

Análise e Seleção de Redes de Interconexão para Síntese de Sistemas no Ambiente S³E²S

Márcio E. Kreutz, Luigi Carro, Cesar A. Zeferino e Altamiro A. Susin
{kreutz, carro, zeferino}@inf.ufrgs.br}, {susin}@iee.ufrgs.br

Resumo

A seleção e síntese de Redes de Interconexão cresce em importância a medida que são implementados sistemas cada vez mais complexos, compostos por vários núcleos de processamento (cores), com comportamento heterogêneo. Em muitas situações, a arquitetura de comunicação adotada para um sistema pode ser fator determinante para a adequação do sistema às restrições de projeto. O presente trabalho propõe-se à definição e implementação de algoritmos para analisar e selecionar arquiteturas de redes de interconexão que satisfaçam as restrições de projeto, para a implementação de sistemas complexos.

Abstract

The Interconnection Networks analysis and synthesis process grows in importance since we are moving towards more complex systems, made by several processing elements (cores) with heterogeneous behavior. In many cases, the choose for an interconnection network architecture can be the most crucial factor to attend the design constraints. This work aims to define and implement algorithms devoted to analyse and select those Networks that better match the constraints defined for the user to a System.

1. Introdução

Sistemas complexos caracterizam-se por executarem funções com diferentes características e restrições de projeto, delegando ao processo de síntese a seleção de componentes microarquiteturais específicos à execução de cada função. Assim, um sistema poderá ser constituído, quando de sua implementação, por componentes tais como ASICs, processadores de sinais, microcontroladores, etc. Esses componentes podem constituir-se por elementos completos de processamento, também chamados de *cores* ou núcleos de processamento, o que implica em uma síntese cuja entrada especifica o sistema em um alto nível de abstração. Desse modo, a implementação do um sistema completo será efetivada através da configuração e conexão entre diferentes *cores*.

O ambiente S³E²S (Specification, Simulation and Synthesis of Embedded Electronic Systems) [CAR00] destina-se à especificação, simulação e síntese de sistemas embarcados. Para a síntese é adotado um estilo que se baseia na seleção de diferentes processadores (*cores*, *blocos IP*) para a execução das funções da aplicação com características diferenciadas.

Devido ao fato de as funções possuírem características diferentes, o resultado do processo de síntese em S³E²S resulta em um sistema completo e distribuído, onde processadores com características próprias são selecionados para a execução das diferentes funções.

Assim sendo, torna-se necessário também a inclusão no ambiente, de um processo de síntese de comunicação entre os processadores, pelos motivos expostos acima. Uma vez que tem-se a síntese dos nodos de processamento (neste caso, processadores) e do sistema de comunicação, a integração de ambos processos leva a implementação de um sistema completo, objetivo final dessa abordagem.

Dentro deste contexto, o presente trabalho propõe-se a analisar diferentes arquiteturas de redes de interconexão, a fim de selecioná-las para a síntese da comunicação entre os processadores escalonados para a execução de uma aplicação distribuída, especificada e simulada no ambiente S³E²S.

A análise será realizada a partir de primitivas de alto nível para comunicação, do tipo *send* (envia) e *receive* (recebe), e utilizará a biblioteca de simulação do ambiente. Portanto, o processo de análise dos sistemas de comunicação aqui proposto, se realizará com base em resultados de simulação, ao contrário da síntese baseada na seleção de processadores que é realizada estaticamente.

