Paper title: HW/SW Codesign of Protocols Based on Performance Optimization Using Genetic Algorithms

Authors:	Marcio Nunes de Miranda	E-mail: `marcio@gta.ufrj.br`
	Ricardo N. B. Lima	E-mail: `belem@gta.ufrj.br`
	Aloysio C. P. Pedroza	E-mail: `aloysio@gta.ufrj.br`
	Antônio C. de Mesquita	E-mail: `mesquita@coe.ufrj.br`

Affiliation: Universidade Federal do Rio de Janeiro

Address: Grupo de Teleinformática e Automação – GTA
COPPE - Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
P. O. Box 68504 - 21945-970 - Rio de Janeiro - RJ - Brasil

Phone number: +55 (21) 260-5010 ext. 240

Fax number: +55 (21) 290-6626

Key Topics: Performance Analysis
Quality of Service

HW/SW Codesign of Protocols Based on Performance Optimization Using Genetic Algorithms

Marcio Nunes de Miranda marcio@gta.ufrj.br
Ricardo N. B. LIma belem@gta.ufrj.br
Aloysio C. P. Pedroza aloysio@gta.ufrj.br
Antônio C. de Mesquita mesquita@lpc.ufrj.br

Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro
COPPE/EE - Programa de Engenharia Elétrica/Departamento de Eletrônica
Caixa Postal 68504 - CEP 21945-970 - Rio de Janeiro - RJ - Brasil
FAX: + 55 21 2906626

Abstract

A design-to-hardware methodology applied to the synthesis of communication protocols based on performance analysis techniques and genetic algorithms is presented. The process requires a choice between the tasks to be implemented in hardware and those to be implemented in software, i.e., the selection of the best HW/SW partition. The proposed methodology uses a genetic algorithm to optimize an error function defined by the required performance and the implementation cost of the protocol. The protocol performance is analyzed by a modelling environment called Tangram-II. The cost of the hardware corresponding to a given implementation is calculated by the Synopsys and ALTERA tools. The methodology is intended as a tool to help protocol designers to select the best performance/cost compromise.

1 Introduction

High-speed networks require specific communication protocols. The use of software optimization in the design of these protocols for the current high-performance networks not always allows high-speed operation. An important factor to improve the performance of high-speed protocols is the integration between software (SW) and hardware (HW).

In general, tasks that do not have timing constraints can be implemented in SW while those ones that require a higher performance must be implemented in HW. Due to the increasing demand for protocols with high throughput, solutions using HW are being more frequently investigated. For example, full ASIC-based forwarding engines deliver more than 10-times higher throughput than routers that rely entirely on microprocessors [1].

The basic principle of *Codesign* is a cooperative design based on two specific project environments, HW and SW, where verification, test and simulation can be performed at any design level. During the specification refinement it is useful and necessary to apply techniques that aid the designer to decide between the tasks to be implemented in SW and/or HW. These tasks represent the behavior of the protocol and can be described by a state diagram, where each state transition is a set of clauses and expressions to be evaluated.

It should be noted that the higher cost of HW implementation requires a compromise between the desired performance and the cost of the implementation, when deciding for a given HW/SW partition. This *partitioning process* has a direct impact on the performance of the communication system where the protocol is inserted.

Generally, protocol designers are not concerned with performance issues *during* the design process and relies on his own experience and informal techniques to define a HW/SW partition, which limits considerably the extent of the solutions space search that can be performed. In this case it's useful to provide the designer with performance evaluation tools, in order to derive quantitative measures of the effectiveness of a given partition.

The decision to adopt a given HW/SW partition is always taken by the designer. Although the efforts to automate this process, he still has few resources to aid him in the specification refinement, development and partitioning of the protocol [2][3].

Several projects currently in progress are trying to integrate both HW and SW into the same design process: COSMOS [4], SpecSyn [5], Ptolomey [6], LYCOS [7], Chinook [8] e PISH [9]. Fischer [3] proposes an integrated HW/SW design to meet a high performance in distributed systems, like multimedia systems. It's presented an environment for development and support of the design and implementation stages. Hidalgo [10] proposes a methodology for HW/SW partitioning based on genetic algorithms (GA's) [11]. The choice between HW and SW functional blocks are based on the value of an objective function using tabulated performance values for HW and SW. The performance parameters are not calculated during the search of the best partition, unlike it's done in this work.

