	7-3
LOGIC_SET	7-4	DRIVE	7-3
LOGIC_VALUE	7-2	EVENT_VALUE	7-3
MERGE_ETIME	7-4	HIGH	7-3
NEW_FRAME_SET	7-4	LOW	7-3
NEW_LOGIC_MAP	7-4	MIDBAND	7-3

Index - continued

OBSERVED	7-3		
PREDICTED	7-3		
RELEVANCE	7-3		
REQUIRED	7-3		
RESISTIVE	7-3		
RESPONSE	7-3		
STATE	7-3		
STIMULUS	7-3		
STRENGTH	7-3		
SUPPLY	7-3		
UNKNOWN	7-3		
UNSPECIFIED	7-3		
WAVES_SYSTEM package	7-1		
direction_keyword	7-2		
direction_type	7-2		
event_value_element	7-2		
null_event	7-2		
null_temp	7-2		
operator_kind	7-2		
operator_type	7-2		
promote	7-2		
relevance_keyword	7-2		
relevance_type	7-2		
state_keyword	7-2		
state_type	7-2		
state_value	7-2		
strength_keyword	7-2		
strength_type	7-2		
strength_value	7-2		
system_delay_time_basis	7-2		
system_delay_time_type	7-2		
system_direction_summary	7-2		
system_event	7-2		
system_event_time	7-2		
system_event_time_basis	7-2		
system_event_value	7-2		
system_frame	7-2		
system_frame_list	7-2		
system_frame_set	7-2		
system_frame_set_array	7-2		
system_logic_map	7-2		
system_match_control	7-2		
system_tag_string	7-2		
system_tagged_event	7-2		
system_tagged_flag	7-2		
system_tagged_temp	7-2		
system_tagged_time	7-2		
system_waves_match_list	7-2		
system_waves_port	7-2		
system_waves_port_list	7-2		
unknown_keyword	7-2		
WTIME_SLICE			
see WAVES_OBJECTS package			
WTIME_SLICE_LIST			
see WAVES_OBJECTS package			
X			
x_generation		4-43, 5-1, 5-4, 5-13	

This page intentionally left blank

Chapter 1

1.0 A VHDL Modeling Guide

1.1 Background

1.1.1 SHARP TIREP

1.1.1.1 This document was developed under the Standard Hardware Acquisition and Reliability Program (SHARP) for the Technology Independent Representation of Electronic Products (TIREP) project. The TIREP project was directed toward the generation of a paperless design specification, based upon VHDL, which could be provided to manufacturing engineering personnel for the development of a production data package.

1.1.1.2 The TIREP project was initiated as a joint effort between the Naval Research Lab (NRL), the Naval Surface Warfare Center (NSWC), Crane and the Naval Air Warfare Center, Aircraft Division (NAWC-AD), Indianapolis. The project was kicked off in January 1992 and is scheduled for completion in FY-95.

1.1.1.3 This document is not intended to teach VHDL. Rather, it has been established to provide guidance and direction in the development of VHDL models which are compliant with the VHDL Data Item Description (DI-EGDS-80811). This document assumes a working knowledge of VHDL on the part of the reader.

1.1.1.4 The reader is encouraged not to be intimidated by the size of this document. In providing guidance, this document relies very heavily on example models and code. These example listings contribute significantly to the size of this document.

1.1.1.5 This document relies very heavily on VHDL standards, documents and modeling examples which have been previously developed. Key references are identified in section 1.2. The TIREP team would like to express their appreciation to the individuals who have put so much time and energy into the development of these resources.

1.1.2 Approach

1.1.2.1 The TIREP project was established with VHDL in mind. The intent was to determine the extent to which VHDL could be used in the development of a paperless specification for hardware.

1.1.2.2 For the TIREP project, a set of Standard Electronic Module (SEM) candidates were identified. These candidates were all digital, high use, format A (see MIL-STD-1389, paragraph 1.2) modules faced with current or near term obsolescence problems. Each design activity then selected a group of modules for which they would develop complete VHDL models.

1.1.2.3 With VHDL adopted as the language in which the TIREP modules would be modeled, WAVES (IEEE-STD-1029.1) was selected as the mechanism for the development of testbenches.

1.1.2.4 Early in the project it became evident that the VHDL BIT data type would not be adequate to describe model functionality. Therefore the IEEE-STD-1164 multi-value logic package was selected to provide additional versatility for modeling unknown and high impedance states. This package also provides flexibility in modeling forcing versus weak digital signals.