Dentre as variáveis utilizadas para a análise sobre as arquiteturas de redes de interconexão, destacam-se [ZEF99]:

1. Topologia das redes;
2. Chaveamento;
3. Roteamento; e
4. Fluxo de controle.

A topologia da rede informa quais as capacidades da rede em relação às trocas de mensagens. Por exemplo, se um nodo freqüentemente troca mensagens com outro, torna-se desejável que não hajam muitos outros nodos no caminho de comunicação entre eles. Dependendo do número de nodos e das trocas mensagens, a topologia da rede pode ter grande influência no desempenho das funções que trocam mensagens. A topologia pode ser estática ou dinâmica [ASH98]. Para este trabalho serão consideradas topologias estáticas, visto que a análise será realizada para a seleção de uma arquitetura de rede de interconexão. A topologia dinâmica pode ser obtida após a seleção, se um mecanismo de avaliação das trocas de mensagens em tempo de execução estiver presente.

O chaveamento é o mecanismo pelo qual uma mensagem é removida de um canal de entrada e colocada em um canal de saída [NI93]. A latência da rede é muito dependente da abordagem de chaveamento utilizada. Neste trabalho serão analisados os mecanismos de chaveamento por circuito e por pacotes.

O roteamento é o método utilizado por uma mensagem para escolher um caminho para o envio de uma mensagem na rede. Existem vários métodos de roteamento tais como fonte, baseado em tabela, distribuídos, etc. [ASH98] [NI93]. Em um primeiro momento, neste trabalho, será abordado o roteamento fonte, onde o nodo que envia a mensagem (fonte) escolhe todo o caminho entre ele e o destino antes de enviar a mensagem. Em seguida pretende-se ampliar a análise para levar em consideração outros métodos de roteamento citados acima.

O fluxo de controle serve para decidir o que fazer com uma mensagem quando esta não pode ser transmitida no tempo em que estava prevista. Isso ocorre devido ao fato que as mensagens que circulam pela rede competem por recursos, tais como *buffers* e canais de comunicação. Como esse recursos são limitados, em certas ocasiões podem haver colisões de recursos, como, por exemplo, duas mensagens competindo um determinado canal. Neste trabalho, o fluxo de controle será tratado de acordo com a rede sendo analisada, como explicado na seção 4.

Por meio da análise realizada sobre as redes de interconexão, através das restrições impostas pelo projetista, torna-se possível selecionar a arquitetura e os protocolos de comunicação que melhor otimizam as trocas de mensagens de uma aplicação especificada, simulada e implementada no ambiente S³E²S.

A seção seguinte mostra como são especificados os sistemas no ambiente S³E²S. A seção 3 explicita os níveis de abstração utilizados para os protocolos de comunicação, enquanto que na seção 4 são mostrados os algoritmos para análise e seleção das Arquiteturas de Interconexão utilizados nesta abordagem. Os resultados obtidos até o momento são mostrados na seção 5. Finalmente, na seção 6 são apresentadas algumas conclusões e trabalhos futuros.

2. Especificação de Sistemas no Ambiente S³E²S

O ambiente S³E²S permite a uma aplicação ser especificada, simulada, e sintetizada criando assim um caminho que permite a continuidade do modelo desde a sua concepção até a implementação.

Para a especificação da aplicação, o ambiente utiliza a ferramenta SIMOO MET (Model Editor Tool) [WAG99]. Com essa ferramenta, uma aplicação pode ser especificada através do paradigma de orientação a objetos, utilizando a linguagem C++. Na Figura 1 é mostrado um exemplo com duas classes (*A* e *B*) conectadas através do canal de comunicação denotado pelas mensagens *msg1* e *msg2*. Dessa forma, a aplicação naturalmente será modelada como um sistema distribuído, onde as entidades componentes (classes) possuem memória local (atributos) e comunicam-se por trocas de mensagens.

Assume-se, então, que a forma como o projetista modela a aplicação constitui-se da primeira partição entre os seus componentes. Assim, ao modelar a aplicação, o projetista estará distribuindo as funcionalidades e determinando quais objetos que comunicam-se e de que forma (ponto a ponto, *broadcast*).