In this work a methodology to help in the HW/SW partitioning of a protocol, based on performance requirements, is presented. The proposed codesign technique relies on the use of a performance analysis tool [12][13], the Synopsys [14] and ALTERA [15] synthesis tools and a Genetic Algorithm as optimization method.

The protocols are specified by a state machine where each transition represents a task or a set of tasks to be performed. This specification is used to build up the Tangram-II model and the Synopsys and ALTERA descriptions. Tangram-II aids the designer to find the protocol critical operations in terms of delay, or other parameters like throughput and loss probability. A GA is used to optimize an objective function composed of these parameters. Synopsys and ALTERA tools are used to obtain delay and area measures from synthesized circuits.

Section 2 presents a detailed description of the main techniques used in the proposed methodology which is described in section 3. In section 4 the results of the application of the methodology to a design example are presented. Final remarks are drawn on Section 5.

2 Design Techniques

The proposed methodology involves the use of many techniques in a unique process. The process begins with a functional representation of the system. The partitioning determines which functions will be implemented in HW and which ones will be implemented in SW. Clustering, iterative-improvement and GA's are some examples of algorithms used in the partitioning

process.

The modelling and performance analysis of the protocol are done using the Tangram-II tool, which is a general purpose modelling tool to specify and analyse the performance of computer and communication systems. The main features of this tool are: graphical interface based on the object oriented paradigm and a variety of solution methods to obtain the measures related to the model [12].

GA's are used as optimization tool to minimize an objective function of the performance parameters. GA's have been widely used due to the simplification that it introduces in the formulation and solution of optimization problems. This feature is particularly useful in complex optimization problems, involving a large number of variables and, consequently, solution spaces of high dimensions. Moreover, in many cases where other strategies fail in finding a solution, genetic algorithms converge.

The process of solution adopted in the GA's consists in generating, through specific rules, a large number of individuals, called *population*, in order to cover as much as possible the solution space. After the initialization, each iteration of the GA corresponds to the application of a set of four basic operations: fitness calculation, selection, cross-over and mutation. At the end of these operations, a new population is generated, which is expected to represent a better approximation to the solution of the optimization problem.

The HW synthesis is performed using the VHDL language [16]. The behavioral description of the protocol in VHDL is used as input to the Synopsys and ALTERA tools, that give as output a logic circuit from which the delay and area measurements can be obtained.

3 HW/SW Partitioning of Protocols

Figure 1 illustrates the proposed methodology. Initially, the designer specifies the desired performance and cost values. In the sequence, the protocol is described by a state machine, where each state transition represents a task or a set of tasks to be performed. At this stage, the design process follows two parallel branches. In the first one the parameters that meet the performance requirements are determined by Tangram-II tool and the GA. In the second

branch the delay and area measures are obtained by Synopsys or ALTERA tools.

