VHDL Simulation with Access to Transistor Models

J. Gibson
NASA SERC for VLSI Systems Design
University of Idaho
Moscow, Idaho 83843

1 Introduction

Hardware description languages such as VHDL have evolved to aid in the design of systems with large numbers of elements and a wide range of electronic and logical abstractions. For high performance circuits, behavioral models may not be able to efficiently include enough detail to give designers confidence in a simulation's accuracy. One option is to provide a link between the VHDL environment and a transistor level simulation environment. The coupling of the Vantage Analysis Systems VHDL simulator and the NOVA simulator provides the combination of VHDL modeling and transistor modeling.

2 Vantage VHDL Simulator

The Vantage Analysis Systems VHDL simulation environment is a full implementation of the IEEE 1076 VHDL Standard. The Vantage system is entirely written in "C". Hierarchical designs from Mentor Graphics' NetEd, EDIF or other schematics can be imported into Vantage. The Vantage system compiles VHDL designs and simulates their behavior either interactively or in a batch mode. Incremental symbol or schematic changes can be made in the Vantage environment, from which structural VHDL can be automatically be generated. The created or edited schematics can be exported back into the original schematic environment. Connectivity and structural checks are made by the schematic viewer.

Results may be viewed as waveforms or as entries in a table, and can be viewed as the simulator is running or after the simulation run has completed. Circuit node values can also be displayed on the associated node in the circuit schematic.

The Vantage simulation control gives the user source code level breakpoints and triggering capability based on a very wide range of conditions specified by the user. Convenient viewing and the manipulation of the values of signals, variables and constants is provided. Breakpoints can be based on boolean expressions, change in a signal, source code lines, or by design units (by instance or globally).

In the Vantage system, the VHDL source code is parsed by a C code generator. Then the host "C" code compiler prepares the executable file or files. The Vantage Intermediate Format is used for the generation of the "C" code. All design units lower in hierarchy must have older time stamps than the designs that reference them. The Vantage system automatically recomiles all "out-of-date" design units referenced during a recompile.

Several conveniences are provided by the Vantage system, including automatic mapping

of signal names that do not conform to the VHDL signal name convention to an internal, VHDL compatible form. This permits familiar names of existing systems to be used when interfacing with the Vantage simulator. The Vantage Control Language facilitates the generation of test vectors.

Extensive libraries of vendor supplied VHDL models, parts and packages, from SSI to VLSI, are available. Any VHDL in a Vantage library can be exported to an ascii file. Vantage also supplies a concurrent compiler that spreads the compilation task of a design across a network, to improve compilation speed.

3 The NOVA Simulator

NOVA is a logic simulator that was recently developed at the University of Idaho NASA Space Engineering Research Center for VLSI Systems Design. NOVA is a second generation design, targeted for designs of up to a few million primitives (transistors and logic gates). NOVA has been used to simulate integrated circuits designed for the NASA Space Station and Explorer missions and other NASA projects, and for Hewlett Packard disk and tape drives. NOVA presently is ported to HP 9000 Series 300, 400 and 700's, the HP/Apollo DN10000, the Cray X-MP and NeXT systems. Behavioral models can be used in NOVA to assist in the architectural definition of major functional blocks before circuit details are completely known, and to improve simulation performance at most levels of system modeling.

Structural description is accomplished using the BOLT or HP Block Description (BDL) languages. NOVA utilizes hierarchical design methodology, allowing designs to be conveniently partitioned. Efficient management of design complexity is made possible by the block oriented circuit description. SCIP schematics, based on Hewlett Packard design tools, provide schematic documentation, from which the BOLT design description can be extracted.

NOVA supports synchronous and asynchronous modeling of hierarchical designs using logic primitives and intrinsic devices. Most types of transistors and logic gates are represented in the existing library, including bidirectional CMOS devices and bidirectional transmission gates. If the designer requires new primitives to accurately model a special circuit, NOVA provides a means of incorporating the user defined model.

NOVA provides for full timing analysis of combinational and sequential circuits, specified by rise and fall delays, using a timing wheel based simulation engine.

Behavioral modeling, using a "C" based functional model capability, allows the designer to generate high level descriptions of a block of circuitry. Productivity is improved by allowing the designer to simulate the function of a block before the detailed circuit implementation is available. Very good behavioral modeling performance is achieved by compiling the functional models with the simulation engine. Transistor or logic gate circuit models can be mixed with, or replace, behavioral models, with full timing and delay modeling capability.

A configurable X11 graphics user interface assists the designer in viewing and interpret-

ing simulation results. Signals and busses can be viewed as waveforms. Trigger conditions can be defined to find specific signal relationships in the simulation output, allowing in depth analysis of complex events.

NOVA also provides numerous analysis features, such as node coverage, simulation debugging, output formatting, node forcing, simulation state logging and saving, and others.

Software tools in NOVA greatly simplify test vector development for design verification. A test vector programming language makes it possible for the designer to develop compact descriptions of complex simulation sequences.

The overall capability of the NOVA simulator closely matches that of many commercial simulators. A major issue is the fact that NOVA's behavioral modeling capability is not close to any industry standards, like VHDL, which makes it difficult for NOVA users to leverage existing model libraries or to import existing designs. On the other hand, the transistor level modeling in NOVA is highly evolved and well known by the NOVA user community. Using VHDL as a modeling language will likely open many opportunities for an organization like the NASA SERC for VLSI Systems Design, compared to using a proprietary modeling language.