Seguindo o paradigma de orientação a objetos, o ambiente S³E²S permite especificar a aplicação em qualquer nível de abstração. Dentro do escopo do presente trabalho, está-se assumindo que a especificação será em nível de sistema, sendo a granularidade especificada pelas classes C++ definidas através da ferramenta SIMOO MET. Seguindo essa definição, a síntese dos componentes da aplicação baseada na seleção de processadores [CAR00] [WAG99] é realizada de modo a mapear cada classe para o processador que melhor se adapta às características de cada função.

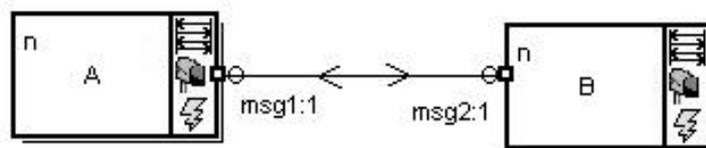


Figura 1. Modelagem na ferramenta SIMOO MET.

Em relação à arquitetura de comunicação, a especificação ao nível de sistema representa o mais alto nível de abstração sobre a qual o sistema de comunicação pode ser especificado. Isto equivale a dizer que, nesse nível, o projetista modela a comunicação de acordo com a semântica presente nas comunicações entre os componentes da aplicação. Sendo assim, não são levadas em consideração variáveis como a topologia da rede de interconexão, largura de banda, sincronismo, etc. O que é relevante nesse nível de abstração é a definição de quais componentes trocam mensagens, o tamanho da mensagem e o tempo de transmissão permitido. As análises realizadas neste trabalho estarão focadas nesse nível, uma vez que espera-se obter a semântica de comunicação de todo o sistema a fim de tornar possível, em conjunto com a seleção de processadores, a síntese de um sistema completo. Além disso, a análise realizada nesse nível de abstração leva a uma redução do tempo de simulação e análise, por omitir muitos dos detalhes presentes em níveis de abstração inferiores, como, por exemplo, tipos de empacotamento.

Como a especificação do sistema é realizada dentro do paradigma de orientação a objetos, torna-se natural a extensão da descrição do sistema em direção a níveis mais baixos de abstração. Isso cria um caminho descendente em relação aos níveis de abstração, o que permite chegar-se a síntese da aplicação, partindo-se de uma especificação em nível de sistema. Dessa forma, obtém-se uma característica muito desejável para as ferramentas de análise e síntese, que é a representação da continuidade do modelo. Além disso, a ferramenta torna-se facilmente extensível para permitir análises mais apuradas e, finalmente, sintetizar as arquiteturas de comunicação selecionadas para a aplicação.

Na fase atual do projeto, são realizadas análises ao nível de abstração de Sistema. Entretanto, estão previstas análises a serem realizadas em níveis de abstração inferiores, os quais podem ser derivados diretamente das classes descritas ao nível de Sistema.

A implementação do sistema de comunicação será baseado na seleção em uma biblioteca de comunicação orientada a objetos especificada em diferentes níveis de abstração, como descrito acima. O protocolo e a arquitetura de comunicação serão implementados em função do nível de abstração dos componentes selecionados para a execução de cada função do sistema. Por exemplo, quando um processador for selecionado à execução de uma função, pode-se implementar o protocolo através de uma função em software, que por sua vez pode comunicar-se com um ASIC através da implementação VHDL (presente no componente) da arquitetura de comunicação.

3. Níveis de Abstração em Protocolos de Comunicação

Conforme definido no modelo OSI [TAN96], um protocolo de comunicação pode ser especificado em até sete níveis de abstração, também chamados de camadas. Essas camadas especificam o serviço de comunicação implementado, bem como as operações executadas em cada nível relativo ao nível de abstração em que a comunicação é realizada.

Entretanto, pode-se pensar em uma abordagem para especificação da comunicação em sistemas que definem três níveis de abstração diferentes. Esses três níveis por sua vez, podem ser decompostos nos sete citados acima.

