HW/SW Codesign de Protocolos Baseado na Otimização de Desempenho por Algoritmos Genéticos *

Marcio Nunes de Miranda[1,2†]
marcio@gta.ufrj.br

Ricardo N. Belem Lima[1,2‡]
belem@gta.ufrj.br

Aloysio C. P. Pedroza[1,2]
aloysio@gta.ufrj.br

Antônio C. de Mesquita Filho[2]
mesquita@coe.ufrj.br

[1] Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro

[2] COPPE/EE - Programa de Engenharia Elétrica - Depto. de Eletrônica
Caixa Postal 68504 - CEP 21945-970 - Rio de Janeiro - RJ - Brasil
FAX: + 55 21 2906626

Resumo

Este trabalho apresenta uma metodologia de HW/SW *Codesign* aplicada à síntese de protocolos baseada em técnicas de análise de desempenho e em algoritmos genéticos. O processo implica na escolha das operações do protocolo que devem ser implementadas em *hardware* e naquelas que devem ser implementadas em *software*, ou seja, na escolha da melhor *partição* HW/SW. É usado um algoritmo genético para otimizar uma função de erro definida pelo desempenho e pelo custo de implementação do protocolo. O desempenho do protocolo é analisado pela ferramenta Tangram-II e o custo do *hardware* associado é calculado pelas ferramentas Synopsys e ALTERA. A metodologia pretende auxiliar o projetista de protocolos na seleção da melhor partição HW/SW.

Palavras-chave: HW/SW codesign, algoritmos genéticos, especificação, verificação, implementação e testes de sistemas distribuídos e protocolos

Abstract

A design-to-hardware implementation methodology applied to the synthesis of protocols based on performance analysis techniques and genetic algorithms is presented. The process implies in 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. A genetic algorithm is used 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 tool called Tangram-II and the associated cost of the hardware is calculated by the Synopsys and ALTERA tools. The methodology is intended as a tool to help protocol designers in the selection of the best performance/cost compromise.

Keywords: HW/SW codesign, genetic algorithms, specification, verification, implementation and testing of distributed systems and protocols

*Trabalho realizado com recursos da UFRJ, FUJB, CNPq, CAPES, COFECUB, FAPERJ e REENGE

†Aluno de doutorado.

‡Aluno de doutorado.

1 Introdução

As redes de computadores de alta velocidade exigem protocolos de comunicação específicos. O uso da otimização em *software* (SW) na elaboração de protocolos para as atuais redes nem sempre permitem operações a altas taxas de velocidade. Um fator importante para o bom desempenho dos protocolos de alta velocidade é a utilização integrada de *hardware* e *software* durante sua implementação. Algumas operações do sistema, normalmente as menos críticas em termos de velocidade, podem ser implementadas em SW, enquanto que aquelas que necessitam de uma maior velocidade devem ser implementadas em *hardware* (HW). Devido à crescente demanda por protocolos com altas vazões, as soluções que utilizam HW são cada vez mais investigadas. Num roteador, por exemplo, o encaminhamento de pacotes totalmente implementado em ASIC's (*Application Specific Integrated Circuits*) pode fornecer uma vazão até dez vezes maior do que se fosse implementado inteiramente em microprocessadores comuns [1].

O princípio do *Codesign* [2] é a realização de um projeto cooperativo baseado em dois ambientes de projeto específicos, HW e SW, onde pode-se verificar e simular todo o sistema em qualquer etapa do projeto. À medida que se avança no detalhamento da especificação, torna-se necessária a aplicação de técnicas que auxiliem o projetista a tomar decisões relativas à partição HW/SW das operações do protocolo. Estas operações representam o comportamento do protocolo e podem ser descritas por um diagrama de estados, onde cada transição de estado representa um conjunto de cláusulas e expressões a serem avaliadas.

Apesar dos tempos de execução em HW serem, normalmente, muito menores do que na implementação em SW, o custo deste tipo de solução é maior. Portanto, deve-se estabelecer um compromisso entre o desempenho desejado e o custo de implementação no momento de se decidir qual parte do protocolo deve ser implementada em HW e qual parte deve ser implementada em SW. Esta etapa é denominada de *particionamento* e sua decisão tem impacto direto no desempenho do sistema de comunicação do qual o protocolo faz parte.

O projetista ainda dispõe de poucos recursos para auxiliá-lo no refinamento da especificação, no desenvolvimento do projeto e na partição HW/SW de um dado protocolo [2][3]. Normalmente, a experiência do projetista aliada a técnicas informais constituem a base para se definir a melhor partição, o que limita a exploração do espaço de soluções. Neste caso, é útil fornecer ao projetista subsídios que auxiliem-no na escolha da melhor partição.

Muitos grupos de pesquisa em renomadas universidades estão desenvolvendo ambientes de projeto baseados na metodologia de *Codesign*. Dentre eles podemos citar: COSMOS [4], SpecSyn [5], Ptolomey [6], LYCOS [7], Chinook [8] e PISH [9]. Estes ambientes de *Codesign* diferem na linguagem de especificação, no método de particionamento, na arquitetura alvo e nos métodos de validação utilizados.

Fischer [3] ressalta a importância do uso combinado do HW e do SW para se alcançar um alto desempenho de sistemas distribuídos, como sistemas multimídias. É apresentado um ambiente de desenvolvimento para o suporte das etapas de projeto e implementação. Hidalgo [10] propõe uma metodologia para a partição HW/SW baseada no uso de algoritmos genéticos. A divisão dos blocos funcionais em HW e SW é baseada na avaliação de uma função objetivo. Esta função utiliza uma tabela previamente estabelecida para os valores dos desempenhos, não calculando esses parâmetros durante o processo de busca da melhor partição, como é realizado no presente trabalho.