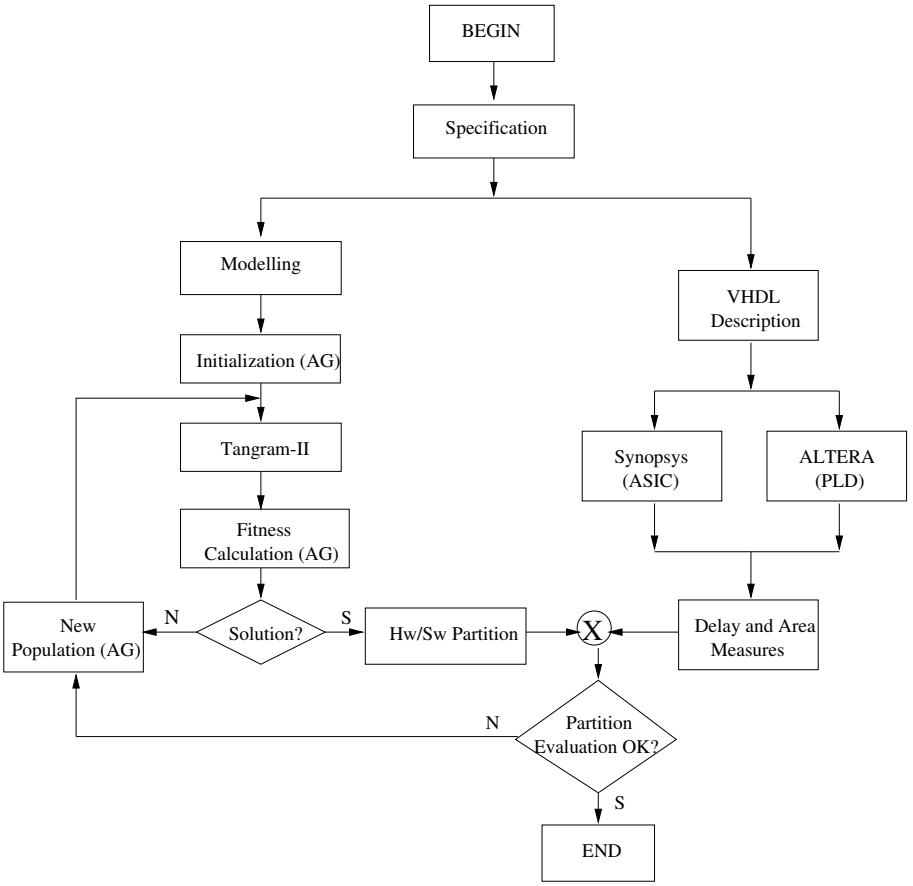

Figure 1: The design methodology.

3.1 Performance Optimization

The protocol specified by a state machine is used to build up the model in the syntax of the Tangram-II. This is a parametric model, where the parameters are the exponential rates of events that describe the behavior of the protocol. The problem is to find the state transition rates that meet the performance requirements. These rates help the designer to establish a partition criterion.

The numerical values needed to solve the Tangram-II model are given by the GA. Tangram-II uses these numerical values to calculate the steady state probabilities of a Markov Chain [17] associated to the model. These probabilities are used to estimate the protocol performance. These measures are the *obtained values* to be compared in the objective function with the *desired*

values specified by the designer. The objective function is evaluated for each set of parameters of the model, i.e., each individual generated by the GA. The value of the objective function being the *fitness* of an individual.

At this stage, GA begins the optimization process applying the selection, cross-over and mutation operators generating a new *population* to be evaluated by Tangram-II until a solution is found. The rates found at a solution are used to obtain a HW/SW partition that will be evaluated for the delay and area by Synopsys and ALTERA tools. If the HW cost meets the design specification, the solution is found, otherwise another population is generated by applying a criterion to be discussed in section 3.4.

3.2 HW Implementation

The VHDL behavioral descriptions specified from the protocol state machine are inputs to Synopsys and ALTERA tools. Each state transition being associated to a VHDL *entity*. For each transition a logic circuit is synthesized by the Synopsys tool. This implementation is obtained using a standard cell library for a specific technology.

Programmable Logic Devices (PLD's) are digital integrated circuits that can be programmed to perform complex functions, i.e, PLD's can implement any boolean expression defined through programming. ALTERA implements the logic functions of each transition from the VHDL descriptions.

At the end of synthesis process, delay and area measures for each transition are obtained. These measures are used to evaluate if the cost and delay of the chosen partition are within the bounds calculated by the GA.

3.3 Objective Function

In order to formulate de optimization problem, it's necessary to combine the multiple parameters into a single *cost* or *objective function* that defines the quality of a partition. Since in general one parameter compete with another, it is useful to associate a weighting factor to each parameter in the objective function. These weighting factors are used to equalize possible differences between the parameters sensitivities. Thus, the objective function allows to compare two

partitions and to select those that satisfy the constraints. The performance parameters used depend on the Qos requirements. The objective function used in this work has the following general structure:

$$F(x_1, ..., x_n) = w_1 * |x_{1d} - x_{1o}| + ... + w_n * |x_{nd} - x_{no}| \quad (1)$$

where:

x_{id} - desired values for the performance parameters;

x_{io} - calculated values for the performance parameters;

w_i - weights atributed to each parameter.