Assim, pode-se pensar em um sistema de comunicação através dos três níveis de abstração a seguir:

1. *Aplicação*: neste nível estão especificadas as características de comunicação de uma aplicação distribuída que podem ser definidas em relação a:
 - *restrições de projeto*: neste item são especificadas as restrições impostas pelo projetista para a aplicação. Como exemplo pode-se citar restrições temporais, onde especificam-se o tempo permitido para as trocas de mensagens entre as funções da aplicação; e restrições espaciais, onde especifica-se a área permitida para a implementação dos barramentos ou redes de interconexão;
 - *escalonamento das operações*: diz respeito à execução das funções da aplicação no tempo, de forma a estabelecer-se quais executarão em seqüência e quais em paralelo. A indicação do escalonamento pode ser obtida do sincronismo entre as mensagens.

Na abordagem utilizada neste trabalho, a definição do sistema comunicação neste nível dar-se-á pelas primitivas de comunicação da ferramenta, como, por exemplo, *Send*, *Receive*, etc.

2. *Protocolo*: neste nível define-se qual o protocolo de comunicação a ser utilizado, bem com as suas características através das variáveis que indicam sincronismo e frequência de trocas de mensagens. Assim, o protocolo será definido em termos de sincronização, abrangência da comunicação (*Broadcast*, *Multicast* e *RPC - Remote Procedure Call*), controle da comunicação (fluxo de controle, chaveamento e roteamento), tamanho dos *buffers*, tipo de empacotamento, etc. Para o protocolo também são definidos os *drivers* que realizam a *interface* entre o elemento de processamento e a arquitetura de comunicação. No presente trabalho, em um primeiro momento, a análise será realizada levando em consideração apenas comunicações do tipo *RPC*.
3. *Arquitetura de Interconexão*: neste nível estão especificadas as características do meio de comunicação. Dentre as características pode-se citar a capacidade de transmissão de dados em relação ao número de bits por segundo (*bandwith*), a topologia de interconexão, a homogeneidade do tempo de troca de dados entre os nodos, o controle de acesso aos recursos (ex.: árbitros em barramentos, FIFO), tamanho do canal de comunicação (em bits), determinismo no tempo de transmissão, o sincronismo entre as mensagens, a capacidade de acesso simultâneo dos nodos ao caminho de dados e o tipo de chaveamento e roteamento utilizados. Percebe-se claramente que a definição do sistema de comunicação neste nível de abstração é função das variáveis discutidas na seção anterior.

Pode-se perceber que os três níveis citados estão relacionados diretamente, pois constituem-se em três níveis diferentes de abstração que definem o mesmo sistema de comunicação. Assim, um protocolo é definido em relação às características de comunicação da aplicação enquanto que necessita de uma arquitetura de comunicação adequada para satisfazer essas características. Por sua vez, o protocolo pode também adequar-se às características da arquitetura de comunicação como, por exemplo, utilizar diferentes políticas de empacotamento das mensagens através do seu *driver*, para adequar-se à capacidade de transmissão do meio de comunicação e implementar as políticas de chaveamento e roteamento estabelecidas pelas diferentes arquiteturas de comunicação.

4. Análise e Seleção de Arquiteturas de Redes de Interconexão

Nesta seção são apresentadas as três arquiteturas de redes de interconexão implementadas até o momento neste projeto, bem como a sua análise em relação às variáveis que as definem e aos níveis de abstração dos sistemas de comunicação apresentados nas seções precedentes.

O algoritmo para a análise do sistema de comunicação pode ser definido em relação à duas operações principais:

1. Obtenção da estrutura de comunicação da aplicação a partir das primitivas de alto nível definidas pela ferramenta; e
2. Análise e Seleção de uma arquitetura de rede de interconexão dentre as implementadas, em relação às restrições de projeto.

