The Analysis of Modeling Styles for System Level VHDL Simulations

Andrew P. Voss, Robert H. Klenke, James H. Aylor,
Department of Electrical Engineering
University of Virginia
Charlottesville, VA 22903

Abstract

This paper presents the results of a study to examine the effects of various VHDL model characteristics on simulation execution times. Four different modeling characteristics of complex VHDL models were examined: the size of signals, the use of file input and output (I/O) operations, the use of bus resolution functions, and the overall size and complexity of VHDL models. To develop models and tests for these characteristics, the University of Virginia's Advanced Design Environment Prototyping Tool (ADEPT) was used. This performance modeling environment provided an easy framework to develop tests for the four characteristics to be examined.

After developing the different tests to examine these characteristics, multiple runs were conducted to minimize random variations due to processor loading. The results of these tests are presented here along with detailed explanations of how each test was developed and conducted. From the results presented here future VHDL model builders will be able to develop more efficient models by knowing the effects different model characteristics will have on their simulation execution times.

1. Introduction

As the use of the VHDL Hardware Description Language (VHDL) to describe complex systems grows, the simulation execution speed of VHDL becomes increasingly important. Quick execution of simulations for various modeling alternatives is required for efficient exploration of the design space. Further, as VHDL is used to describe systems at higher levels of abstraction, the use of large complex data structures and bus resolution functions is required. The use of these complex constructs can be at odds with the requirements for efficient simulation execution. This report presents the results of an investigation of the effect of several constructs typically used in high-level VHDL models on simulation execution times.

2. Background

In order to test the effect of various model characteristics on simulator execution time of VHDL models, the University of Virginia's ADEPT (Advanced Design Environment Prototype Tool) [1] environment was used. This performance modeling environment is based upon a building block approach where the basic building blocks are referred to as ADEPT modules. These ADEPT modules can be interconnected to create complex structures which represent systems. The individual behavior of these modules has been described in VHDL. The modules use a token passing mechanism to transfer information between modules. These tokens are composed of an array of integers that can be broken down into two different groups. The first group is the status field; this field is used to control the flow of the token. The status field can assume one of four values: present, acknowledged, released, or removed, and is used to implement a fully interlocked handshaking protocol. This status field is not modifiable by the user. The second group is composed of eighteen integer fields. All of these fields can be edited by the user and contain user-specified data. These eighteen integer fields will be referred to as the tag fields. The use of this token structure means that all signals in an ADEPT simulation are composed of an array of nineteen integers.

3. Simulation Tests

There are four primary characteristics of VHDL models that will be examined in this paper: the size of signals, the use of file input and output (I/O) operations, the use of bus resolution functions, and the overall size and complexity of the model. Numerous tests have been implemented to evaluate the effect of each characteristic on VHDL simulation execution times.

---

1 This work was sponsored by the Advanced Research Projects Agency under Contract No. F33615-93-C1313.
3.1 Reduction of Signal Size

A set of tests has been developed to examine the effect of signal size, that is the number of fields in a token, versus the overall simulation execution time. The ADEPT system, since it incorporates an array of nineteen integers as a token, provides a convenient way to reduce the size of a signal throughout a VHDL model. Many models built using the ADEPT environment do not make full use of all eighteen of the tag fields. This situation allows for the size of a signal within a VHDL simulation to be reduced by simply modifying the size of the ADEPT token. By reducing the number of tag fields within the standard ADEPT token, the effect of signal size versus simulation execution time has been studied.

3.2 File I/O

During the examination of two different simulators, it was observed that these simulators handled file input and output differently. This result led to the investigation of the effect of file operations on VHDL simulation execution time. By comparing the simulation execution time for a system using VHDL models which incorporated file outputs versus the exact same system with the file outputs removed, this effect could be observed. No data on the effect of file input operation on these times was derived since all of the models examined included only file output operations. However, it is felt that the results would be similar.