3.4 The Partition Criterion

In order to aid the designer in the partitioning process, a criterion to correlate the rates calculated by the GA with a specified HW/SW partition must be established. The value of the rate is proportional to the number of times that a set of protocol tasks is performed. But it must be taken into account the steady state probabilities of the system, that is, even if an event has a high rate, the probability of the system being on a state in which this event is enabled may be low.

From the considerations above, it has been established as a partition criterion the product $\lambda_i * \pi$, where:

λ_i - rate of the event i determined by the GA.

π - sum of the steady state probabilities for which the event i is enabled

The partition criterion presented is used as an initial guess of the best HW/SW partition. Consequently, the designer can take a decision based on objective data and not only on his experience.

4 Results

4.1 Case Study

This example analyses the performace of a TCP protocol when it reaches the Congestion avoidance stage. Under the assumptions of the mathematical model given in Mathis [18], the congestion window traverses a perfectly periodic sawtooth that can be seen in Figure 2, where *cwnd* is the congestion window and RTT is the round-trip-time.

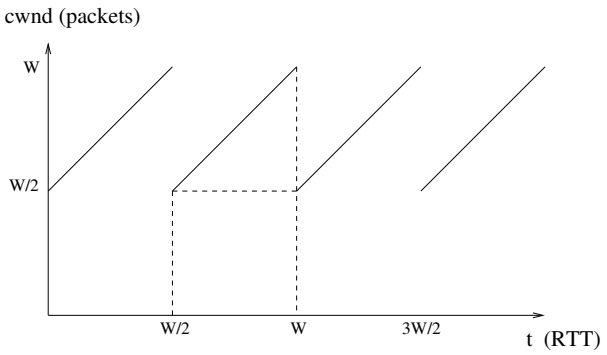

Figure 2: TCP window evolution under periodic loss.

Analysing figure 2, it can be found [18] the congestion window, in packets, as a function of the loss probability:

$$BW * RTT/MSS = C/\sqrt{p} \tag{2}$$

where BW is the bandwidth, MSS is the maximum segment size, C is a constant of proportionality and p is the loss probability. The parameters of the protocol are found by the GA, fitting the curve $(BW * RTT)/MSS$ *versus loss*, given by equation 2, with $C = 1$. Figure 3 shows this curve. The expression $BW * RTT/MSS$ is an estimate of the mean size of the congestion window [18].

4.1.1 Choice of the Partition

We choose 10 points of the curve to be approximated. The population size has 100 individuals and the criterion used to stop the GA was a number of generations equal to 10. The

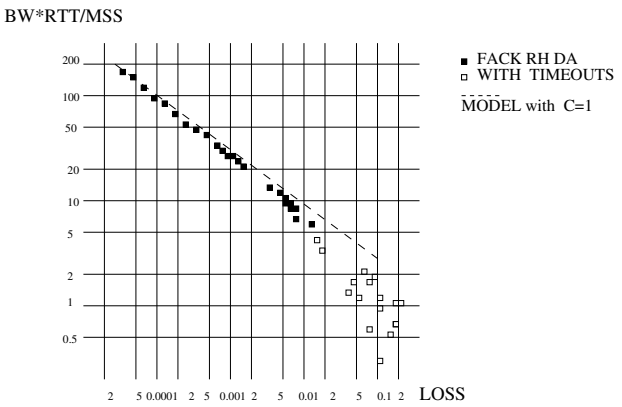

Figure 3: Window vs loss.

objective function was calculated based on the difference between the obtained window size and the desired window size. The results are presented in table 1. The small error can be explained by the value of the constant of proportionality C: for random losses its value is slightly lower than 1 [18].

