HW/SW Codesign of Handoff Protocol for Wireless ATM Networks based on Performance Optimization using Genetic Algorithm

Ricardo N. B. Lima[1], Marcio N. Miranda[1,2], José V. S. Filho[1],
{*belem,marcio,valentim*}*@gta.ufrj.br*
Aloysio C. P. Pedroza[1] and Antônio C. de Mesquita[1]
aloysio@gta.ufrj.br and *mesquita@coe.ufrj.br*

[1] Grupo de Teleinformática e Automação - GTA
COPPE/EE - Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
[2] Instituto de Informática
Universidade Federal de Goiás

Abstract

A design-to-hardware methodology applied to the synthesis of handoff protocol for wireless ATM 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. This 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. Hardware implementations using Synopsys and ALTERA tools are presented and compared to software implementations. 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 handoff protocols for Wireless ATM (WATM) networks not always allows high-speed operation. Another important factor to improve performance of high-speed protocols is the integration between software (SW) and hardware (HW).

In general, no timing constraints tasks can be implemented in SW while those that require higher performance must be implemented in HW. Due to the increasing demand for high throughput protocols, 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 the development of 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 those to be implemented in HW, i. e., the selection of the best HW/SW partition. One should note that because of the higher HW implementation costs, upon deciding for a given partition, a compromise between the desired performance and the implementation costs must be taken into account. 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 their own experience and informal techniques to define a HW/SW partition, which considerably limits the extent of the solutions space search. In this case it is useful to provide the designer with performance evaluation tools, in order to derive quantitative measures of the effectiveness of a given partition. In spite of the existing efforts to automate this process, yet there are few resources to aid the specification refinement, development and partitioning of a protocol [2][3].

Several projects currently in progress integrate both HW and SW into the same design process: COSMOS [4], SpecSyn [5], Ptolemy [6], LYCOS [7], Chinook [8] e PISH [9]. Fischer [3] proposes an integrated HW/SW design to meet high performance in distributed systems, such as multimedia systems. It presentes 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 (GAs) [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 is done in this work.

This work implements the handoff protocol for WATM [12] using HW/SW Codesign methodology based on performance requirements. The codesign technique relies on the use of Tangram-II [13][14], a performance analysis tool, on the Synopsys [15] and ALTERA [16] tools plus a genetic algorithm (GA) as optimization method. The protocol behavior is represented by a state diagram, where each state transition is a set of clauses and expressions to be evaluated, in order to select the best HW/SW partition in terms of a performance/cost compromise.

Section 2 briefly describes the handoff control protocol. The design methodology is described in section 3. In section 4 the obtained results are presented. Final remarks are drawn on Section 5.

2. The Handoff Control Protocol

Unlike connection setup protocols in ATM, the location of a terminal with respect to the network when dealing with mobile ATM can not be deduced from its endpoint address. Additional addressing schemes and protocols are needed to locate and track mobile terminals, along with suitable modifications to the connection setup process. The path of the connection for a mobile user needs to be frequently rerouted as the user moves from one cell to another. An efficient handoff control protocol does dynamic path re-routing instead of whole path re-establishment. Handoff schemes can be classified into path re-routing and path extension. The first scheme is based on removing a part of the existing connection and adding a new sub-path from the point of detachment (the “cross-over point”). Different methods to select the cross-over point give different performance for handoff control in terms of latency, data loss and resource utilization. The second scheme is based on extending the original connection from previous access point to new access point. Although it looks relatively simple for a single handoff, the overall process has to include other complicated algorithms for loop removal and route optimization [12]. Figure 1 presents the specification of the handoff control protocol given by a condition-action Petri net.

3. Design Methodology

Figure 2 illustrates the design methodology. Initially, the protocol is described by a state machine, where each state transition represents a task or a set of tasks to be performed. In this phase, the state diagram is analyzed in order to verify any live-locks or dead-locks. At this stage, the design process follows two parallel branches. In the first one, the specification is detailed and the model is built using the Tangram-II environment. The parameters of the model which meet the performance requirements are found by the use of Tangram-II tool and a genetic algorithm (GA). GA is used as an optimization tool to minimize an objective function of the performance parameters. In the second branch the delay and area measures are obtained by Synopsys and ALTERA tools. Synopsys gives, as output, a logic circuit from which the delay and area measurements can be obtained. ALTERA gives the delay values obtained.

