

Performance Modeling of High Performance Distributed Computing Systems Using VHDL-Based Methodology

Dr. Mohammad A. Mikki*

ملخص البحث

إن تصميم نموذج للأنظمة التوزيعية ذات الأداء العالي (High Performance Distributed Computing Systems) وكذلك تحليل أدائها هو مشكلة ذات تحد كبير من الناحية البحثية. هدفنا في هذه الورقة هو تطوير و تصميم نموذج مبني على لغة VHDL (لغة توصيف المكونات المادية للدوائر الرقمية المتكاملة الكبيرة جدا) و هي لغة تستخدم في مجال التصميم الرقمي. حاليا لا توجد طريقة شمولية متكاملة تقوم باستخدام لغة VHDL في مجال تحليل و نمذجة أداء الأنظمة التوزيعية ذات الأداء العالي. إن طريقتنا تقوم بتوسيع إمكانيات لغة VHDL من أجل طرح طريقة منهجية تقوم بتصميم نظام نمذجة هيكلي يقوم بتزويد ثلاثة أنواع من التحليلات: تحليل على مستوى الشبكة، تحليل على مستوى البروتوكولات، و تحليل على مستوى التطبيقات. كل نوع من انواع التحليل هذه يتم عمل نموذج تصميمي له باستخدام المكونات الاساسية للغة VHDL لقد قمنا باستخدام لغة VHDL كلفة نمذجة اداء و تحليل الأنظمة التوزيعية ذات الأداء العالي للاعتبارات التالية: انها تدعم تمثيل العمليات المتوازية، انها تدعم التمثيل الهيكلي من الاعلى الى الاسفل على المستوى البنائي و كذلك التصرفي. اضافة الى ذلك فان التمثيل التنفيذي للغة VHDL يمكن استخدامه للتحقق من الاداء الوظيفي للنموذج و كذلك خصائصه الزمنية. لقد قمنا في هذا البحث بتصميم تجارب محاكاة للتحقق من صحة النموذج الذي قمنا بتصميمه و مقارنته بالنتائج التي تم الحصول عليها باستخدام برنامج المحاكاة CSIM. ان نتائج المقارنة قد اثبتت أن طريقتنا دقيقة مقارنة بنتائج المحاكاة)

* Assistant Professor in the Electrical and Computer Engineering department at the Islamic University of Gaza

ضمن مدى 3% من الدقة). أن طريقة النمذجة التي قمنا باستخدامها أيضا تظهر أن استخدام لغة VHDL في نمذجة الأنظمة التوزيعية ذات الأداء العالي هي طريقة فعالة و دقيقة.

ABSTRACT

Performance modeling and analysis of High Performance Distributed Computing (HPDC) systems are challenging research tasks. Our objective is to develop a VHDL-based performance modeling and analysis of HPDC systems. Currently, there is no comprehensive method to use VHDL in performance modeling and analysis. Our approach extends the capability of VHDL to achieve a hierarchical modeling approach that provides three types of analysis: Network Analysis, Protocol Analysis, and Application Analysis. Each analysis type is modeled and carried out using VHDL primitives. We choose VHDL as the performance modeling language because it supports concurrency, top-down hierarchical modeling and structural and behavioral modeling. Furthermore, its executable representation can be used to verify functionality and timing characteristics. We use simulation experiments to validate our modeling approach and compare the results with those obtained using CSIM simulator. The comparison results show that our approach is accurate compared to the simulation results (within 3% of accuracy). Our modeling approach also shows that using VHDL in modeling HPDC systems is powerful and accurate.

1. INTRODUCTION

The evolution of heterogeneous computing systems interconnected by high-speed communication networks has led to an increased interest in studying HPDC systems and their applications. There has been a special interest in modeling, analyzing and designing such networked systems. Most of the current research in this area focus on performance issues related to the lower layers of the communications protocol stack. Very few address the end-to-end performance analysis that takes into consideration higher layers and their impact on the performance. This research is intended to fix this problem and to develop a unified approach to design and analyze networked systems and their applications.