Event	Rate (1/sec.)
Increase_jan (λ_1)	446.94
Send_pct (λ_2)	1078.67
Send_ack (λ_5)	984.02
Decrease_jan (λ_4)	46.17
Ack_ok (λ_3)	803.90
Send_loss_signal (λ_6)	116.41
Msg_ok (λ_7)	1

Table 1: Rates for the TCP "Congestion Avoidance" events found by GA.

It should be noted that $1/p$ packets on average are sent before a loss occurs. Using the rates of table 1, we apply the partition criterion of section 3.4 and calculate the product $\lambda_i * \pi$ for each event of the protocol. The results found are:

$$\lambda_1 * \pi_1 = 1.07 \quad (event \quad 1) \tag{3}$$

$$\lambda_2 * \pi_2 = 1.76 \quad (event \quad 2) \tag{4}$$

$$\lambda_3 * \pi_7 = 1.04 \quad (event \quad 3) \tag{5}$$

$$\lambda_4 * \pi_6 = 0.013 \quad (event \quad 4) \tag{6}$$

$$\lambda_5 * \pi_5 = 1.71 \quad (event \quad 5) \tag{7}$$

$$\lambda_6 * \pi_4 = 0.23 \quad (event \quad 6) \tag{8}$$

$$\lambda_7 * \pi_3 = 0.99 \quad (event \quad 7) \tag{9}$$

Thus, the tasks of the protocol related to the events 2 and 5 must be implemented in HW while the ones related to events 4 and 6 must be implemented in SW. The values for the events 1, 3 and 7 are too close and we should analyze the model. The events 2, 3, 5 and 7 are those related to the packet transmission and ACK's receiving. It is like these events corresponded to a *critical path of the protocol*, that is, these events are related to the tasks executed more times in the steady state. Thus, according to this criterion, the protocol tasks related to the events 2, 3, 5 and 7 should be implemented in HW while the ones related to the events 1, 4 and 6 should be implemented in SW. This is the first partition to be evaluated by Synopsys and ALTERA. If the HW cost meets the specification and the rates found by GA can be implemented by the synthetized circuits, the solution is found. Otherwise a new population is generated to continue the optimization process.

After the GA have found the rates, the rate of the event LOSS (λ_8), with exponential distribution, was fixed in 0.01 and a simulation was performed in Tangram-II. This simulation performs a trace of the variable that represents the window size. The result can be seen in figure 4.

It can be noted, observing figure 4 and the values given by the simulation, that the congestion window increment on average, for each cycle, is approximately equal to the congestion window increment of the **ideal** model (figure 2), for the same loss probability. The lower the loss probability, better results will be given by the GA. With moderate loss the model of equation 2 fits better to real cases. This result indicates that the Tangram-II model was built correctly, the values for the rates found are coherents and the behavior of the model approximates the real mecanism of the Congestion Avoidance found in many implementations of TCP.