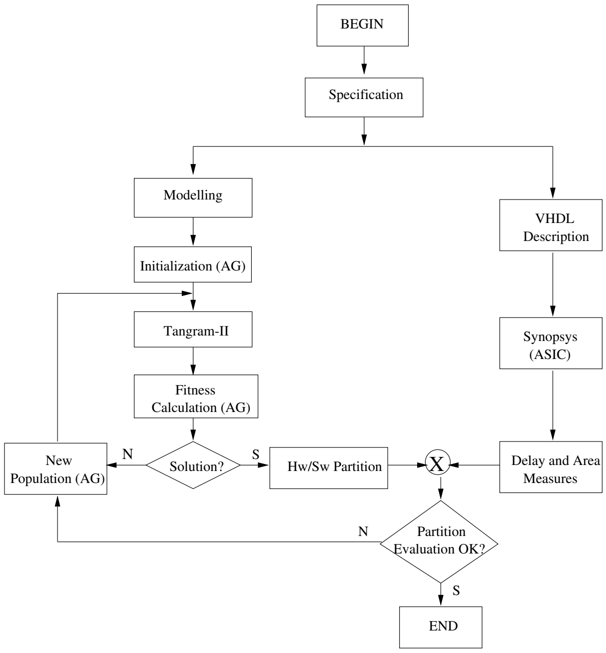

Figure 2. The design methodology.

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 which meet the performance requirements. These rates help the designer to establish a partition criterion.

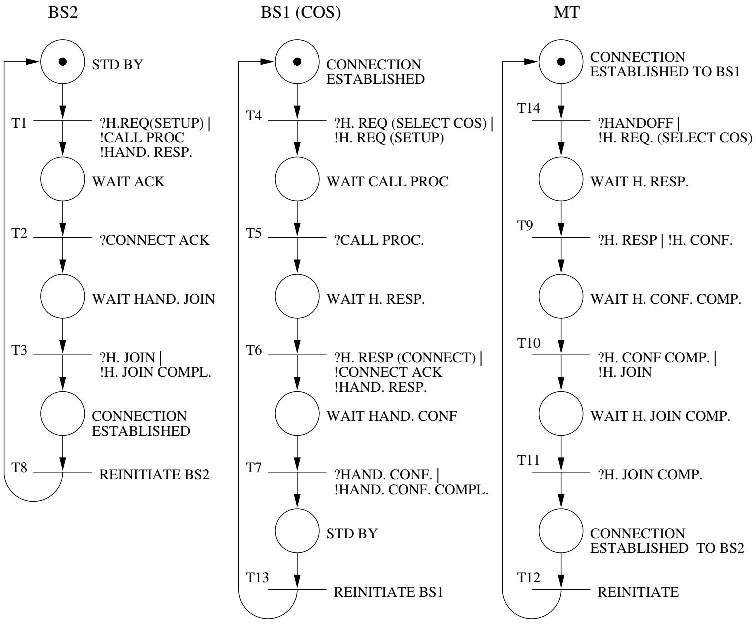

Figure 1. The handoff control protocol.

The numerical values needed to solve the Tangram-II model are given by the GA through a random number generator. Tangram-II uses these numerical values to calculate the steady state probabilities of the Markov chain associated to the model. These probabilities are used to estimate the protocol performance through the value of an objective function 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, 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 by the delay and area measures supplied by Synopsys and ALTERA tools. If the delay and area of the HW partitions meet the design specification, the solution is found, otherwise another population is generated.