Na primeira etapa, busca-se descobrir através das primitivas de comunicação definidas em alto nível pelo ambiente, qual a arquitetura de comunicação da aplicação definida pelas trocas de mensagens entre os seus componentes; enquanto que na segunda etapa busca-se analisar e verificar qual dentre essas arquiteturas é a que melhor atende as restrições de projeto. Nas análises realizadas até o presente momento, foi considerada a adequação das redes em relação ao desempenho.

Durante o processo de análise, os protocolos e as arquiteturas de comunicação podem ser analisadas sobre diferentes níveis de abstração, o que implica em entrar ou não em outras camadas do modelo OSI [TAN96]. Por exemplo, em um nível alto de abstração, uma rede em barramento pode ser analisada pela sua capacidade de transmissão em relação ao número de dados a serem transmitidos num determinado tempo; enquanto que num nível de abstração menor, podem ser analisadas diferentes configurações de empacotamento das mensagens.

Isso é possível devido ao fato de a modelagem do sistema permitir uma fácil extensão no sentido de aumentar ou diminuir o nível de detalhamento sobre o qual deseja-se realizar a análise e seleção. Até o momento, os protocolos estão sendo analisados levando-se em consideração os três níveis de abstração explicitados na seção anterior.

Verifica-se assim, que o diferencial desta abordagem está na realização da análise e seleção das arquiteturas de comunicação ao nível do sistema (a partir das primitivas de alto nível), onde podem ser analisadas diferentes arquiteturas de comunicação e suas características, bem como diferentes protocolos de comunicação. Assim, para uma determinada aplicação, pode-se escolher diferentes protocolos e arquiteturas, o que vem a refletir uma característica de uma aplicação com diferentes necessidades de comunicação entre as suas funções. Outra característica interessante constitui-se da fácil extensão do modelo para realização de análises em diferentes níveis de abstração, devido à modelagem Orientada a Objetos.

4.1 Redes de Interconexão analisadas e semântica das Variáveis de Análise

O processo de análise foi até o momento realizado sobre três arquiteturas de redes de interconexão:

1. Barramento;
2. Árvore binária; e
3. Grelha (*mesh*).

A escolha dessas três arquiteturas justifica-se por possuírem topologias bem diferenciadas entre si e pela importância de uso. A arquitetura de barramento por exemplo, é grandemente difundida e largamente utilizada para a interconexão dos componentes em sistemas integrados em um único chip (SOCs - *Systems On Chip*).

A Tabela 1 mostra a semântica das variáveis de análises empregadas para cada uma das arquiteturas analisadas.

Tabela 1. Semântica das variáveis de análise

<i>Arquitetura (Topologia)</i>	<i>Fluxo de Controle</i>	<i>Chaveamento</i>	<i>Roteamento</i>
Barramento	Daisy chain	-	-
Árvore	-	Pacotes	Estrutura da árvore
Grelha	Tamanho dos <i>buffers</i>	Pacotes e Circuito	Fonte (X, Y)

Para todas as arquiteturas foi assumido que os *buffers* possuem capacidade para armazenar completamente uma mensagem bloqueada.

O barramento, por tratar-se de uma estrutura com apenas um caminho de dados para todos os componentes do sistema, não possui políticas de Chaveamento e Roteamento. O fluxo de controle adotado para o barramento foi o *Daisy Chain* [HAM96], que define a prioridade com base na ordem de conexão dos componentes em relação ao mestre. Por exemplo, se numa estrutura como a ilustrada na Figura 2 o processador 1 (P1B) for o mestre, a prioridade no atendimento às mensagens será 2, 3 e 4; ou seja, se os processadores P2B e P3B desejarem enviar uma mensagem simultaneamente pelo barramento, o processador P2B a envia primeiro por possuir prioridade maior. A classe *Bus_w* representa o caminho de dados do barramento.