3.3 Bus Resolution Functions

The next issue examined was the effect of bus resolution functions on simulation times. Bus resolution functions are commonly used within VHDL to allow for the abstraction of different bus protocols as well as different interconnection topologies. Examples of such bus resolution functions include: wired-and, wired-or, and various handshaking protocols. Currently, the ADEPT environment uses a two way, four state fully interlocked handshake on a single signal. This handshake protocol is implemented using a bus resolution function in which certain states have a higher priority over others and can overwrite these states when assigned to the same signal. This implementation allows the ADEPT system to use a signal with one resolved status field to pass tokens between modules.

A two wire handshake system, which does not use bus resolution functions, was developed for a comparison. This type of handshake still allows for a four state fully interlocked handshake protocol, but eliminates the need for a bus resolution function. Simulations of models built with the bus resolution function implementation have been compared against the same models built using the two wire handshake architecture. By replacing this bus resolution function with a two wire scheme, the effect of VHDL bus resolution functions on simulation execution times was examined.

3.4 Size of a Model

Another concern with the growing use of VHDL is how it handles models with a larger number of modules and how these more complex models effect simulation execution times. Numerous users have claimed that as their models become increasingly large their simulation execution times seemed to grow superlinearly. This claim of superlinear growth in execution times prompted investigation into this issue. To determine what effect the size of a model has on the simulation execution time, a special test model was created. This model was composed in such a way that the number of modules can be increased in a measurable fashion. By developing a type of modular model, the relationship between model complexity and the resulting simulation execution times was determined.

3.5 Testing Procedure

To accurately determine the effect of these tests on simulation times, two different simulators were used. The use of two different simulators was needed to determine if the effect of a specific model characteristic on execution time is simulator specific. These tests are for comparison purposes only, therefore, the names of these two simulators will remain anonymous. The simulators will be referred to as simulator A, and simulator B throughout this paper.

To develop these tests each simulator was run in batch mode using the Unix time command. This scheme allowed for each simulation run to be timed during its execution. The tests were all run on one Sun station SPARC 10 fitted with dual processors and 128 megabytes of memory. The tests were run during off peak hours to guarantee near exclusive use of the machine for these trials. Small discrepancies seen in some of the data taken can be attributed to other processes being executed during these tests. Using the Unix time command, three separate times are displayed: real, user, and system. The user times were collected for this experiment, which correspond to the amount of time the specific processes spent in the system. Note that this is not the wall clock time which is referred to as the real time with the "time" command. To ensure a more reliable test, it was decided that each simulation should be run at least three times. Three different lengths of simulation were also chosen resulting in a total of nine simulations for every VHDL model.

Two different ADEPT models were used to test the effect of signal size on simulation times. The first model chosen was the Algorithm to Architecture Mapping Methodology.

1.8
(ATAMM) [2]. This model consists of slightly more than 200 ADEPT library modules connected by a total of 196 signals. The function of this model is to represent a seven node arbitration graph. The ATAMM model essentially generates a token every two nanoseconds at the input which in turn corresponds to a signal being generated at the model input every two nanoseconds. This model was first timed using the standard ADEPT modules which contain eighteen tag fields, and one status field. Since the ATAMM used only two of these tag fields for its execution, the tag field size was modified to contain various sizes of tag fields ranging from two to eighteen. Only the size of the tag field was reduced. Since the majority of the tag fields in this model were unused, the function of the system level model was unchanged.

The second model tested was the model of a stream memory controller (SMC) [3]. This model is composed of approximately 1,500 ADEPT modules. This model used a minimum of 10 tag fields allowing the number of tag fields to be varied from 10 to 18. By using two different models more specific conclusions about the effect of signal size on simulation execution times can be drawn.

To determine whether file input and output have any effect on the VHDL simulator execution times, a second test was developed. This test compared the simulation execution times of the ATAMM, both with and without file I/O. To eliminate file output from this model one of the ADEPT primitives was recoded. This recoding did not affect the model in any way except to eliminate write statements from the VHDL code.

A test to show the effect of the bus resolution function on simulation execution times was developed. The ADEPT modules had to be recoded to support a two wire handshaking protocol rather than using the standard ADEPT bus resolution function. Once the modules were recoded, the ATAMM model was modified to use these new modules. Once this model was created using the two wire handshaking primitives, the number of tag fields was also changed to determine if the size of a VHDL signal affected simulations using the two wire handshaking and the bus resolution function in the same manner.