3.2. HW Implementation

Programmable Logic Devices (PLDs) are digital integrated circuits that can be programmed to perform complex functions, i. e., PLDs can implement any boolean expression defined through programming, such as ALTERA tool. The VHDL behavioral descriptions specified from the protocol state machine are inputs to Synopsys and ALTERA tools. Each state transition that performs a set of operations is 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 integration specific technology. The cells allocation and routing process and the optimization step are made automatically. The same transition is implemented by ALTERA tool. At the end of this process, delay and area measures for each transition are obtained by Synopsys and delay measures are obtained by ALTERA.

3.3. Objective Function

In order to formulate the optimization problem, the multiple parameters must be combined into a single *cost* or *objective function*, whose value defines the quality of a partition. Once one parameter generally competes 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 used performance parameters depend on the QoS requirements. The objective function used in this work has the following general structure:

$$F(x_1, ..., x_n) = \sum_{1 \leq i \leq n} w_i * |x_{id} - x_{io}| \qquad (1)$$

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 assist the designer in the partitioning process, a criterion must be established to correlate the state transition rates calculated by the GA with a specified HW/SW partition. The value of the rate is proportional to the number of times that a set of protocol tasks is performed. But 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.

Regarding the above consideration, a partition criterion has been established as product $\lambda_i * \pi_i$, where:

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

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

The presented partition criterion is used as an initial guess of the best HW/SW partition. This partition is analyzed through the delay and area results obtained from Synopsys and ALTERA tools.

4. Results

In this section, the obtained results with the design of the handoff protocol for WATM using HW/SW Codesign methodology based on performance optimization with GA are presented. The performance evaluation, choice of the partition, partition evaluation processes and a final performance analysis of the system are discussed.

4.1. Performance Evaluation

This section analyses the performace of a handoff control protocol for a wireless ATM network, specified by figure 1, using the signaling sequence given by Acharya et al. [12]. The performance analisys will be focused on the packet loss probability as a function of the transmission rate. The handoff transition delay (T_d) gives the minimum time the switch takes to change the routing table for multiple VCs. This number is directly related to the packet loss rate caused by handoff in our experiment. In the model used [17], a packet loss happens when a VC routing table change occurs within a stream of cells all belonging to the same packet. Let us assume that the handoff event occurs randomly within the packet transmission interval $T = 1/R$, where R is the packet transmission rate. So the packet loss probability P of the periodic packet stream is given by $P = T_d * R$. Figure 3 shows the curve P versus R.

4.2. Choice of the Partition

It was chosen 6 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

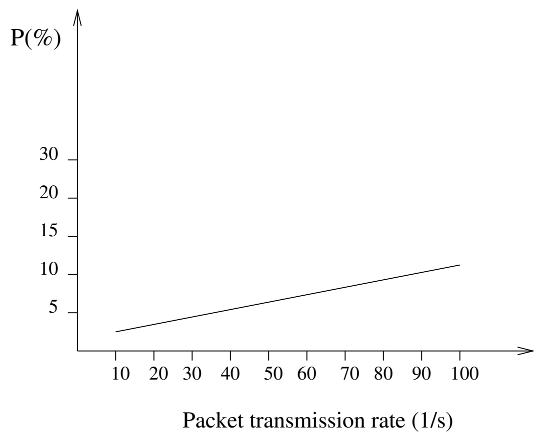

Figure 3. Packet loss probability as a function of the transmission rate.

10. The objective function was calculated based on the difference between the obtained and the desired loss probability, i.e., to calculate the parameters of the protocol we must fit the curve of figure 3. The results are presented in table 1. The error for the sum of the six loss probabilities found was 0.006.

Table 1. Rates for the protocol events found by GA.

Event	Rate (1/sec.)
t_1 (λ_1)	42.34
t_2 (λ_2)	60.12
t_3 (λ_3)	421.63
t_4 (λ_4)	344.91
t_5 (λ_5)	477.38
t_6 (λ_6)	397.72
t_7 (λ_7)	156.63
t_8 (λ_8)	14.31
t_9 (λ_9)	20.76
t_{10} (λ_{10})	8.43
t_{11} (λ_{11})	114.90
t_{12} (λ_{12})	338.55
t_{13} (λ_{13})	474.19
t_{14} (λ_{14})	68.03

Using the rates of table 1 and the steady state probabilities calculated by Tangram-II, one can apply the partition criterion of section 3.4 and calculate the product $\lambda_i * \pi_i$ for each event of the protocol. The results can be seen in table 2.