O objetivo deste trabalho é apresentar uma nova metodologia que automatize o processo de partição de um protocolo estabelecendo um critério objetivo para que a mesma seja realizada sem necessidade do projetista utilizar técnicas manuais. Esta proposta utiliza de forma conjunta uma ferramenta de análise de desempenho, ferramentas de síntese de HW em ASIC e PLD (*Programmable Logic Devices*) e um algoritmo genético.

Na metodologia desenvolvida neste trabalho, os protocolos são especificados, inicialmente, por um modelo de máquina de estados, onde cada transição de estado representa uma operação ou um conjunto de operações a serem avaliadas. Este modelo serve como base para a descrição na linguagem da ferramenta de análise de desempenho Tangram-II [11][12] e para a especificação na linguagem VHDL [13] para a síntese de HW.

A ferramenta Tangram-II auxilia o projetista na identificação das operações críticas em termos de tempo (retardo), assim como no cálculo de outros parâmetros de desempenho tais como vazão e probabilidade de perda de mensagens. Um algoritmo genético é utilizado visando a otimização de uma função objetivo composta por esses parâmetros. As ferramentas Synopsys [14] e ALTERA [15] fornecem informações de custo de HW e medidas de atraso a partir da análise dos circuitos sintetizados.

Este artigo está estruturado da seguinte forma: a seção 2 apresenta uma descrição suscinta das principais técnicas utilizadas na metodologia proposta; a metodologia de projeto é descrita na seção 3; na seção 4 são apresentados os resultados obtidos. As considerações finais encontram-se na seção 5.

2 Técnicas de projeto

A metodologia apresentada neste trabalho, congrega uma série de técnicas em um único processo e propõe uma solução para o problema da automatização da partição HW/SW. O ponto de partida consiste na adoção de um projeto cooperativo, HW/SW *Codesign*, baseado em dois ambientes de projeto específicos.

A modelagem e a análise de desempenho de um determinado protocolo é realizada através do uso da ferramenta Tangram-II. Um algoritmo genético é utilizado visando a otimização de uma função objetivo composta por parâmetros de desempenho da partição.

A síntese do HW é realizada utilizando-se a linguagem VHDL, que é utilizada como entrada para as ferramentas Synopsys e ALTERA, de onde são obtidas as medidas de atraso e área dos circuitos sintetizados. A seguir, são descritos os conceitos e técnicas utilizadas na metodologia proposta.

2.1 HW/SW Codesign

A interação entre os caminhos de desenvolvimento de projetos de HW e SW é a idéia principal do *codesign*. Esta interação entre os projetos ocorre de diferentes modos e em diferentes níveis do ciclo de desenvolvimento. A figura 1 ilustra uma abordagem de *codesign* comumente empregada. O processo se inicia com uma representação funcional do sistema, independente do domínio do projeto, HW ou SW. Algumas representações do sistema são realizadas utilizando-se máquinas de estado finitas (MEF) e processos concorrentes.

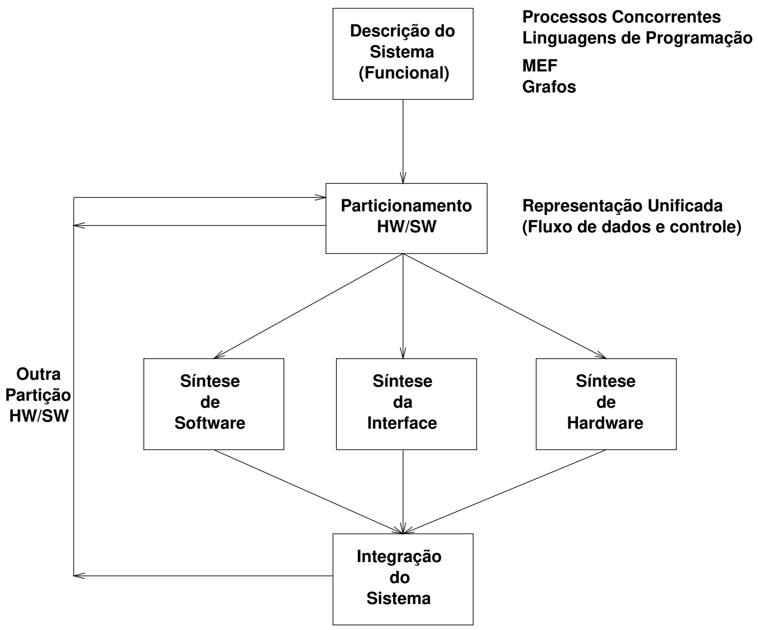

Figura 1: Abordagem hardware/software codesign.

O processo de particionamento determina quais funções serão implementadas em HW e quais serão implementadas em SW. Algoritmos de *clustering*, de melhoria incremental e genéticos são alguns exemplos de algoritmos usados neste processo. Diversos grupos de pesquisa adotam, basicamente, a metodologia apresentada na figura 1 como solução para a implementação HW/SW de diversos sistemas.

2.2 Ferramentas de Apoio ao Projeto

Para a modelagem do protocolo e obtenção das medidas de desempenho e custo que orientam a escolha da melhor partição HW/SW do protocolo são utilizadas três ferramentas: o Tangram-II, o Synopsys e o ALTERA.