The three general approaches used in modeling and analysis are measurement, simulation and analytical techniques. Analytical techniques use mathematical models to represent the functions of different layers of a system. Modeling includes queuing network models, Markov chains, and stochastic Petri Nets. There are several existing network modeling and simulation tools such as QNA from Bell labs [Whitt 1983], Netmod from the University of Michigan [Bachmann et al. 1991], Netmodeler from IBM [Ridos et. al], OPNET from MIL3 Inc., COMNET from CACI Inc., and a few others. Most of the tools do not address the end-to-end performance analysis that takes into account the effect of application, protocol and network on the performance. We present a unified approach that can be used to design as well as model HPDC systems and their applications. The use of VHDL in digital system design is well established. However, its use in performance modeling and analysis is a new research area and is not yet fully explored. Our modeling approach uses VHDL to model the performance of such systems and their

applications. The approach is based on a three-layer hierarchy that analyzes the end-to-end performance by partitioning the system into network layer, protocol layer and application layer. The functions and operations of each layer are modeled using VHDL. This approach enables the designer to study system performance for different types of networks and protocols and different strategies.

The rest of the paper is organized as follows: Section 2 presents a brief overview of related work to using VHDL in modeling and analysis. Section 3 presents our VHDL-based modeling approach. Section 4 presents an illustrative example on using VHDL to model an ATM switch. Section 5 presents some simulation and experimental results. And finally, Section 6 concludes the paper.

2. RELATED WORK

Performance modeling and analysis of networked systems have been conducted extensively. Queuing models are extensively used in the literature. Most of the existing analytical techniques are based on queuing theory [Hariri et al. 1999; Eliazov et al. 1990; Fong and Singh 1998]. A well-known approach to analyze large queuing networks is to decompose the network into individual queues and analyze each queue in isolation [Onvural 1995]. [Hariri et al. 1999] developed a hierarchical queuing networks modeling approach for analyzing the end-to-end performance of applications running on HPDC systems. The approach decomposes the overall system into application level, protocol level and network level. Functions at each level are modeled using queuing networks. Most of the queuing models fail to accurately model the dynamic behavior of the system operations [Eliazov et al. 1990].

Recently, some researchers used VHDL in modeling networked systems. In the following, we present a brief overview of VHDL modeling techniques. [Eliazov et al. 1990] presented a parametric model for the aggregate cell streams of the ATM input traffic. They built an event-driven simulator written in C++ and used queuing analysis to determine the cell loss probability, average cell delay and steady-state queue length distribution. [McHenry and Midkiff 1994] used VHDL to model the performance of multiprocessor networks. They used supportive programs to supply statistical workloads and derive performance results. [Sriram and Kurdahi 1996] developed a test-bench in which the switch generated from the SpecCharts description was instantiated. A VHDL behavioral model was created for an ATM traffic source. Test cases were developed to test the unicast, multicast and broadcast transmission of ATM cells. [Baldi et al. 1996] used VHDL to analyze and model communication protocols in IEEE 802.3. [Marchese et al. 1996] presented an ATM node simulation model using VHDL. They build an experimental bench to validate theoretical models of congestion control in ATM through simulation. [Smith and Morris 1995] presented a VHDL-based methodology for modeling computer communication systems. The methodology is based on the concept of a hierarchy of modules. The LAN hardware structures can be either processor entity, channel entity or memory entity. In addition, they include a bridge entity that acts as a channel translator.

3. VHDL-BASED MODELING AND ANALYSIS

VHDL is the newest standard language to address the rapidly growing complexity and sophistication of digital system design. VHDL is rapidly gaining acceptance and is influencing

advancements in design methodologies and design automation technology [Dewey 1997]. It plays an increasing part in digital systems design and modeling. VHDL is currently being used as a common modeling language [Amadori and Goerezza 1992; Smith and Morris 1995]. Once written, a VHDL model can be executed by a software program called a simulator. A simulator runs a VHDL description and computes the outputs of the modeled digital system in response to a series of inputs applied over time [Dewey 1997]. We chose VHDL for performance modeling and analysis because it has the following important features [Dewey 1997; McHenry and Midkiff 1994; Sriram and Kurdahi 1996; Smith and Morris 1995]:

1. It provides a standard, portable and flexible design representation for complex digital hardware.
2. It provides support of concurrency, program constructs and behavioral completion.
3. It encompasses both structural and behavioral models.
4. It is an executable representation in that a simulator can be used to verify functionality and timing specifications.
5. It enables models to be refined as design details become available.
6. It supports top-down hierarchical modeling that supports accurate modeling and simplifies the design process.
7. VHDL models may be developed as part of the design process for synthesis and verification, so performance modeling activities can be integrated with the design process by using the same models.