1.1.2.5 Finally, it was decided to make the TIREP models DID compliant (in accordance with DI-EGDS-80811). In order to do this, the TIREP team elected to make use of the EIA-567 modeling approach developed by Mr. Len Finegold for the F-22 program.

1.1.2.6 With this release of the modeling guide, EIA-567A has been incorporated for use with the TIREP modeling approach. It is noted that there are still a number of features which are required by the TIREP models which are not supported by EIA-567A. As a result, a modified modeling approach which implements the EIA-567A features and then builds upon these capabilities has been developed.

1.1.3 Model Development

1.1.3.1 In the TIREP project, the module specification (from MIL-M-28787) forms the baseline upon which the models were developed. Existing schematics and parts lists were obtained for the selected modules. VHDL behavioral models were developed for the module components which were then structurally tied together to realize the module level model.

1.1.3.2 Once the module level model was generated, and the electronic data sheet was built around the model based upon guidance provided through the EIA-567A modeling approach.

1.1.3.3 From this point, a WAVES testbench was developed. When possible, the functional vector set was obtained from the MIL-M-28787 specification. If not provided therein, an appropriate vector set was generated.

1.1.3.4 Once an operational testbench was established, a behavioral model was generated for the module. The testbench was then used to validate this new model.

1.1.3.5 In the next phase of the modeling effort, the new behavioral model was targeted for hardware. In general, this targeting was accomplished through synthesis to a field programmable device. In some instances, component level replacements were recommended.

1.1.3.6 Finally, a circuit card assembly model was developed using structural VHDL. This model was transferred to a manufacturing site for fabrication.

1.1.4 Project Objectives

1.1.4.1 To develop paperless design specifications for selected SHARP modules. VHDL was selected as the vehicle by which to accomplish this. As part of this task, it was desired to capture as much of the MIL-M-28787 module specification as possible in the VHDL model for the module.

1.1.4.2 VHDL models developed shall meet the requirements of the VHDL Data Item Description (DID DI-EGDS-80811).

1.1.4.3 VHDL models developed shall include a WAVES testbench in accordance with IEEE-STD-1029.1.

1.1.4.4 The development of processes to provide for module fabrication from the VHDL models, including the transfer of selected models to a manufacturing facility for fabrication, and module validation testing at NSWC, Crane.

1.1.4.5. The generation of this modeling guide.

1.2 Reference Documents

1.2.1 Baseline References

The SHARP TIREP project has drawn heavily from the following sources. It should be noted that several holes exist in the version of the F-22 VHDL Model Specification (document 5PTA3009) which was available to the SHARP TIREP team. As a result original code was developed for some of the packages and interface components originally described in this package. Since that time, a complete model of the Viterbi Decoder has been obtained by the TIREP team. It is these packages and interface components which will be detailed in Chapter 3. The original code developed by the TIREP team will be detailed in Chapter 4.

<i>Document</i>	<i>Title</i>
<i>IEEE-STD-1076</i>	<i>VHDL Language Reference Manual</i>
<i>IEEE-STD-1029.1</i>	<i>Waveform and Vector Exchange Specification (WAVES)</i>
<i>IEEE-STD-1164</i>	<i>Multivalued Logic System for VHDL Model Interoperability (Std_logic_1164)</i>
<i>EIA-567A</i>	<i>Commercial Component Model Specification</i>
<i>DI-EGDS-80811</i>	<i>VHDL Documentation Data Item Description</i>
<i>5PTA3009</i>	<i>F-22 VHDL Model Specification (dated 4 March 1992)</i>
<i>(no number)</i>	<i>A User's Guide to WAVES (draft version 4.4, dated 10 December 1990); prepared by the WAVES Analysis and Standardization Group</i>
<i>MIL-M-28787</i>	<i>General Specification for Standard Electronic Modules</i>

1.2.2 Supporting Documents

There are a number of supporting documents for VHDL which have been identified as a result of the SHARP TIREP activities. These documents are noted below.