O Tangram-II é uma ferramenta utilizada para especificação e análise de desempenho de sistemas. Entre suas características principais estão: possuir uma interface gráfica baseada no paradigma de orientação a objetos e uma variedade de métodos de solução para obtenção de medidas de interesse relacionadas ao modelo. O sistema a ser modelado é representado por uma coleção de objetos, cujo estado é representado por um conjunto de variáveis e o seu comportamento é definido por *eventos* e *mensagens*, assim como pelas *condições* que habilitam os eventos e as *ações* executadas quando um evento é disparado ou uma mensagem é recebida. Maiores detalhes sobre a sintaxe da ferramenta Tangram-II podem ser obtidos em Silva [12].

O projeto de circuitos integrados se inicia a partir de uma descrição comportamental do sistema na linguagem VHDL. Implementações em ASIC e PLD são sintetizadas de modo a se obter uma variedade maior de opções de projeto. O Synopsys produz como resultado um circuito lógico do qual são extraídas informações referentes ao custo de implementação e medidas de atraso. O ALTERA é um ambiente de prototipagem rápida capaz de produzir um circuito integrado que realiza funções complexas através de lógica reprogramável. Deste circuito são extraídas medidas de atraso.

2.3 Algoritmos Genéticos

Os algoritmos genéticos (AG's) [16][17] vêm tendo larga aceitação devido à simplificação que eles permitem na formulação e solução de problemas de otimização. Esta característica é particularmente útil em problemas de otimização complexos, envolvendo um grande número de variáveis e, consequentemente, espaços de solução de dimensões elevadas. Além disso, em muitos casos onde outras estratégias falham na busca de uma solução, os AGs convergem. A estrutura básica do AG é mostrada na figura 2.

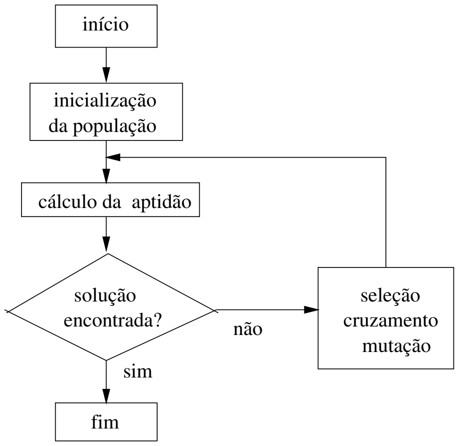

Figura 2: Um algoritmo genético simples.

AGs simples normalmente trabalham com descrições de entrada formadas por cadeias de bits de tamanho fixo. Cada cadeia corresponde a um *cromossomo*, onde um ou mais parâmetros são codificados. Neste caso, cada parâmetro do modelo corresponde a um *gene*. Deve ser notado que cada cromossomo, chamado de *indivíduo*, corresponde a um ponto no espaço de soluções do problema de otimização. O processo de solução adotado nos AGs consiste em gerar, aleatoriamente, um grande número de indivíduos, uma *população*, de forma a promover uma varredura tão extensa quanto necessária do espaço de soluções.

Após a inicialização da população, cada iteração do AG corresponde à aplicação de um conjunto de quatro operações básicas: cálculo de aptidão, seleção, cruzamento e mutação. Ao fim destas operações cria-se uma nova população, chamada de *geração* que, espera-se, represente uma melhor aproximação da solução do problema de otimização que a população anterior. O tamanho da população e o número de gerações definem diretamente o tamanho do espaço de busca a ser coberto.

3 Particionamento HW/SW de Protocolos

A metodologia para o particionamento HW/SW de um protocolo, apresentada neste trabalho, congrega as técnicas descritas na seção 2 e propõe uma solução que auxilia o projetista na escolha da melhor partição, de maneira a atender as medidas de desempenho e custo estabelecidas na especificação. Nos parágrafos que se seguem é realizada uma descrição detalhada dos passos da metodologia.

3.1 Metodologia Proposta

A figura 3 apresenta o diagrama de blocos da metodologia. Inicialmente, o projetista especifica os valores de desempenho e custo desejados. Nesta fase, o protocolo é descrito por um modelo de máquina de estados, onde cada transição de estado representa uma operação ou um conjunto de operações a serem avaliadas. A partir desta especificação, o fluxo de projeto segue dois caminhos paralelos. O primeiro caminho é responsável pela determinação dos parâmetros que atendem aos requisitos de desempenho do protocolo através do uso da ferramenta de análise de desempenho e do AG. O segundo caminho é responsável pela extração das medidas de atraso e custo a partir da análise das implementações em HW de todas as transições do protocolo, tanto em ASIC quanto em PLD.