8. It is an abstract modeling language that could describe the temporal behavior and structure of a system from the overall block diagram level down to the gate level.
9. It can model conceptual distributed computing systems as well as physical ones.
10. Finally, VHDL models are technology-dependent, i.e., they imply a particular logic family. For example, technology-dependent information may be required to improve modeling accuracy so that a VHDL model will more closely reflect the behavior of the actual hardware.

Figure 1 shows the main components of our VHDL-based modeling and analysis approach that consist of the following modules: Traffic Module, Controller Module and System (component) Datapath Module. The Traffic Module can model different types of traffic models (e.g., Poisson, Bernoulli). The Datapath Module models the main functions in any system or component (e.g., computer, router, switch). The Controller Module defines how these functional units interact with each other during the system operations (normal or abnormal).

Each system (component) is divided into two parts: a controller and a datapath. The datapath manipulates data according to the commands from the controller. The controller is modeled as a finite state machine (FSM). We chose finite state machines because they are commonly implemented in programmable logic devices [Dewey 1997; Hsu et al. 1995; Skahill 1996]. Writing a behavioral state machine description in VHDL is simply a matter of translating a state flow diagram to *case-when* and/or *if-then-else* VHDL statements. Furthermore, the controller module can also model the communication protocols. Once the VHDL models that represent the traffic, the controller and the datapath associated

with the system (component) are modeled, the next step is to invoke the VHDL simulator and obtain the performance metrics as specified in the VHDL models. In the next section, we will discuss in further detail how to use our VHDL-based modeling to analyze the performance metrics (cell delay, cell loss probability and average queue length) of an ATM switch.

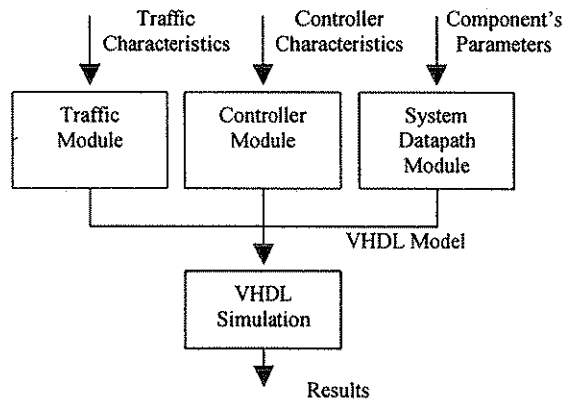


Figure 1: Procedure of the VHDL-based modeling approach

4. VHDL-BASED MODELING OF AN ATM SWITCH

Figure 2 shows a VHDL model of the $N \times N$ shared-memory non-blocking ATM switch. The ATM switch is divided into a datapath and a controller. The datapath of the ATM switch includes the Serial to Parallel (S/P), Header Converter (HD CNV), Multiplexer (MUX), Buffer, Demultiplexer (DMUX), and Parallel to Serial (P/S) functional units. The MUX component multiplexes the incoming cells from different input ports into a single stream of data. The delay of the MUX is a function of the cell size and number of inputs. The buffer is used for storing the outgoing cells temporarily. The buffer write rate is modeled according to the characteristics of the application. The DMUX component demultiplexes the stream

of cells. And finally, the P/S component performs parallel to serial data conversion.

The controller models the protocol layer as a FSM which is modeled as a VHDL *process* statement. The protocol is initially in the *idle* state waiting for cells to arrive at the input port. When a cell arrives, it sets the *cell_arrived* signal and the protocol goes to the *receive_cell* state. In the *receive_cell* state the controller handles the arriving cell by accumulating the cell header and data. In the *process_cell* the protocol routes the cell and translates cell address in parallel. It routes the cell by determining the output port to which the incoming cell is to be routed. The protocol translates the address by determining the new VPI and VCI addresses and creating new cell header fields. In *switch_cell* state the protocol switches the cell by storing it in an internal buffer in the switch where it is queued waiting to be transmitted. If the buffer is full the cell will be lost. In the *transmit_cell* state, the protocol transmits the cell to the output channel. In the next section, we discuss the VHDL simulation results and compare them with simulation results obtained using CSIM.