<i>Document</i>	<i>Title</i>
<i>MIL-STD-454</i>	<i>Standard General Requirements for Electronic Equipment</i>
<i>Requirement 64</i>	<i>Microelectronic Devices</i>

<i>MIL-STD-1389</i>	<i>Design Requirements for Standard Electronic Modules</i>
<i>FIPS-PUB-172</i>	<i>VHSIC Hardware Description Language (VHDL)</i>
<i>(no number)</i>	<i>VHDL Model Verification and Acceptance Procedure (version 1.0); prepared by Rome Laboratory/ERDD</i>
<i>(no number)</i>	<i>Army Handbook, The Documentation of Digital Electronic Systems with VHDL (Preliminary Final Draft dated 18 November 1993);</i>
<i>ANSI Y32.2</i>	<i>Graphic Symbols for Electrical and Electronic Diagrams</i>

1.3 Definitions

1.3.1 Analysis

Analysis of a design unit defines the corresponding library unit in a design library. This is the process of converting the VHDL source code into a machine executable format.

1.3.2 Application Specific Integrated Circuit (ASIC)

A custom or semi-custom integrated circuit, generally digital, designed to perform a custom function for a specific application.

1.3.3 Architecture Body

An architecture body is a VHDL design unit which is used to describe the behavior or structure of a VHDL entity. There is only one entity declaration permitted for a given architecture body, however, multiple architecture bodies may be generated for a single entity declaration.

1.3.4 Behavioral Architecture

The functional or algorithmic architecture of a design. Behavioral VHDL is generally based upon sequential VHDL processes.

1.3.5 Complete Model

A VHDL model is complete if it defines the interface and behavior of a design and it includes an electronic data sheet, a testbench and supporting information which explicitly describes the characteristics of the design.

1.3.6 Compliant Model

A complete VHDL model is compliant if it meets the requirements of the VHDL DID (DI-EGDS-80811).

1.3.7 Construct

One or more statements grouped into a logical unit (e.g. an entity, a process, an architecture, a function, a signal assignment, etc.).

1.3.8 Data Flow Architecture

The architecture of a design expressed as a set of data transformations. Data Flow VHDL is structured using concurrent VHDL signal assignment statements in order to compute output signal values. It should be noted that many sources do not differentiate between behavioral and data flow VHDL architectures.

1.3.9 Design Configuration

The design configuration may be generated as part of a structural VHDL architecture, or it may be provided as a separate VHDL design unit. The purpose of the design configuration is to provide the engineer with the freedom to select different architectures for use in a VHDL model.

1.3.10 Design Library

A design library is a collection of one or more analyzed VHDL design units. A design library may reside anywhere on the development platform, as long as a map is provided to indicate where the library can be found. The default design library is the "work" library as defined in paragraph 11.2 of the VHDL LRM (IEEE-STD-1076).

1.3.11 Design Unit

A VHDL design unit is an analyzable segment of VHDL code. Allowable VHDL design units include entity declarations, architecture bodies, package declarations, package bodies, and design configurations.

1.3.12 Electronic Data Sheet (EDS)

An electronic data sheet is comprised of a set of VHDL packages which describe parameters associated with a hardware implementation of a VHDL model. These parameters include, but are not limited to, interface voltage and current levels, timing characteristics, operating points, etc.

1.3.13 Entity Declaration

An entity declaration is a VHDL design unit which is used to the interface between a design entity and the environment in which it operates. Each entity declaration should be supported by at least one architecture body.

1.3.14 Leaf Level Module

Leaf level modules are VHDL modules for which no VHDL structural body is required.

1.3.15 Library Unit

The machine executable code resulting from the analysis of a VHDL design unit.

1.3.16 Operating Point

An operating point is a specific set of operating conditions declared for a simulation.

1.3.17 Package Body

A package body is a VHDL design unit which defines the operation of subprograms defined in the package declaration. Additionally, the value assignment of constants identified in the package declaration may be deferred to the package body.

1.3.18 Package Declaration

A VHDL package declaration is a VHDL design unit which is used to describe data types, deferred constants and subprograms which will be made available to other VHDL design units within the same design environment. Subprograms (functions and procedures) identified in the package declaration must be defined in the package body. Constants established in the package declaration may be assigned values in the package body (hence, the constant assignment is deferred).

1.3.19 Propagation Delay

The period of time measured from the 50% transition point of an input signal to the corresponding 50% transition point of an output signal.

1.3.20 Register Transfer Level (RTL) Architecture

A register transfer level architecture describes a design in terms of registers and combinational circuitry.

1.3.21 Structural Architecture

The design architecture which most closely models hardware. Structural VHDL is comprised of interconnected lower level components. It therefore provides a "netlist" representation of the design.

1.3.22 Synthesis

The process of translating a VHDL design entity into a hardware design such as an ASIC gate-level schematic.