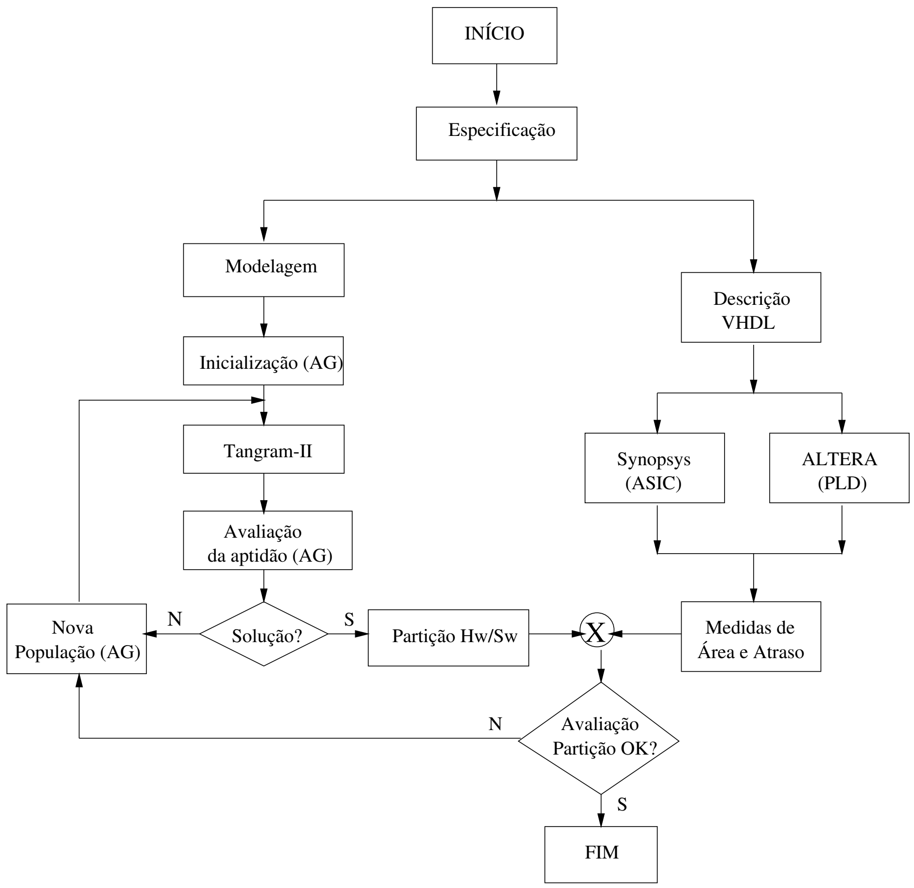

Figura 3: Metodologia de projeto.

3.1.1 Otimização do Desempenho

A partir da especificação do protocolo em máquina de estados é realizada a sua modelagem na ferramenta Tangram-II, através de um modelo com parâmetros, que são as taxas dos *eventos*. Essas taxas têm distribuição exponencial e, portanto, cada modelo tem uma Cadeia de Markov a Tempo Contínuo (CMTC) [18] associada. O problema inicial consiste em determinar as taxas de transição de estado da CMTC, obedecendo certas restrições de medidas de desempenho impostas pela especificação. A determinação dessas taxas fornecem subsídios para o estabelecimento de um critério de partição.

Uma vez construído o modelo do Tangram-II, a cadeia de Markov associada já pode ser gerada, porém ela só pode ser solucionada quando os parâmetros forem substituídos por valores numéricos. Esses valores são fornecidos por um método de otimização baseado em AGs que, inicialmente, gera conjuntos de valores para os parâmetros de forma aleatória. Após a substituição dos parâmetros pelos valores numéricos o Tangram-II resolve a cadeia de Markov, calculando as probabilidades em estado estacionário de cada um dos seus estados. Estas probabilidades são utilizadas para o cálculo de medidas de interesse relacionadas com o desempenho do protocolo. Essas medidas são os *valores obtidos*. Cada protocolo a ser particionado tem uma especificação a ser atendida. Esta especificação pode ser dada, por exemplo, por uma curva relativa a uma determinada medida de interesse, que deve ser aproximada pelo AG. Essa curva fornece os *valores desejados* da função objetivo. De posse dos valores desejados e dos valores obtidos pode-se calcular o valor da função objetivo para cada um dos indivíduos da população gerada pelo AG, cujo tamanho é especificado pelo projetista. O projetista também especifica o número de pontos da curva que ele deseja aproximar. O valor da função objetivo relativo ao desempenho, que é a soma dos erros em relação a cada ponto especificado, fornece uma medida da *aptidão* do indivíduo para resolver o problema.

Nesta etapa inicia-se o processo de otimização, calculando-se a aptidão de cada indivíduo da população e verificando-se se algum destes indivíduos representa a solução do problema. Caso a solução não tenha sido encontrada, realiza-se uma seleção dos indivíduos, com maior probabilidade para os indivíduos mais aptos, faz-se o cruzamento de cada um com seu *parceiro* e gera-se uma nova população. O Tangram-II é realimentado com a nova *geração* de indivíduos e repete-se o processo até que se encontre um conjunto de taxas que atenda ao erro máximo especificado pelo projetista. A partir das taxas calculadas, realiza-se a partição do protocolo, que é avaliada através das medidas de área e atraso fornecidas pelas ferramentas Synopsys e ALTERA. Caso o custo do HW atenda à especificação o problema está resolvido, caso contrário uma nova otimização é realizada. Repare que o algoritmo genético otimiza a função objetivo para atender aos requisitos de desempenho, porém esta solução não é única. O requisito de custo do HW restringe o número de soluções possíveis. A partição HW/SW é determinada através do critério aplicado às taxas calculadas pelo AG, que será descrito na seção 3.3.

3.1.2 Implementações em HW

A especificação do protocolo em máquina de estados serve como referência para a entrada das descrições comportamentais em VHDL no Synopsys e no ALTERA. Cada transição de estado, contendo uma ou mais operações do protocolo, é associada a uma *entidade* em VHDL. Cada *entidade* é sintetizada, de modo a se obter as medidas de atraso e área referentes às transições de estado quando implementadas em HW.

A ferramenta Synopsys sintetiza um circuito lógico a partir da descrição comportamental de cada transição em VHDL. Esta implementação é realizada utilizando-se uma biblioteca de células padrão (*standard cells*) numa dada tecnologia. As etapas de alocação e roteamento de células e otimização são automáticas e se tornam "transparentes" para o projetista.

PLDs são circuitos integrados digitais que podem ser programados a fim de implementar funções complexas, isto é, PLDs podem implementar qualquer expressão booleana definida através de programação. A ferramenta ALTERA implementa as funções lógicas de cada transição a partir de suas descrições em VHDL.

Ao final do processo de síntese, obtém-se uma tabela contendo todas as medidas de atraso e custo de área para cada uma das transições do protocolo. Estas medidas serão utilizadas para avaliar se a partição escolhida atende à especificação de custo do HW fornecida pelo projetista e se os respectivos atrasos atendem às taxas calculadas pelo AG. Se a avaliação for positiva, o processo se encerra, caso contrário, uma nova otimização é realizada.