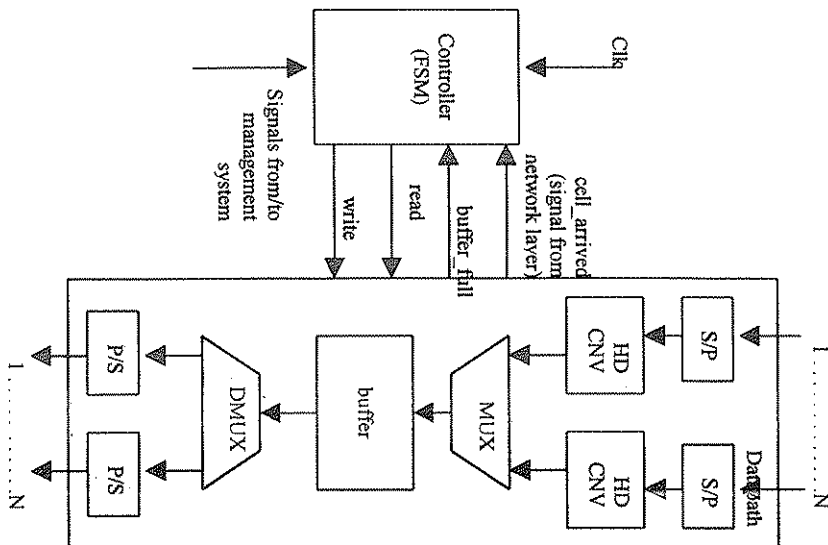


Figure 2: Block diagram of the ATM switch model

5. EXPERIMENTAL RESULTS

In this section we present preliminary results to validate the proposed VHDL model. We use CSIM to simulate the shared memory non-blocking ATM switch shown in Figure 2. In our analysis and simulations, we assume 8x8 ATM switch. We focus our attention on a particular “tagged” output buffer and concentrate only on the traffic destined for that output buffer. We model the ATM switch arrival traffic as bursty and the cell arrival stream as correlated. There are different models of traffic burstiness that were developed in the literature. [Feng et al. 1996] overviews some of these models. Most of these models include ON/OFF bursty model for VBR traffic and bursty traffic with long-range dependence which is known as self-similar traffic model. We use the bursty traffic model used in [Eliazov et al. 1990] to analyze the delay, cell loss probability and average queue length in the buffer (see Figure 3).

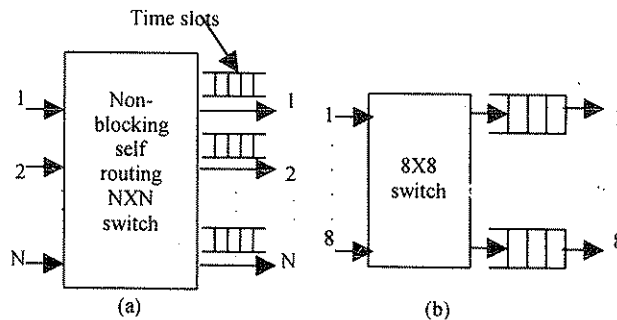


Figure 3: (a) Model of shared-memory NXN ATM switch (b) Simulation experiments ATM switch configuration

The simulation was done at the cell level, assuming fixed-length cells transmitted in equal-length time slots, where a slot is the unit of time necessary for transmitting one cell. The output buffer speed is eight times higher than the speed of the input and output ports so that regardless of the number of contending input ports for an output buffer, all cells can be switched in one time slot to that buffer. We model all the traffic coming from different input ports to the tagged output buffer as one aggregate cell stream and thus avoid the dimensionality problem. The incoming cell traffic is a bursty process. Each input port alternates between silent periods (S) and active periods (A) from the perspective of the tagged output line with the following parameters: $m(A)$, $m(S)$, $c^2(A)$, $c^2(S)$, $k(A)$ and ρ where:

$m(A)$ is the mean of the active period

$m(S)$ is the mean of the silent period

$c^2(a)$ is the squared coefficient of variation of active period

$c^2(S)$ is the squared coefficient of variation of silent period

$k(A)$ is constant cell inter-arrival time during active periods.