1.3.23 Synthesizable

A VHDL design entity is considered to be synthesizable when a selected synthesis tool is capable of translating the entity into a hardware design. It should be noted that a design entity which is synthesizable by one synthesis tool, may not be synthesizable by a different synthesis tool.

1.3.24 Testbench

A VHDL testbench is a VHDL module which applies stimuli to the VHDL module under test (MUT) and compares the MUT's responses with the expected output for the module. The testbench will report any discrepancies encountered during the simulation of the MUT.

1.3.25 Transition Time

The period of time measured from the 10% transition point to the 90% transition point on an output signal.

1.3.26 Very High Speed Integrated Circuit (VHSIC)

A government sponsored program designed to push the leading edge of technology in the development of ASIC's.

1.3.27 VHDL Module

A VHDL module consists of an entity declaration, one or more behavioral VHDL bodies, and except for allowable leaf level modules, a structural VHDL body.

1.3.28 VHSIC Hardware Description Language (VHDL)

Developed under the VHSIC program as a tool to capture and document the intent of an ASIC design. Basically, a software language which is used to describe hardware. IEEE-STD-1076 is the language reference manual for VHDL.

1.3.29 Waveform and Vector Exchange Specification (WAVES)

A proper subset of VHDL constructs used to describe test/vector information for a VHDL design (IEEE-STD-1029.1).

1.3.30 WAVES Header File

A WAVES header file identifies the files which comprise the WAVES dataset, their intended use and the order of analysis for the dataset.

1.4 Acronyms

<i>ASIC</i>	<i>Application Specific Integrated Circuit</i>
<i>CCA</i>	<i>Circuit Card Assembly</i>
<i>DID</i>	<i>Data Item Description</i>
<i>DoD</i>	<i>Department of Defense</i>
<i>DUT</i>	<i>Device Under Test</i>
<i>EDS</i>	<i>Electronic Data Sheet</i>
<i>EIA</i>	<i>Electronics Industry Association</i>
<i>IEEE</i>	<i>Institute of Electrical and Electronics Engineers</i>
<i>LRM</i>	<i>Language Reference Manual</i>
<i>MUT</i>	<i>Module Under Test</i>
<i>PCB</i>	<i>Printed Circuit Board</i>
<i>RTL</i>	<i>Register Transfer Level</i>
<i>SEM</i>	<i>Standard Electronic Module</i>
<i>SHARP</i>	<i>Standard Hardware and Reliability Program</i>
<i>TIREP</i>	<i>Technology Independent Representation of Electronic Products</i>
<i>VHDL</i>	<i>VHSIC Hardware Description Language</i>
<i>VHSIC</i>	<i>Very High Speed Integrated Circuit</i>
<i>WAVES</i>	<i>Waveform And Vector Exchange Specification</i>

1.5 Chapter Overview**1.5.1 Chapter 1**

This chapter provides a high level overview for the SHARP TIREP project and introductory information for this modeling guide. Included in this chapter are reference documents, definitions and acronyms.

1.5.2 Chapter 2

Chapter 2 performs several functions. First, it provides recommended modeling conventions and style guide information. Secondly, it provides an overview of the TIREP model structure and content. Finally, this chapter offers some key VHDL reference tables.

1.5.3 Chapter 3

Chapter 3 has been established to provide a "quick start" model development guide. In this chapter, the design engineer is escorted through the development of a design example. This chapter will not address the details of the model content. Rather, it will show the engineer how to adapt the TIREP modeling approach to a new design.

1.5.4 Chapter 4

The modeling approach presented in Chapter 4 is based upon the EIA-567A approach. Although a significant departure from the original EIA-567, most of the information captured through the original approach is captured under the new approach. One of the key advantages of EIA-567A is that it provides a mechanism for modeling multiple delay constraints on a pin. The approach offered is not without its limitations, it does provide additional flexibility in this area.

1.5.5 Chapter 5

During the course of the SHARP TIREP model development, several optional and/or recommended changes have been identified for EIA-567A approach outlined in Chapter 4. These changes and the reasons for them will be detailed in Chapter 5.

1.5.6 Chapter 6

Chapter 6 carries the VHDL model beyond the integrated circuit or ASIC level to the assembly level, including the modeling of components other than integrated circuits.

1.5.7 Chapter 7

The Waveform and Vector Exchange Specification (WAVES) was adopted for use in the SHARP TIREP model testbenches. Although getting started in WAVES can be a somewhat tedious undertaking, Chapter 6 will strive to introduce WAVES and provide a "head start" to testbench development.

This page intentionally left blank