A test to compare model complexity and size, versus execution time was developed. A modular type of model was developed in order to allow the size and complexity to be increased in a measurable fashion. A model of an N stage linear pipeline was used for this experiment. By using the linear pipeline, the number of stages can be altered allowing the size of a model to be changed without affecting the overall functionality of the design. The number of stages in the pipeline corresponds directly to the total number of VHDL signals within the model. Several models ranging from twenty to four hundred pipeline stages were examined for this test.

The data presented here is only a sample of the actual data that has been collected refer to [4] for the entire data set.

4. Results

4.1 Reduction of Signal Size

Three sets of results were generated for each model. For the ATAMM model, the three simulation lengths chosen were: 1,000, 10,000, and 100,000 ns. A sample of these graphs can be seen in Figure 1. The simulation execution times with respect to the number of tag fields were taken. In addition, the results of each simulator were placed on the same graph to give a comparison between the two. A best fit linear regression line is also drawn in for each data set.

Figure 1 shows the effect of tag field reduction versus simulation execution times for the ATAMM Model with a simulation length of 10,000 nanoseconds. Using this model and input configuration, simulator A exhibited a speedup factor of 1.80 when the tag fields were changed from eighteen down to two, while simulator B yielded a speedup of 2.52. Simulator A, proved to be 2.66 times faster than simulator B with only two tag fields. This ratio, however, steadily expanded to 3.72 for the original token size of eighteen tag fields. Thus simulator A, on average, demonstrated a decrease in simulation execution time of 20.13 seconds per tag field eliminated, while simulator B showed an average decrease of 101.01 seconds per tag field reduced.

![Tag Field Simulation Comparison](image)

Figure 1. Tag Field Simulation Comparison for ATAMM Model
The effect of signal size versus simulation execution time was also examined using the SMC Model. This model was considerably larger than the ATAMM model and contained roughly eight times the amount of VHDL signals. It should be noted that since the SMC utilized a larger portion of the tag fields, the tag field size could only be reduced from eighteen down to ten. The simulation lengths chosen for this model, 1350,000, 270,000, and 540,000 ns correspond to actual simulation times needed to complete different applications on the SMC model. Results of the tag field reduction versus execution times for the SMC model with a 135,000 ns simulation time can be seen in Figure 2.

![Tag Field Simulation Comparison](image)

**Figure 2. Tag Field Simulation Comparison for SMC Model**

In this case simulator A demonstrated a speedup factor of 1.40, while simulator B produced a speedup of 1.65. Simulator A also finished an average 3.36 times faster than simulator B with only ten tag fields. However, this ratio grew to 3.97 for the full eighteen tag fields. These results give simulator A an average decrease in simulation execution time of 84.3 seconds per tag field eliminated, while simulator B has an average decrease of 464.9 per tag field removed. For all of this data it should be noted that the reduction in tag fields is a direct correlation to the reduction of signal size in a VHDL simulation. Each tag field that is eliminated corresponds to the removal of one integer element of an array within each signal.

Several conclusions can be made after examining the results of the signal size versus simulation lengths studies.

The first point is that in all of these graphs the relationship between signal size and simulation execution times is linear. Therefore, no matter which simulator is chosen, the average simulation execution time can be significantly decreased by reducing the size of the token's tag field (the signal). For the ATAMM model, simulator A, on average, resulted in a factor of 1.83 in speedup when the standard eighteen integer tag field token was reduced to the minimal two integer tag field token. The speedup results for simulator B were even better than those of A, resulting in an average overall speedup of 2.53. However, even though simulator B resulted in higher speedup factor than that of A, its performance was still considerably slower. As shown by the graphs, simulator B was at least 2.67 times slower than that of A. Unfortunately, this ratio grew even larger as the size of the signal increased. Therefore, one must be very careful in choosing the simulator that is used.

Analogous results for the SMC were also seen. An average speedup factor for simulator A of 1.43 was obtained when reducing the full eighteen integer signal size down to ten. Simulator B produced an average speedup factor of 1.63 when comparing its reduced signal size to the standard ADEPT signal size. It must be noted that the difference between the SMC and ATAMM results are due to the fact that the SMC model used a larger number of the available tag fields thus restricting the amount by which the tag fields may be reduced. However, the overall results between the two different models are very similar. As was the case with the ATAMM model, the ratio of simulator B to simulator A grew as the size of the signal was increased for the SMC model. The overall conclusion is that not only is simulator A on average faster than B, but it is also able to handle larger signal sizes more efficiently.