ρ is the total load.

$k(A)$ is assumed constant for the sake of incorporating periodicity which is inherent in most systems.

The load destined from one input port to the tagged output port is given by

$$\lambda = \left[\frac{m(A)}{m(A) + m(S)} \right] \frac{1}{k(A)}$$

We assume that the lengths of the active and silent periods to be independent and identically distributed random variables and their distributions are mixtures of geometric distributions. The procedure to obtain their distributions is given below.

The distribution function of a mixture of two geometric distributions with parameters p_1 and p_2 respectively and mixing ratios of α and $1 - \alpha$ is given by

$$p(n) = \alpha(1 - p_1)p_1^{n-1} + (1 - \alpha)(1 - p_2)p_2^{n-1}$$

where

$$n \geq 1, p_1 \quad \text{and} \quad p_2 > 0$$

$$\alpha = 0.5 \left[1 + \sqrt{\frac{(c^2 - 1)m + 1}{(c^2 + 1)m + 1}} \right]$$

$$p_1 = (m - 2\alpha) / m$$

$$p_2 = [m - 2(1 - \alpha)] / m$$

Test-bench was created to simulate the model by instantiating the traffic model and ATM switch.

The results presented in this section were obtained using the VHDL model presented in the previous section. The model was written in VHDL 93. The VHDL simulator used is VeriBest simulator under the environment of MS Windows 98.

The first experiment models the effect of the buffer size on the cell loss probability. The cell loss probability is calculated as the ratio of the number of lost cells due to the full buffer to the total number of arriving cells. Counters of these quantities are embedded in the code. The results of the simulation experiment are shown in Figure 4. The graph shows that the VHDL model has a higher cell loss probability than the CSIM

model. This is due to the fact that the VHDL model takes into consideration the other components of the ATM switch which have their failure rate. The CSIM model considers only the cell loss due to buffer overflow. Our model reflects the actual cell loss probability more accurately than CSIM model. For cell arrival processes with small values of $c^2(A)$, the cell loss probability decreases significantly as the buffer size increases while in case of large $c^2(A)$ this decrease is not that significant. Thus, increasing the buffer size to reduce cell loss probability is not an attractive solution in case of large $c^2(A)$. Also, for a fixed buffer size the cell loss probability increases as $c^2(A)$ increases. Our results agree with the results obtained in [Eliazov et al. 1990] and other previously reported results.

The second experiment models the effect of the offered load of cell arrival process with three different values of the squared coefficient of variation $c^2(A)$ on the average cell delay. The results of the simulation experiment are shown in Figure 5. As shown in the figure, the cell delay calculated by the VHDL model is always greater than the cell delay calculated by the CSIM model. This is due to including in the VHDL model the delay of components of the ATM switch other than the buffer delay and is due also to including the protocol layer in the model. Thus, the VHDL model reflects the actual cell delay in the ATM switch more accurately than the CSIM model does. The figure also shows that the average cell delay increases as the load increases which is as expected. Also, for a fixed load, the average cell delay increases as $c^2(A)$ increases which is in agreement of previously reported results and our expectations.

The final experiment models the effect of the offered load of cell arrival process on the queue length. The overall load was varied by varying the average length of the cycle by changing

$m(S)$. The queue length is calculated as the average queue length over time after initialization. The results of the simulation experiment are shown in Figure 6. As shown in the figure, the queue length calculated by the VHDL model is always less than the queue length calculated by the CSIM model. This is due to the loss of some cells in the VHDL model in the previous ATM switch components due to their failure before they arrive to the buffer. Thus, the VHDL model reflects the actual queue length in the ATM switch more accurately than the CSIM model does. The figure also shows that the average queue length increases as the load increases which is in agreement of previously reported results and our expectations.