3.2 A Função Objetivo

Os diversos parâmetros utilizados para orientar a partição devem ser combinados numa função única, denominada *função objetivo*. Como os parâmetros especificados normalmente "competem entre si", é útil atribuir-se pesos a cada um dos parâmetros que compõem a função objetivo para equalizar as possíveis diferenças de sensibilidade entre eles. Esta função fornece uma medida da qualidade da partição em questão e através de sua avaliação pode-se selecionar aquela que satisfaz melhor os requisitos de desempenho. Para a escolha da melhor partição HW/SW devem ser definidos os parâmetros a serem utilizados na função objetivo. A função objetivo utilizada neste trabalho concentra-se em medidas de desempenho do protocolo (vazão, retardo, probabilidade de perda, etc), obtidas pelo Tangram-II, e possui a seguinte forma:

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

Onde:
x_{id} - são os parâmetros de desempenho desejados;
x_{io} - são os parâmetros de desempenho obtidos;
w_i - são os pesos atribuídos a cada parâmetro;

Os parâmetros de desempenho utilizados dependem dos requisitos de qualidade de serviço desejados. Deseja-se minimizar $F(P)$, onde P é uma partição que satisfaz um conjunto de restrições impostas pelo projetista.

3.3 O Critério de Partição

Para a automatização do particionamento é fundamental estabelecer um critério que relacione as taxas calculadas pelo AG com uma determinada partição HW/SW. Quanto maior a taxa de um determinado evento maior o número de vezes, por segundo, que um determinado conjunto de operações do protocolo é executado. Este fato poderia levar à conclusão (precipitada) de que essas operações são as que devem ser implementadas em HW. No entanto deve-se considerar, a partir dos resultados obtidos resolvendo-se a cadeia de Markov associada, as probabilidades em estado estacionário de cada estado do sistema. Ou seja, mesmo que um evento tenha uma alta taxa, a probabilidade do sistema estar num estado em que o mesmo está habilitado pode ser muito baixa.

A partir das considerações realizadas acima foi estabelecido como critério de partição HW/SW dos eventos, onde cada evento representa um conjunto de operações do protocolo, o produto $\lambda_i * \pi$, onde:

λ_i - taxa do evento i determinada pelo AG.

π - soma das probabilidades em estado estacionário dos estados nos quais o evento i está habilitado.

É como se cada evento tivesse uma "vazão" associada a ele. Desta forma, as operações do protocolo relacionadas com os eventos de maior "vazão" são implementadas em HW e aquelas associadas aos eventos de menor "vazão" são implementadas em SW.

A metodologia de otimização e o critério apresentados fornecem subsídios ao projetista para uma escolha objetiva da melhor partição, mantendo a análise em alto nível, sem a necessidade de um conhecimento detalhado no nível de circuitos. Consequentemente, o projetista pode tomar uma decisão consistente baseada em dados concretos e não apenas em sua experiência.

3.4 Algoritmo Genético Implementado

No algoritmo utilizado neste trabalho, as faixas iniciais de variação permitidas para cada taxa de cada evento do modelo, que são utilizadas para a geração da população inicial, são determinadas de acordo com a sensibilidade de cada uma em relação ao desempenho final desejado para o protocolo. Diz-se que o algoritmo utilizado é *híbrido*, pois mistura AGs com métodos de otimização tradicionais, que utilizam derivadas na determinação das faixas iniciais.

O algoritmo desenvolvido é um AG simples, onde o cromossomo foi codificado como um vetor de variáveis reais que representam os parâmetros (taxas dos eventos) a serem determinados. A dimensão do vetor depende do número de taxas que se deseja determinar, conforme o modelo construído para o protocolo em questão.

A população inicial é obtida por uma função que gera números reais aleatoriamente, utilizando um banco de sementes e o relógio do computador. Os valores numéricos correspondentes a cada indivíduo são transferidos pelo AG ao modelo de parâmetros do Tangram-II. Desta forma é possível executar um dos métodos numéricos de solução da cadeia de Markov existentes no Tangram-II através de uma rotina específica, que é chamada pelo AG, e determinar para cada indivíduo da população as probabilidades em estado estacionário da cadeia de Markov, obtendo-se as respectivas medidas de interesse para cada um destes indivíduos. A aptidão bruta é determinada através do valor da função objetivo, dado pela soma de erros entre os valores das medidas obtidas e os valores desejados, conforme a equação 1. Determinada a aptidão bruta de cada indivíduo, normaliza-se essa aptidão, realiza-se o processo de seleção, cruzamento, mutação e uma nova população é gerada, caso nenhum dos indivíduos seja a solução para o problema.

Quando o AG obtém os valores de taxas que satisfazem a especificação em termos de desempenho, determina-se a primeira partição HW/SW através do procedimento descrito na seção 3.3.

A estrutura do algoritmo possui uma complexidade polinomial (n^3), proporcional ao número de pontos da curva de desempenho que se deseja aproximar, ao tamanho da população utilizada e ao número de gerações especificadas.

4 Resultados

Esta seção tem como objetivo apresentar os resultados numéricos obtidos através da implementação computacional da metodologia proposta utilizando o protocolo TCP. São apresentados gráficos e tabelas que permitem avaliar a viabilidade da metodologia e do critério de partição HW/SW utilizado.