### 4.2 File I/O Comparison

To perform this comparison the ATAMM model was again run for three different simulation lengths without file I/O included. The average of three execution times for simulations run at each of these lengths was taken. The number of tag fields was also varied for these tests in order to show the existence of any correlation between signal size and file input and output. Sample results are shown in Figures 3 and 4. The result of this experiment was that the addition of file output did not seem to have a significant effect on the simulation execution times. However, it is also apparent from the approximately parallel lines on these figures that there is an associated fixed overhead involved with file output.

The results show that by removing the file outputs, which correspond to 64.64 Megabytes of output data generated by invoking the VHDL write command over 400,000 individual times, a small constant amount of speedup was
gained in the simulation execution times. The amount of speedup depends on the specific model and the amount of file I/O operations it performs, and the total length of the simulation being tested. Again, the amount of speedup is simulator dependent.

As shown in Figure 5, the average speedup obtained on simulator A using a two wire handshaking protocol was 2.7, while for simulator B, the average speedup was 6.5.

4.4 Model Complexity

By creating a model of an N stage linear pipeline within the ADEPT environment the effect of simulator execution time versus model complexity could be observed. Using the ADEPT environment allows for the creation of this linear pipeline which is a series of buffered delay elements. The modular framework of the ADEPT environment also allows for the number of stages in the pipeline to be altered rather easily. Each stage in the pipeline example corresponds to a 1 ns fixed delay module followed by a buffer module. Therefore, to construct a twenty stage linear pipeline twenty of these fixed delay-buffer pairs were strung together.
A slope of 53.26 with a standard error of 0.21 resulted for simulator B. When compared to the slope for Figure 6 (the corresponding result for simulator A), the differences between the two simulators can again be noted.

5. Conclusions

This paper presented the results of tests that were conducted to determine the effect of VHDL system level model characteristics on simulation execution times. Significant insight into what can be done within the model to decrease the total time for simulation of complex VHDL models has been gained. For the models tested the size of the signal has a very linear effect on execution times. Even though most of this signal was not being used within the models, it still presented significant overhead to the simulator. This overhead results from the simulator passing around the full signal of an array of integers. By reducing the size of the signal by a factor of 1.8-6.33, the average speedup factor of 1.43-1.83 or 1.63-2.53 was obtained depending on which simulator was used.

The file I/O experiment showed that by removing file writes within a design, the user can reduce, for a given model and simulation length, the execution time of a simulation by a constant factor. Although this constant might not be significant for small designs or short simulation times, effects may be significant for larger and more complex models.

The use of bus resolution functions was shown to have a significant effect on simulation time. This single factor resulted in decreased simulation execution times by a factor of 2.77 or 6.43 depending upon which simulator was used. The results also showed that by replacing the bus resolution function, the execution times of the two simulators were comparable. This one experiment illustrated the significant differences in various simulators.

It has been conjectured that as the size and complexity of models increase, the simulator execution times grow superlinearly. It was shown through these experiments that the number of VHDL signals has a very linear effect on the simulation execution time. However, the slopes of these lines between the two simulators were significantly different. For example, these results show that simulator A can handle VHDL models with an increased number of signals more efficiently than simulator B.

The data taken to date has given considerable insight into what can be done to speed up these simulation execution times. Several different aspects of VHDL code were examined and the results have shown that careful planning of VHDL code can greatly reduce simulation execution times. Not only do these results show different methods by which to reduce execution times but they also show that the
choice of simulators plays an important part in the simulation execution time and the effect of these methods on the execution time. This more detailed understanding of the simulation environment allows future VHDL users to build and simulate models in a more efficient manner.

Acknowledgments

A number of CSIS students and staff have contributed to the development of the ADEPT environment and to the material presented in this paper. These students include Bob McGraw, Ramesh Rao, Gnanasekaran Swaminathan, Charles Choi, Maximo Salinas, and Moshe Meyassed.

References