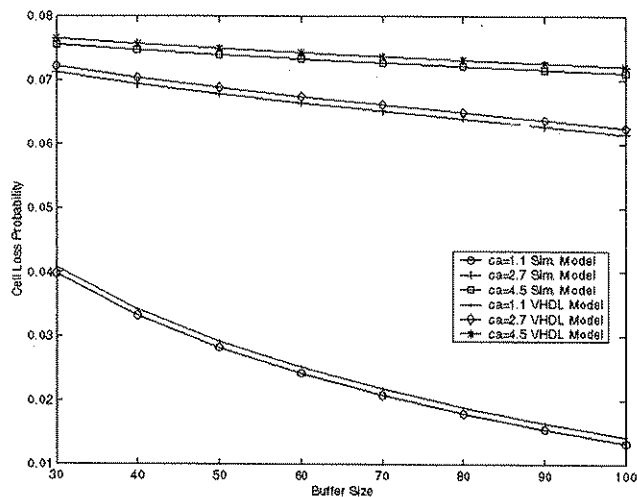


Figure 4: $m(A) = 25$, $m(S) = 37$, $c(A) = c(S)$, $k(A) = 4$, $\rho = 0.8$

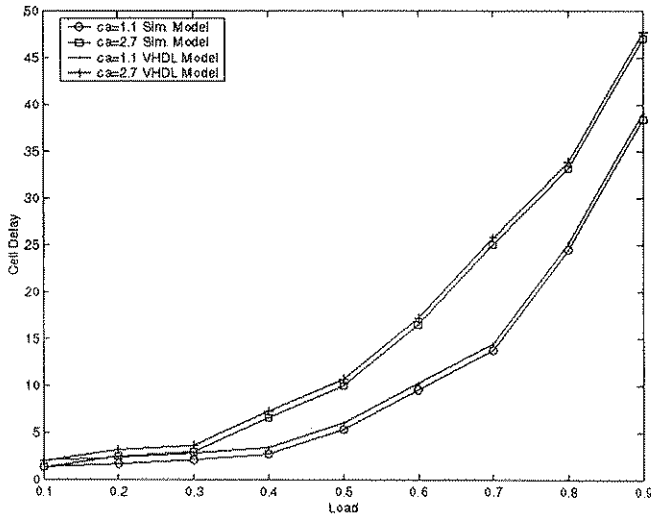


Figure 5: $m(A) = 25$, $c(A) = c(S)$, $k = 4$, buffer size = 100

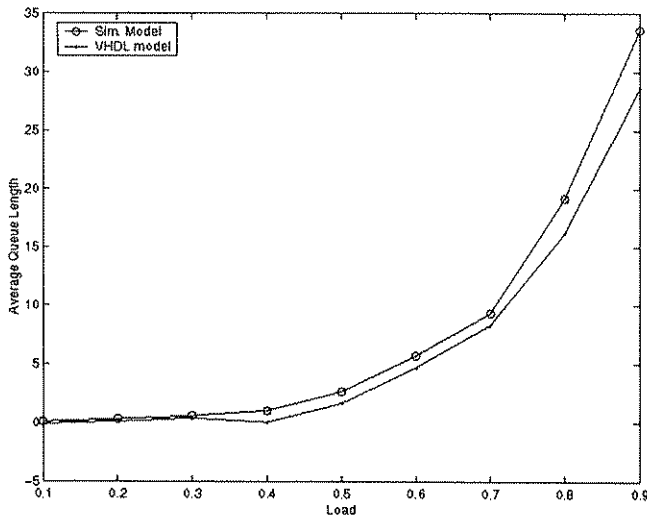


Figure 6: $m(A) = 25$, $k = 4$, $c(A) = c(S) = 1.1$, buffer size = 100

6. CONCLUSION

We presented a VHDL-based performance modeling and analysis approach that can be used to model systems or networked systems. The approach is hierarchical and it models the network, protocol and application behaviors. We used our VHDL modeling approach to analyze the performance of an ATM switch. We have also used CSIM simulator to compare and validate our model results. The performance results obtained from our model show that our model is accurate compared to those obtained using CSIM (within 3 % of accuracy). The VHDL model results also show that they are in agreement with previously reported results. Our modeling approach also shows that using VHDL in modeling and analysis is accurate; this is true because we are using the same VHDL models used to build the systems or components in our modeling and analysis.