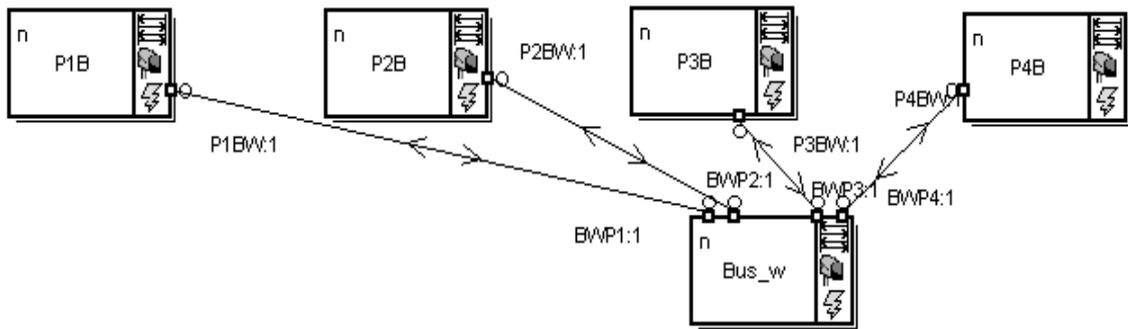


Figura 2. Arquitetura de Barramento

Para a arquitetura em Grelha, foi implementado um fluxo de controle definido pelo tamanho dos *buffers*. Dessa forma, quando um componente desejar enviar uma mensagem através de um roteador que esteja com seu *buffer* cheio, essa mensagem terá o seu envio atrasado pelo tempo necessário para que uma mensagem seja enviada e conseqüentemente, libere espaço no *buffer*. Até o momento, não foi implementado um controle de fluxo para a árvore, de forma que está-se assumindo que a Árvore possui *buffers* sem restrições de tamanho.

Em relação ao chaveamento, para a rede em Grelha, foram implementados duas políticas, por pacotes e por circuito. No chaveamento por circuito todos os roteadores no caminho de dados entre um componente fonte e um destino são alocados para o envio da mensagem, de modo que o tempo computado para cada roteador na análise corresponde ao tempo total do envio da mensagem. No chaveamento por pacotes, à cada roteador, é computado apenas o tempo que este leva para transmitir a mensagem. Esse tempo pode variar em função do tamanho dos *buffers*, como explicado acima.

Finalmente, a política de roteamento empregada para a Árvore corresponde à própria estrutura/topologia de interconexão entre os seus roteadores. Através dessa política de roteamento, os nodos no nível 2 podem facilmente tornarem-se o gargalo nas transmissões de mensagens se ocorrerem muitas trocas de mensagens entre componentes conectados à roteadores diferentes. No entanto, essa é uma das características desse tipo de interconexão. A rede em Grelha, por sua vez, emprega o roteamento estático (fonte) do tipo XY, ou seja, cada componente ao enviar uma mensagem, determina por quais roteadores esta vai ser transmitida em relação à sua distância em X (linha) e em Y (coluna) do componente destino.

4.2 O Processo de Análise e a Aplicação alvo

A análise das arquiteturas de interconexão é realizada sobre a disputa por recursos espaciais (*buffers*) e temporais (tempo de transmissão dos canais), tendo como parâmetro de controle e avaliação o número de mensagens sendo enviadas simultaneamente e as variáveis determinadas na seção anterior. A simultaneidade das mensagens é determinada pelos tempos de simulação. Assim, à cada tempo de simulação, é determinado o tempo de execução em cada roteador envolvido na transmissão (para as redes em Árvore e em Grelha) e no canal de comunicação do barramento, para cada mensagem transmitida.

Dessa forma, torna-se possível avaliar a capacidade temporal de transmissão das arquiteturas analisadas. A disputa pelos recursos espaciais (*buffers*) serve também como parâmetro para a avaliação temporal (no momento, apenas da rede em Grelha), pois implementam o fluxo de controle como explicado na seção anterior. Como passo seguinte da análise sobre as arquiteturas das redes de interconexão, pretende-se avaliar também as restrições de área e potência e o seu impacto, em conjunto com os resultados da análise temporal aqui apresentadas, na arquitetura do sistema completo e no reescalonamento de funções.