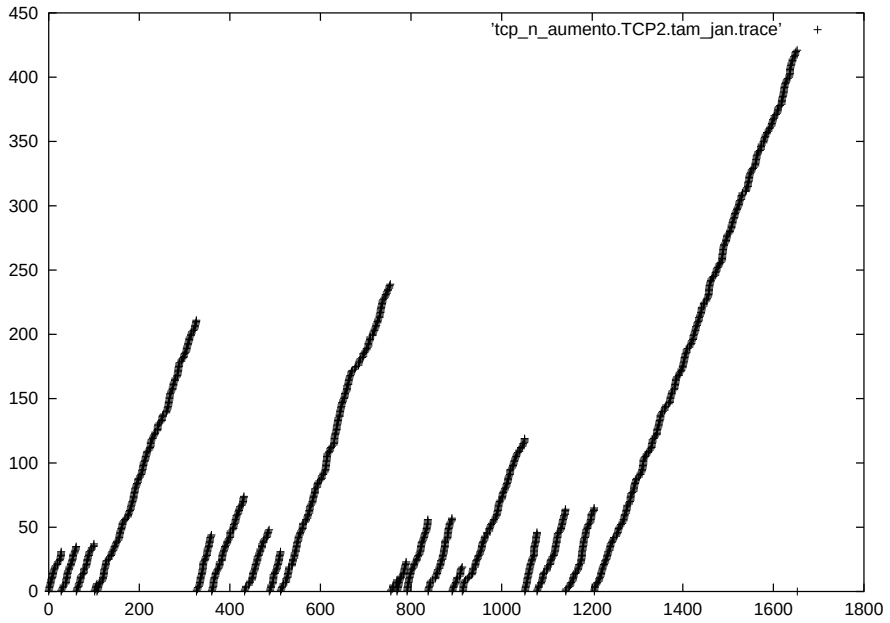

Figure 4: Congestion window evolution for exponential loss

4.1.2 Partition Evaluation

The VHDL descriptions of each state transition of the protocol are the inputs to the Synopsys and ALTERA tools. The synthesis was performed using a standard cell library ES2 with technology 0.7μm. The implementation of the logic circuits in ALTERA was performed using the environment compiler in auto mode. Table 2 shows the delay and area measures obtained for each state transition of the protocol.

These measures show the advantage of the Synopsys tool over the ALTERA tool if we compare the speed implementations. Synopsys allows a higher flexibility in cell alocation and routing, implying in a better compromise between area and speed. On the other hand, it must be stressed that PLD implementations are immediate, that is the system can be re-programmed without a new fabrication cycle, as is the case of standard cell, being an advantageous tool for rapid prototyping.

Looking at the results presented in table 2, the partition proposed in section 4.1.1 was evaluated. If we consider, for example, a specified area cost of 1 mm^2, it can be noted that transitions 2 and 7 exceed the desired requirements. Thus, these transitions must be implemented in PLD's.

Transition	ALTERA Delay	SYNOPSYS Area (mm^2)	SYNOPSYS Delay
1	17 ns	0.213	5.5 ns
2	50 ns	8.456	10 ns
3	7 ns	0.017	5 ns
4	7 ns	0.194	5 ns
5	7 ns	0.017	5 ns
6	7 ns	0.017	5 ns
7	50 ns	8.456	10 ns

Table 2: Delay and Area measures.

Nevertheless, the delay measures implemented in PLD satisfy the rates calculated by GA. The transitions 3 and 5 may be implemented in ASIC because the area measures are below 1 mm^2 and the associated delay measures satisfy the rates calculated by GA.

4.2 Convergence Time of the Algorithm

To obtain the numeric results, it was developed a program in C programming language that implements the proposed methodology. This program was executed in a Pentium-II 400Mhz with 128 MB of RAM memory, running the Linux operating system.

The algorithm has converged in some minutes to the number of points, population size and number of generations given at the above example. For experiments with 5000 individuals or more, 10 points, 10 generations and/or a specified error of 0.0001 to the objective function, the algorithm took around 2 to 4 hours to converge.

The convergence time shows that the program of the Tangram-II tool that executes the numerical method to solve the Markov chain is faster enough and despite being called hundreds of thousands times during the optimization, depending on the population size, the number of points to be fitted and the number of generations specified, the algorithm is executed in a few hours, in the environment mentioned at the beginning of this section. This is a short time if

we consider the whole process of synthesis.

5 Final Remarks

High-speed networks require specific communication protocols design. An important issue to improve the performance of high-speed protocols is the integration between SW and HW. Due to the increasing demand for protocols with high throughput, solutions using hardware are being more frequently investigated. For example, full ASIC-based forwarding engines deliver more than 10-times higher throughput than routers that rely entirely on microprocessors [1].

Actually, the choice of the best HW/SW partition of a communication protocol is a subjective issue and the design process too slower and subject to errors. It is important to supply the designer with measurements and objective criteria in order to aid him in taking design decisions. With this goal, the first contribution of this work was to develop a methodology based on cost and performance requirements that supply the designer with those measures. The integration of many powerful techniques simplifies the formulation of the described methodology, where the GA integrated with Tangram-II was fundamental.

Although the GA has a high computational cost, it's simple to be implemented and avoids complex error functions like, for example, the exponential function used in *Simulated Annealing* [19].