REFERENCES

- [Amadori and Coerezza 1992] Amadori, S. and P. Coerezza. 1992. "Design of Complex Systems with a VHDL Based Methodology." In Proceedings of the *Design Automation Conference, EURO-VHDL '92, EURO-DAC '92* (Sept. 7-10). 658-663.
- [Bachmann et al. 1991] Bachmann, D.; M. Sigal; M. Srinivasan; and T. Teorey. 1991. "Netmod: A Design Tool for Large Scale Heterogeneous Campus Networks." *IEEE Journal on Selected Areas in Communications*, Vol. 9, No. 1, (Jan.).
- [Baldi et al. 1996] Baldi, M; E. Macii; and M. Poncino. 1996. "Efficient Analysis of Communication Protocols Using VHDL

Modeling and Simulation.” In Proceedings of the *2nd International Conference on ASIC* (Oct. 21-24). 428-431.

[Daveau et al. 1998] Daveau, J.; G. Marchioro; and A. Jerraya. 1998. “Hardware/software Co-design of an ATM Network Interface Card: A Case Study.” In Proceedings of the *sixth international workshop on hardware/software Codesign, CODES/CASHE' 98* (March 15-18). 111-115.

[Dewey 1997] Dewey, A. M. 1997. *Analysis and Design of Digital Systems*. International Thompson Publishing Inc.

[Eliazov et al. 1990] Eliazov, T.; V. Ramaswami; W. Willinger; and G. Latouche. 1990. “Performance of an ATM Switch: Simulation Study.” In Proceedings of the *Ninth Annual Joint Conference of the IEEE Computer and Communication Societies, INFOCOM 90* (June 3-7). 644-659.

[Feng et al. 1996] Feng, F; C. Li; A. Raha; S. Yu; and W. Zhao. 1996. “Modeling and Regulation of Host Traffic in ATM Networks.” In Proceedings of the *21st IEEE Conference on Local Area Networks* (Oct. 13-16). 458-467.

[Fong and Singh 1998] Fong, S. and S. Singh. 1998. “Performance Analysis of Shared Buffer ATM Switch with Different Cell Departure Models.” In Proceedings of the *1998 IEEE International Performance, Computing and Communications Conference IPCCC '98* (Feb. 16-18). 386 – 392.

[Hariri et al. 1999] Hariri, S.; P. Varshney; L. Zhou; H. Xu; and S. Ghaya. 1999. “A Hierarchical Analysis Approach for High Performance Computing and Communication Applications.” In Proceedings of the *32nd Annual Hawaii*

International Conference on Systems Sciences, HICSS-32 (Jan. 5-8). 10pp.

[Hsu et al. 1995] Hsu, Y.; Tsai K.; Liu J.; and Lin E. 1995. *VHDL Modeling for Digital Design Synthesis*. Kluwer Academic Publishers

[Marchese et al. 1996] Marchese, M.; F. Curatelli; M. Chirico; and L. Mangeruca. 1996. "Implementation Issues for Congestion Control in ATM Networks." In Proceedings of the *Third IEEE International Conference on Electronics, Circuits, and Systems, ICECS '96* Vol. 2 (Oct. 13-16). 788-791.

[McHenry and Midkiff 1994] McHenry, J. and S. Midkiff. 1994. "VHDL Modeling for Performance Evaluation of Multicomputer Networks." In Proceedings of the *Second International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, MASCOTS '94* (Jan. 31-Feb. 2). 174-178.

[Onvural 1995] Onvural, R. O. 1995. *Asynchronous Transfer Mode Networks Performance Issues*. Second Edition. Artech House, Inc.

[Skahill 1996] Skahill, K. 1996. *VHDL for programmable logic*. Addison-Wesley Publishing Inc

[Smith 1995] Smith, D. and S. Morris. 1995. "VHDL-based Methodology for Modelling Computer Communication Systems." In Proceedings of the *IEEE Canadian Conference on Electrical and Computer Engineering*, Vol. 2 (Sept. 5-8). 627-631.

[Sriram and Kurdahi 1996] Sriram, A. and F. J. Kurdahi. 1996. "Behavioral Modeling of an ATM Switch Using SpecCharts."

In Proceedings of the *Ninth International Conference on VLSI Design* (Jan. 3-6) 19-22.

[Whitt 1983] Whitt, W. 1983. "The Queuing Network Analyzer." *Bell System Technical Journal*, Vol. 62, No.9, (Nov.)277