4.1 Estudo de Caso

Neste exemplo é analisado o desempenho do protocolo TCP quando em fase de controle de congestionamento, o "TCP Congestion Avoidance". Utilizando-se o modelo matemático e as suposições propostas em Mathis [19], pode-se aproximar a curva *cwnd* vs. *RTT* por um gráfico com a forma de um *dente de serra* como é mostrado na figura 4, onde *cwnd* é o tamanho da janela de congestionamento do emissor em número de pacotes e *RTT* é o *round trip time*.

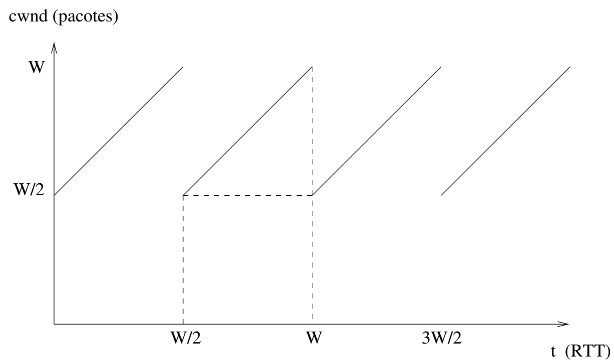

Figura 4: cwnd vs. RTT com perda periódica.

Analisando-se a figura 4 pode-se determinar [19] o tamanho da janela de congestionamento, em número de pacotes, em função da probabilidade de perda, a saber:

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

Onde BW é a banda passante, MSS é o tamanho máximo de segmento, C é uma constante de proporcionalidade e p é a probabilidade de perda. A determinação dos parâmetros do protocolo é realizada pelo AG aproximando a curva $(BW * RTT)/MSS$ vs. *perda*, traçada utilizando-se a equação 2, com $C = 1$, que pode ser visualizada na figura 5. A expressão $BW * RTT/MSS$ é uma **estimativa** do tamanho médio da janela de congestionamento [19]. Deve ser observado que o gráfico está em escala logarítmica.

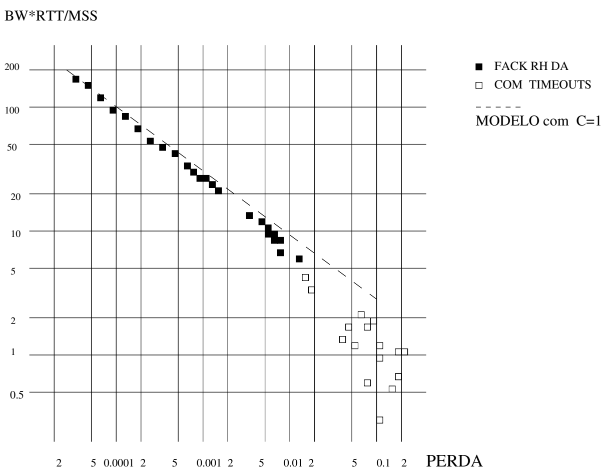

Figura 5: Curva Janela vs. *perda* para o modelo matemático dado pela equação 2.

4.2 Escolha da Partição

Foram utilizados 10 pontos para aproximar a curva do modelo e se obter os parâmetros desejados. Foram realizados experimentos com uma população de 100 indivíduos e utilizou-se como critério de parada do algoritmo genético o limite de 10 gerações. A taxa do evento PERDA (λ_8) foi variada (abscissa da curva) e a função objetivo foi calculada baseada na diferença entre o tamanho da janela obtido e o tamanho da janela desejado.

Obteve-se um erro mínimo para a função objetivo (considerando apenas o desempenho) igual a aproximadamente 1,7% para cada um dos pontos escolhidos. Os resultados obtidos são apresentados na tabela 1. O pequeno erro no valor das janelas, fornecido pelo AG, se deve ao valor da constante C, com perdas aleatórias seu valor é um pouco menor do que 1 [19].

Evento	Taxa (1/seg.)
Aumenta_jan (λ_1)	446,94
Envia_pct (λ_2)	1078,67
Envia_ack (λ_5)	984,02
Diminui_jan (λ_4)	46,17
Ack_ok (λ_3)	803,90
Envia_sinal_perda (λ_6)	116,41
Msg_ok (λ_7)	1

Tabela 1: Taxas para os eventos do TCP "Congestion Avoidance" obtidas pelo AG.

Observe que, pelo modelo, em média $1/p$ pacotes são enviados antes da ocorrência de uma perda. Utilizando-se as taxas da tabela 1, aplica-se o critério descrito na seção 3.3 e calcula-se o produto $\lambda_i * \pi$ para cada evento do protocolo. Os resultados obtidos são apresentados a seguir:

$$\lambda_1 * \pi_1 = 1,07 \quad (evento \;\; 1) \tag{3}$$

$$\lambda_2 * \pi_2 = 1,76 \quad (evento \;\; 2) \tag{4}$$

$$\lambda_3 * \pi_7 = 1,04 \quad (evento \;\; 3) \tag{5}$$

$$\lambda_4 * \pi_6 = 0,013 \quad (evento \;\; 4) \tag{6}$$

$$\lambda_5 * \pi_5 = 1,71 \quad (evento \;\; 5) \tag{7}$$

$$\lambda_6 * \pi_4 = 0,23 \quad (evento \;\; 6) \tag{8}$$

$$\lambda_7 * \pi_3 = 0,99 \quad (evento \;\; 7) \tag{9}$$

De acordo com os resultados acima, as operações do protocolo correspondentes aos eventos 2 e 5 devem ser implementadas em HW enquanto que aquelas correspondentes aos eventos 4 e 6 devem ser implementadas em SW. Com relação aos eventos 1, 3, 7, como os valores são próximos, deve-se analisar o modelo. Observa-se que os eventos 2, 3, 5 e 7 são aqueles que correspondem à transmissão de pacotes e ao recebimento de ACK's. É como se esses eventos correspondessem ao *caminho crítico do protocolo*, isto é, os eventos pelos quais o protocolo passa o maior número de vezes quando em estado estacionário.