The results described show that the methodology still has a great potential to be explored in connection with other applications. Protocols that have heavy computational costs with their performance directly related to the speed of the processing at the nodes of the network are the best candidates to apply the proposed methodology. For example, protocols for reliable Multicast transmission, protocols for errors recovery [20] and the ones related to data cryptography [21].

Acknowledgements

The authors would like to express their gratitude to Professor Edmundo Souza e Silva,

coordinator of the LAND laboratory of COPPE/UFRJ and to all his team, for providing the computational resources and technical support related to the Tangram-II tool. To Professor Jos Ferreira de Rezende for the many profitable discussions on the subject.

This work was partially supported by grants from UFRJ, FUJB, CNPq, CAPES, COFECUB, FAPERJ and REENGE.

References

[1] E. Networks 2000. `http://www.extrenetworks.com/technology/arch.asp`.

[2] D. Gajski and F. Vahid, "Specification and design of embedded software/hardware systems", *IEEE Design and Tests of Computer*, vol. 12, no. 1, pp. 53–67, Jan. 1995.

[3] S. Fischer, J. Wytrebowicz and S. Budkowski, "Hardware/software co-design of communication protocols", in *Proceedinds of IEEE 22nd Euromicro Conference*, 1996.

[4] T. B. Ismail, M. Abid and A. Jerraya, "Cosmos: A codesign approach for communication systems", in *3th International Workshop on Hardware/Software Codesign*, pp. 17–24, 1994.

[5] D. Gajski and F. Vahid, "Specification and design of embedded hardware/software systems", in *IEEE Design and Tests of Computer*, pp. 56–67, 1995.

[6] A. Kalavade and E. A. Lee, "Hardware/software codesign using ptolomey - a case study", in *International Workshop on Hardware/Software Codesign*, 1992.

[7] J. Madsen, J. Grode, P. V. Knudsen, M. E. Petersen and A. Haxthausen, "Lycos: The lyngby co-synthesis system", in *Design Automation for Embedded System*, 1997.

[8] P. H. Chou, R. B. Ortega and G. Borriello, "The chinook hardware/software co-synthesis system", in *8th International Symposium on System Synthesis*, 1995.

[9] P. Maciel, E. Barros and W. Rosenstiel, "Estimating functional unit number in the pish codesign system by using petri nets", in *XII Symposium on Integrated Circuits and Systems Design*, pp. 32–35, Sept. 1999.

[10] J. I. Hidalgo and J. Lanchares, "Functional partitioning for hardware/software codesign using genetic algorithm", in *Proceedings of the 23rd Euromicro Conference*, 1997.

[11] M. Mitchell, *An Introduction to Genetic Algorithms.* MIT Press, 1996.

[12] A. P. C. Silva, "Tangram-ii user's manual", tech. rep., Universidade Federal do Rio de Janeiro, Oct. 2000. `http://www.land.ufrj.br`.

[13] R. Carmo, L. Carvalho, E. Sousa e Silva, M. Diniz and R. Muntz, "Performance/availability modeling with the Tangram-II modeling environment", *Performance Evaluation*, vol. 33, no. 1, pp. 45–65, June 1998.

[14] Synopsys, Inc., *Synopsys Online Documentation, v1999*, 1999.

[15] Altera, Corp., *Data Book*, 1996.

[16] IEEE Std 1076-1987, *IEEE Standard VHDL Language Reference Manual*, Mar. 1988.

[17] S. M. Ross, *Stochastic Processes.* John Wiley & Sons, 1983.

[18] M. Mathis, J. Semke and J. Mahdavi, "The macroscopic behavior of the tcp congestion avoidance algorithm", *Computer Communication Review*, vol. 27, no. 3, pp. 1–16, July 1997.

[19] A. Laarhoven, *Simulated Annealing: theory and applications.* D. Reidel, 1987.

[20] J. Nonnenmacher, E. Biersack and D. Towsley, "Parity based loss recovery for reliable multicast transmission", in *Proceedings of ACM SIGCOMM'97*, 1997.

[21] M. Abadi, "Security protocols and specifications", in *In Foundations of Software Science and Computation Structures: Second International Conference, FOSSACS '99*, 1999.