© 2014 IJIRT | Volume 1 Issue 11 | ISSN: 2349-6002

VHDL MODELING STYLE

Jyoti Yadav,Kriti Bhatia,Kirti Kaushik
Department of Computer Science , MDU

***Abstract*- VHDL (Hardware Description Language) It is a hardware discriptive language used in electronic devices to describe systems such as VHDL. It can also be used as a general purpose . Basically in this various type of modeling is done for the gates and other logic devices. In this we do different type of modeling as entity declaration , datat flow declaration.**

I. INTRODUCTION

The idea of being able to simulate the ASICs from the information in this documentation was attractive so that A developed that could read the VHDL files. The next step was the development of tools that read the VHDL, and output a definition of the physical implementation of the circuit.

Due to the Department of Defense requiring as much of the syntax as possible to be based on Ada, in order to avoid re-inventing concepts that had already been thoroughly tested in the development of Ada,] VHDL borrows heavily from the in both concepts .

The initial version of VHDL, designed to standard IEEE 1076-1987, included a wide range of data types, including numerical (and, logical (and, and, plus of bit called bit_vector and of character.

A problem not solved by this edition, however, was "multi-valued logic", where a signal's (none, weak or strong) and unknown values are also considered. This required, which defined the 9-value logic types: scalar std_logic and its vector version std_logic_vector.

The updated, in 1993, made the syntax more consistent, allowed more flexibility in naming, extended the character type to allow ISO-8859-1 printable characters, added the xnor operator, etc]

Minor changes in the standard (2000 and 2002) added the idea of protected types (similar to the concept of class in C++) and removed some restrictions from port mapping rules.

II. DESIGN

VHDL is commonly used to write text models that describe a logic circuit. Such a model is processed by a synthesis program, only if it is part of the logic design. A simulation program is used to test the logic design using simulation models to represent the logic circuits that interface to the design. This collection of simulation models is commonly called a *testbench*.

VHDL has constructs to handle the inherent in hardware designs, but these constructs (*processes*) differ in syntax from the parallel constructs in Ada (*tasks*). Like Ada, VHDL is and is. In order to directly represent operations which are common in hardware, there are many features of VHDL which are not found in Ada, such as an extended set of Boolean operators including **nand** and **nor**. VHDL also allows arrays to be indexed in either ascending or descending direction; both conventions are used in hardware, whereas in Ada and most programming languages only ascending indexing is available.

VHDL has file input and output capabilities, and can be used as a general-purpose language for text processing, but files are more commonly used by a simulation testbench for stimulus or verification data. There are some VHDL compilers which build executable binaries. In this case, it might be possible to use VHDL to write a *testbench* to verify the functionality of the design using files on the host computer to define stimuli, to interact with the user, and to compare results with those expected. However, most designers leave this job to the simulator.

It is relatively easy for an inexperienced developer to produce code that simulates successfully but that cannot be synthesized into a real device, or is too large to be practical. One particular pitfall is the accidental production of rather than as storage elements.

III. STANDARDIZATION

The Standard 1076 defines the or VHDL. It was originally developed under contract F33615-83-C-1003 from the awarded in 1983 to a team with Intermetrics , Inc. as language experts and prime contractor, with as chip design experts and as computer system design experts. The language has

undergone numerous revisions and has a variety of sub-standards associated with it that augment or extend it in important ways.
1076 was and continues to be a milestone in the design of electronic systems

- 1076-1987 First standardized revision of ver 7.2 of the language from the United States Air Force.
- 1076-1993 (also published with Significant improvements resulting from several years of feedback. Probably the most widely used version with the greatest vendor tool support.
- 1076-2000Minor revision. Introduces the use of *protected types*.
- 1076-2002 Minor revision of 1076-2000. Rules with regard to *buffer ports* are relaxed.
- 1076-2008 (previously referred to as 1076-200x) Major revision released on 2009-01-26. Among other changes, this standard introduces the use of *external name*.
  standards
- IEEE 1076.1 VHDL Analog and Mixed-Signal
- IEEE 1076.1.1 VHDL-AMS Standard Packages
- IEEE 1076.2 VHDL Math Package
- IEEE 1076.3 VHDL Synthesis Package
- IEEE 1076.3 VHDL Synthesis Package - Floating Point
- IEEE 1076.4 Timing

IV. ADVANTAGES

The key advantage of VHDL, when used for systems design, is that it allows the behavior of the required system to be described (modeled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires).
Another benefit is that VHDL allows the description of a. VHDL is, unlike procedural computing languages such as BASIC, C, and assembly code, which all run sequentially, one instruction at a time.
A VHDL project is multipurpose. Being created once, a calculation block can be used in many other projects. However, many formational and functional block parameters can be tuned (capacity parameters, memory size, element base, block composition and interconnection structure).
A VHDL project is portable. Being created for one element base, a computing device poject can be ported on another element base, for example VLSI are various technologies.

REFERENCES

[1] Through various websites related to the DIGITAL SYSTEM DESIGN such as :
En.m.wikipedia.org/wiki/dsd
www.tutorialspoint.com
www.computerhope.com
[2] Through various books of digital system design.