Portanto, de acordo com o critério proposto, as operações do protocolo correspondentes aos eventos 2, 3, 5 e 7 devem ser implementadas em HW enquanto que aquelas correspondentes aos eventos 1, 4 e 6 devem ser implementadas em SW.

Esta é a primeira partição que será avaliada em conjunto com as medidas de atraso e custo de área fornecidas pelo Synopsys e pelo ALTERA. Caso o custo do HW esteja dentro das especificações e as taxas encontradas possam ser atendidas pelos circuitos sintetizados, uma solução foi encontrada. Caso contrário, uma nova população é gerada para que uma nova otimização seja realizada.

Após a determinação das taxas pelo AG, a taxa do evento **Perda** (λ_8), que tem distribuição exponencial, foi fixada em 0,01 e foi realizada uma simulação no Tangram-II onde foi feito um *trace* da variável que representa o tamanho da janela. O resultado pode ser visualizado na figura 6.

Observa-se através do gráfico da figura 6 e dos valores fornecidos pela simulação realizada no Tangram-II que a média de crescimento, em cada ciclo, do tamanho da janela de congestionamento, é aproximadamente igual ao crescimento da janela do modelo **ideal** da figura 4, para uma mesma probabilidade de perda. Quanto menor a probabilidade de perda, melhor é o resultado fornecido pelo AG. É justamente nestes casos (poucas perdas) que o modelo da equação 2 melhor se ajusta aos casos reais. Este resultado indica que o modelo foi bem construído, os valores das taxas encontradas estão coerentes e o seu comportamento se aproxima mais de um comportamento real do mecanismo “Congestion avoidance” encontrado nas diversas implementações do protocolo TCP.

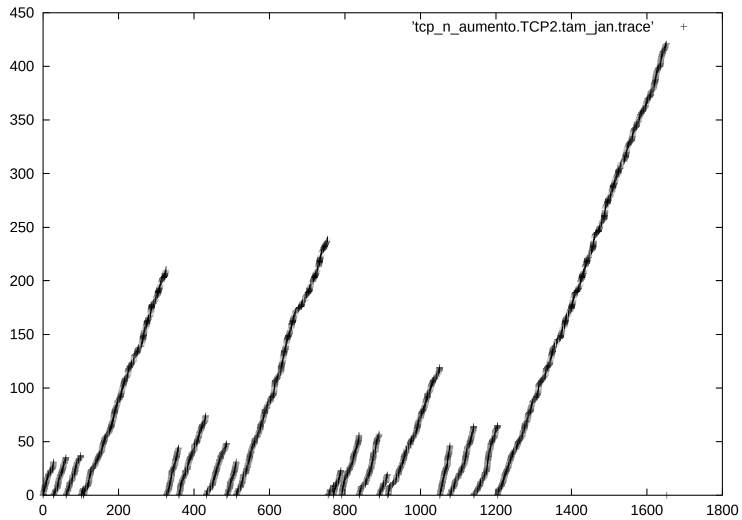

Figura 6: Evolução da Janela de congestionamento com probabilidade de perda com distribuição exponencial.

4.3 Avaliação da Partição

As descrições comportamentais em VHDL de cada transição de estado do protocolo são utilizadas como entrada para as ferramentas Synopsys e ALTERA. A síntese lógica dos circuitos no Synopsys foi realizada utilizando-se uma biblioteca de células padrão (*standard cells*) ES2 com tecnologia 0,7μm. A implementação dos circuitos lógicos no ALTERA foi realizada utilizando-se o compilador do ambiente no modo auto para a escolha do componente mais adequado ao projeto.

A tabela 2 mostra as medidas de atraso e custo de área obtidos para cada uma das transições de estado do protocolo. As medidas de atraso foram calculadas através da simulação das transições variando-se os valores do *clock* para o circuito sintetizado até se encontrar o menor valor de operação. Os valores de área são fornecidos automaticamente pelo Synopsys. Note que uma vez escolhido, o componente ALTERA possui uma área fixa independente do projeto, ou seja, não tem sentido a extração de medidas de área no projeto ALTERA.

Estas medidas mostram uma nítida vantagem em relação à velocidade de operação do projeto utilizando Synopsys em comparação com a implementação utilizando ALTERA. Isto se deve ao fato do Synopsys oferecer um maior grau de liberdade na alocação e roteamento de células, implicando num melhor compromisso entre área e velocidade. Por outro lado, deve ser notado que a implementação em PLDs é imediata e o sistema pode ser reprogramado sem a necessidade de uma nova rodada de fabricação, como no caso das *standard cells*.

Tendo em vista os resultados apresentados na tabela 2, a partição proposta na seção 4.2 é então avaliada. A partição sugeria que as transições 2, 3, 5 e 7 fossem implementadas em HW e que as transições 1, 4 e 6 fossem implementada em SW. Levando-se em conta, por exemplo, um custo de área especificado pelo projetista de 1 mm^2, pode-se observar que as transições 2 e 7 extrapolam os requisitos desejados no início do projeto. Sugere-se, então, que estas transições sejam implementadas em PLD. Apesar das medidas de

Transição	ALTERA Atraso	SYNOPSYS Área (mm^2)	Atraso
Primeira	17 ns	0,213	5,5 ns
Segunda	50 ns	8,456	10 ns
Terceira	7 ns	0,017	5 ns
Quarta	7 ns	0,194	5 ns
Quinta	7 ns	0,017	5 ns
Sexta	7 ns	0,017	5 ns
Setima	50 ns	8,456	10 ns

Tabela 2: Medidas de atraso e área das transições do protocolo.

atraso da implementação em PLD serem maiores do que em ASIC, elas atendem às taxas calculadas pelo AG. As transições 3 e 5 serão implementadas em ASIC, pois as medidas de área atendem ao exigido pelo projetista e as medidas de atraso atendem às taxas calculadas pelo AG.