Tasks of the protocol related to the transitions t_4, t_5, t_6,

Table 2. Values for partition criterion established.

Event	Value
$\lambda_1 * \pi_1$	1.52
$\lambda_2 * \pi_2$	3.05
$\lambda_3 * \pi_3$	5.22
$\lambda_4 * \pi_4$	97.61
$\lambda_5 * \pi_5$	14.31
$\lambda_6 * \pi_6$	27.75
$\lambda_7 * \pi_7$	70.32
$\lambda_8 * \pi_8$	0.20
$\lambda_9 * \pi_9$	1.17
$\lambda_{10} * \pi_{10}$	0.17
$\lambda_{11} * \pi_{11}$	1.88
$\lambda_{12} * \pi_{12}$	80.21
$\lambda_{13} * \pi_{13}$	13.60
$\lambda_{14} * \pi_{14}$	3.67

Table 3. Delay and Area measures.

Transition	Synopsys Area (mm^2)	Synopsys Delay	ALTERA Delay
1	0.150	20 ns	60 ns
2	0.015	5 ns	30 ns
3	0.023	5 ns	30 ns
4	0.223	16 ns	60 ns
5	0.015	5 ns	30 ns
6	0.130	20 ns	60 ns
7	0.023	5 ns	30 ns
8	0.015	5 ns	30 ns
9	0.024	5 ns	30 ns
10	0.015	5 ns	30 ns
11	0.015	5 ns	30 ns
12	0.015	5 ns	30 ns
13	0.015	5 ns	30 ns
14	0.223	16 ns	60 ns

t_7, t_{12}, t_{13} must be implemented in HW because these transitions have a higher $\lambda_i * \pi_i$ product, while those related to the other events, with a lower product, must be implemented in SW. This is the first partition to be evaluated by HW implementation values. If the HW area 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. One should note that, according to the results, the transitions to be implemented in HW are those that speeds up the handoff process (see figure 1). This indicates that the Tangram-II model was built correctly, and that the behavior of the model approximates the real mecanism of the Handoff Control protocol found in many implementations of Wireless ATM.

4.3. Partition Evaluation

The VHDL behavioral descriptions of each state transition of the protocol are the inputs to the Synopsys and ALTERA tools. The synthesis is performed by Synopsys using a standard cell library ES2 with technology 0.7μm. The ALTERA implementation uses Max+PlusII design compiler. Table 3 shows the measures obtained for each state transition of the protocol.

Concerning the results presented in table 3, the proposed partition in section 4.2 was evaluated. One might consider, for example, a specified area cost of 1 mm^2 for HW, it can be noted that all the transitions indicated by the partitioning process (t_4, t_5, t_6, t_7, t_{12} and t_{13}) may be implemented in HW using standard cells or PLD because the area measures are below 1 mm^2 and the associated delay measures satisfy the rates calculated by GA.

4.4. Performance Analysis of the Chosen Partition

Once the chosen partition was evaluated, Tangram-II uses as input the calculated measures given by Synopsys and ALTERA in order to obtain the final performance of the protocol implementation. Figure 4 compares the handoff protocol performance entirely implemented in SW, Synopsys and ALTERA with the performance given by the chosen partition. Two implementations of the chosen partition were performed. The first one using transitions implemented in ALTERA and SW and the other one using transitions implemented in Synopsys and SW. The metric used to compare these implementations is the loss probability as a function of the transmission rate.

One can note that loss probability in Synopsys implementation is approximately 900 times lower than loss probability found in SW implementation. ALTERA implementation gives losses 30 times higher than Synopsys implementation, nevertheless ALTERA is an important tool for rapid prototyping of communication systems with lower costs.