4 Multiple Value Logic Systems

Simulations performed at the logic level of abstraction describe a digital circuit in terms of primitive logic functions such as NAND, NOR, etc., and allow for the nets interconnecting the logic functions to carry states of zero, one, unknown, and high impedance. In the case of NOVA, strengths of active, resistive and floating accompany the logic states, to provide a total of twenty-two logic values. These twenty-two logic values are built into the structure of the simulator and are possible to change or expand, but not necessarily easily.

The VHDL standard provides a Multi-Value Logic structure that allows the individual user of VHDL to tailor the resolution of logic values to satisfy the needs of a general design methodology or the specific preferences of individuals. This flexibility in describing logic values in a digital system can have a considerable influence on the transportability of a VHDL model from one design group to another. Having too few logic values can cause erroneous results in hardware systems that have bidirectional data busses, open-collector or high-impedance conditions. More logic values are necessary to model open-collector devices with pull-up resistors and situations that can occur when initializing a digital system.

For the coupling of NOVA and the Vantage VHDL simulator, the model compatibility issue is resolved by using the same types of logic states and strengths for both simulation systems. The Multi-Value logic system of the Vantage system is set to represent the same values as NOVA, with the resolution functions providing the same logic value when circuit outputs are connected together.

5 Transistor Level Performance Issues

One of the primary motivations for this work is the acceleration of the performance of transistor, or switch level, simulations in the VHDL environment. During the original design of the VHDL language, transistor level simulation was not included as a primary requirement. However, algorithms have been developed in VHDL that can simulate the properties of bidirectional transmission gates, without extensions to the VHDL language.

Given that VHDL can model bidirectional pass transistor networks, a second issue is the amount of memory required to represent a primitive in a VHDL simulator compared to a more "hardwired" simulator that has semantics built into it's runtime kernel, such as NOVA or Verilog. NOVA uses about 35 bytes per primitive (transistor) in the internal representation. The amount of memory required for the average primitive in VHDL is not as easily determined. Based on overall file size, it appears that as much as 1000 bytes of data are associated with each primitive in the Vantage simulation system, a VHDL system that is known for relatively good performance. It is common for VHDL system models to require virtual memory, which automatically invokes at least a 10X performance penalty, relative to a simulation model that runs entirely in RAM. A 500,000 NOVA transistor model, entirely composed of transistor primitives without any use of behavioral models, will fit in a workstation's 32 megabyte random access memory.

Another difference between the Vantage VHDL modeling system and NOVA is the use of resolution functions. A resolution function, applied to a node in a circuit, is used to return the value of a signal when the signal is driven by multiple drivers, during a simulation. All VHDL signals with multiple drivers must have a resolution function tied to that signal. With VHDL, the designer has the capability of defining any type of resolution function desired, either wired-OR, wired-AND or average signal value. NOVA has the equivalent of a resolution function, but it is coded in optimized "C", tightly linked with the rest of the core of the simulator and is fixed in definition. A VHDL resolution function is written in VHDL as a package, a representation that will be translated into "C" code but not optimized for fast execution.

6 "C" Behavioral Model Interface to the Vantage Simulator

A complete simulation system should be able to efficiently and quickly incorporate algorithms not represented in the native language of the simulator. To make possible fast development of designs, algorithms that already exist in program form should be usable in system simulations without requiring a reimplementation. If a design under development uses, for example, output from a digital filtering algorithm that is going to be part of another integrated circuit, it may be of considerable advantage to use a high level programming language version of that algorithm. Developing a new implementation of a digital filtering algorithm is not only a duplication of effort; new sources of error and changes in performance may also result.

Both NOVA and the Vantage Simulator allow "C" based behavioral models to be compiled into each simulation environment. In NOVA, a BOLT description is written for the behavioral model, describing the input and output connections to the rest of the simulation model. The NOVA "C" behavioral model is compiled in "C" and the resulting object data is compiled with NOVA into a form containing the regular primitive based simulation environment and the functional model. In the Vantage VHDL environment, the process is similar, in that the user must provide an entity (the structural or input/output description) written and compiled in VHDL prior to compiling the combined VHDL/C architecture. The architecture (the behavior of the module) is written and compiled in "C". Both systems provide the necessary parameters required for passing state, strength and timing information between the behavioral model and the main simulation model. Again, using an industry standard language, such as VHDL, as the modeling interface, should provide more flexibility and opportunity in the future.

7 "C" Based Simulator Interface to the Vantage Simulator

Using the Vantage Simulator and NOVA is one way of meeting the dual goals of using an industry standard behavioral modeling language and achieving decent transistor level simulation performance. It is presently estimated that the transistor level simulation performance of NOVA will exceed that of the Vantage Simulator by 20 to 50 times. The software to accomplish the link between NOVA and the Vantage Simulator is expected to be available from Vantage in the near future.

8 Future Directions

Research is in progress to identify simplier behavioral modeling methodologies that are quicker and easier to use. The objective is to reduce design time by having only one complete representation of a design, first as a top level behavioral model which is then broken in to subsystems of the design as the function of each block is identified. At any time, either the behavioral or transistor level representation of a block can be used. For performance reasons, the behavioral models can be used. For detailed circuit timing and performance analysis, the transistor level representations of the blocks being designed can be used while the rest are left at the behavioral level. VHDL is satisfactory as the modeling language for this effort, since description standards will be a large part of the solution to an easier to use simulation environment. In a complementary effort, software tools are being developed by the University of Idaho Computer Science Department to compare the functionality of behavioral models and transistor level models.