4.4 Convergência do Algoritmo Genético

Para a obtenção dos resultados numéricos mostrados, foi desenvolvido um programa na linguagem de programação C, que implementa a metodologia proposta. Este programa foi executado em um microcomputador Pentium-II 400 Mhz com 128MB de memória RAM, no sistema operacional Linux.

O algoritmo convergiu em cerca de alguns minutos para o número de pontos, o tamanho da população e o número de gerações dados no exemplo acima. Para outros experimentos realizados, com populações de 5000 indivíduos, 10 pontos, 10 gerações e/ou um erro total de 0,0001 para a função objetivo, o algoritmo levou entre 2 a 4 horas para convergir.

O tempo de convergência indica que o programa da ferramenta Tangram-II que executa o método escolhido para a solução da cadeia de Markov (**solv**) é bastante rápido e apesar de poder ser chamado até centenas de milhares de vezes durante a execução do algoritmo genético, dependendo do tamanho da população, do número de pontos a serem aproximados e do número de gerações, o algoritmo é executado no máximo em algumas horas, no ambiente mencionado no início desta seção. Este é um tempo curto se considerarmos todo o processo de projeto e síntese do protocolo, principalmente pelo tempo economizado pelo projetista na escolha da melhor partição.

5 Considerações Finais

As redes de alta velocidade impuseram modificações no projeto de protocolos de comunicação. Um fator importante para o bom desempenho destes protocolos é a utilização integrada de *hardware* e *software* durante sua implementação. Devido à crescente demanda por protocolos com altas vazões, as soluções que utilizam HW são cada vez mais investigadas. Nas grandes empresas fabricantes de comutadores e roteadores para redes de alta velocidade, a tendência é a implementação de protocolos em ASIC's [1].

Atualmente, a escolha da melhor partição HW/SW de um protocolo de comunicação é muito subjetiva e o processo de implementação muito lento e mais sujeito a erros. Com esta finalidade a primeira contribuição deste trabalho foi o desenvolvimento de uma metodologia para a automatização do processo de escolha da partição no projeto de protocolos, baseada em requisitos de desempenho e custo. A utilização conjunta das técnicas descritas facilita o desenvolvimento desta metodologia e fornece subsídios para que o projetista tome decisões precisas baseado em medidas e critérios objetivos. Para que a metodologia fornecesse resultados satisfatórios, foi fundamental implementar a integração do AG com o Tangram-II.

Apesar de demandar um custo computacional um pouco elevado, o AG é simples de ser implementado, não é tendencioso e evita funções de erro mais complexas como, por exemplo, a função exponencial utilizada no método de otimização *Simulated Annealing* [20].

Os resultados descritos mostram que a metodologia apresentada é viável e que o critério de partição adotado é coerente. Em relação à abrangência das aplicações, ainda existe um grande potencial a ser explorado. Protocolos que possuem um alto grau de processamento nos nós de uma rede e cujo desempenho está diretamente relacionado à velocidade desse processamento são os que mais se aproveitam desta metodologia como, por exemplo, os protocolos de recuperação de erros para transmissão *Multicast* confiável [21] e protocolos relacionados com criptografia de dados [22].

Agradecimentos

Agradecemos ao Prof. Edmundo Souza e Silva, coordenador do laboratório LAND da COPPE/UFRJ, e à toda sua equipe, assim como à equipe do laboratório LPC da COPPE/UFRJ por ter fornecido todo o apoio em termos de recursos coputacionais, suporte técnico e constante troca de idéias relativas à utilização das ferramentas Tangram-II, Synopsys e ALTERA. Agradecemos também ao Prof. José Ferreira de Rezende pela constante troca de idéias e contribuições.

Referências

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

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

[3] S. Fischer, J. Wytrebowicz e 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", in *3th International Workshop on Hardware/Software Codesign*, pp. 17–24, 1994.

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

[6] A. Kalavade e 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 e A. Haxthausen, “Lycos: The lyngby co-synthesis system”, in *Design Automation for Embedded System*, 1997.

[8] P. H. Chou, R. B. Ortega e 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*, pp. 32–35, setembro de 1999.

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

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

[12] A. P. C. Silva, “Tangram-ii user’s manual”, relatório técnico, Universidade Federal do Rio de Janeiro, outubro de 2000. `http://www.land.ufrj.br`.

[13] IEEE Std 1076-1987, *IEEE Standard VHDL Language Reference Manual*, março de 1988.

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

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

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

[17] H. Horner, “A C++ class library for genetic programming: The Vienna University of Economics - genetic programming kernel”, *Internet Draft*, maio de 1996. `http://www.wu-wien.ac.at/usr/h88/h8850092`.

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

[19] M. Mathis, J. Semke e J. Mahdavi, “The macroscopic behavior of the tcp congestion avoidance algorithm”, *Computer Communication Review*, vol. 27, no. 3, pp. 1–16, julho de 1997.

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

[21] J. Nonnenmacher, E. Biersack e D. Towsley, “Parity based loss recovery for reliable multicast transmission”, in *Proceedings of ACM SIGCOMM’97*, 1997.

[22] M. Abadi, “Security protocols and specifications”, in *In Foundations of Software Science and Computation Structures: Second International Conference, FOSSACS ’99*, 1999.