The HW/SW partition implemented using ALTERA gives results 6 times better than SW implementation. The HW/SW partition implemented using Synopsys gives results 40 times better than SW implementation. The solution using ALTERA satisfies the performance requirements with lower costs than Synopsys.

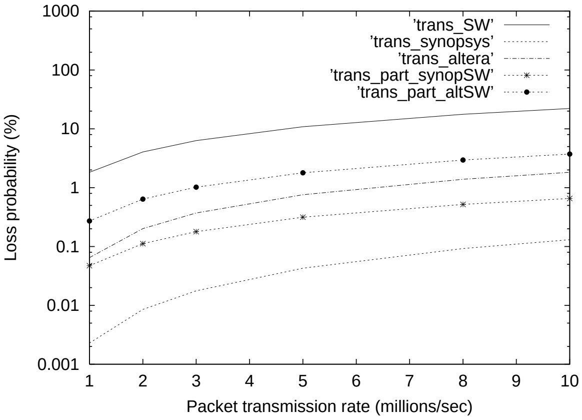

Figure 4. Packet loss probability as a function of the transmission rate.

5. Final Remarks

High-speed networks require specific communication protocols design. With ATM technology playing an increasing important role in this context, it is natural to expect the emergence of wireless ATM networks enabling multimedia access for mobile end systems. An important issue to improve performance of high-speed protocols is the integration between SW and HW during their implementation. Due to the increasing demand for protocols with high throughput, solutions using hardware are being more frequently investigated. Protocols that have high computational costs with their performance directly related to the speed of the processing are the best candidates to have the proposed methodology applied. This is the case of the handoff protocols for WATM like we have seen.

The design of handoff protocol using a methodology based on performance analysis techniques and GA is presented with the objective of finding a faster implementation with a low cost. The described methodology integrates many powerful techniques and simplifies the problem formulation. In this context, the integration of GA with Tangram-II was fundamental. Although the GA has a high computational cost, it is simple to be implemented and avoids complex error functions like, for example, the exponential function used in *Simulated Annealing* [18]. The designer is supplyed with measurements and objective criteria in order to assist him in taking design decisions. There is no other HW/SW Codesign environment in literature that uses performance optimization and GA *during* the design cycle of communication protocols.

References

[1] Extremenetworks. Technology - hardware architecture. *Internet Draft*, 2000. `http://www.extremenetworks.com`.

[2] D. Gajski and F. Vahid. Specification and design of embedded software/hardware systems. *IEEE Design and Tests of Computer*, 12(1):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 e A. Jerraya. COSMOS: A Codesign Approach for Communication Systems. *3th International Workshop on Hardware/Software Codesign*, pages 17–24, 1994.

[5] D. Gajski, F. Vahid, S. Narayan, and J. Gong. System-Level Exploration with SpecSyn. *Design Automation Conference*, pages 812–817, 1998.

[6] A. Kalavade e E. A. Lee. Hardware/Software Codesign using Ptolemy - A Case Study. *Proceedings of IEEE 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. *Design Automation for Embedded Systems*, 2(2):1–43, Feb. 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 e W. Rosenstiel. Estimating Functional Unit Number in the PISH Codesign System by Using Petri Nets. In *XII Symposium on Integrated Circuits and Systems Design*, pages 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. Acharya, J. Li and A. Bakre. Design and prototyping of location management and handoff protocols for wireless ATM networks. In *Proceedings of ICUPC'97*, 1997.

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

[14] R. Carmo, L. Carvalho, E. Sousa e Silva, M. Diniz, and R. Muntz. Performance/availability modeling with the Tangram-II modeling environment. *Performance Evaluation*, 33(1):45–65, June 1998.

[15] Synopsys, Inc. *Synopsys Online Documentation, v1998.02*, 1998.

[16] Altera, Corp. *Data Book and Max + PlusII Getting Started*, 1997.

[17] R. Yuan, K. Biswas, L.J. French, J. Li and D. Raychaudhuri. A signaling and control architecture for mobility support in wireless ATM networks. In *Proceedings of MONET*, pages 287–298, Mar. 1997.

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