Durante o processo de análise, algumas otimizações podem ser realizadas a fim de diminuir o tempo de transmissão das mensagens. As otimizações serão feitas com base no impacto temporal causado nas arquiteturas de interconexão pela semântica de comunicação de um sistema. Até o momento, apenas uma otimização foi implementada, na rede em Árvore.

Para esta rede, como dito acima, o gargalo nas comunicações está no nível 2, uma vez que todas as mensagens que são trocadas entre nodos de roteadores diferentes passam por este nível. Levando-se em consideração essa característica da rede em Árvore, foi implementada uma otimização baseada na heurística que permuta dois componentes folhas conectadas à roteadores diferentes, sempre que o nível 2 possuir mais do que

duas mensagens sendo transmitidas simultaneamente. Os componentes que são permutados são aqueles que estiverem trocando uma mensagem quando esta condição for atingida.

Para a realização da análise, foi escolhida como aplicação alvo a Transformada Rápida de Fourier (FFT, *Fast Fourier Transform*) [QUI94] de 16 pontos, sendo 4 tratados em cada processador. Isto leva a um sistema distribuído composto por 4 processadores. Dessa forma, as redes avaliadas serão compostas por 4 nodos cada uma.

A FFT mostra-se interessante para avaliar o processo de análise aqui proposto, uma vez que possui um modelo de comunicação do tipo borboleta (*butterfly*). Nesse tipo de comunicação, todos os componentes trocam mensagens com todos simultaneamente, de forma que considera-se essa característica desejável para a avaliação das arquiteturas de interconexão implementadas. Isto por que várias mensagens estarão, num dado momento, utilizando simultaneamente vários roteadores das redes.

A fórmula utilizada para a avaliação temporal das redes de interconexão leva em consideração o tempo de transmissão, o tamanho de cada canal de comunicação e o tamanho e tempo permitidos para a transmissão de cada mensagem. Para cada tempo de simulação, calcula-se o tempo total de transmissão para cada roteador das redes em *Árvore* e em *Grelha* e para o canal de comunicação do *Barramento*, através da fórmula:

$$tempo_total_rede [n] \leftarrow tempo_rede[m] \times (tamanho_msg / tamanho_rede[m])$$

onde:

$tempo_total_rede[n]$	tempo total dos n roteadores, genericamente expresso em unidades de tempo (u.t.);
$tempo_rede[m]$	tempo de transmissão de uma palavra em um canal de comunicação da rede m ;
$tamanho_msg$	tamanho da mensagem sendo transmitida.
$tamanho_rede[m]$	tamanho do canal de comunicação de cada rede m , em <i>bits</i> ; e

Portanto, o que está se calculando é o número de palavras a serem transmitidas pelo tempo de transmissão de cada palavra nos canais de comunicação e cada rede de interconexão. Dessa forma, o cálculo realizado pela avaliação temporal é realizado em um alto nível de abstração, como o utilizado em [DAV97]. Outra alternativa seria implementar um algoritmo para analisar o tempo das trocas de mensagens em um nível de abstração mais baixo, levando em consideração por exemplo, o empacotamento das mensagens, como em [KNU99].

5. Resultados

Tendo em vista os objetivos propostos para a ferramenta, espera-se como resultado do processo de análise, a avaliação temporal das arquiteturas de interconexão, bem como a seleção da que melhor atender às restrições (temporais) impostas pelo projetista, com base nessa avaliação.

A avaliação temporal será realizada através da análise do tempo necessário para cada roteador (das redes em *Árvore* e em *Grelha*) ou para o canal de comunicação do *barramento*, transmitir as mensagens em cada tempo de simulação. Assim, como saída do processo de análise, tem-se:

- Uma indicação de quando uma arquitetura não suportar as trocas de mensagens simultâneas no tempo especificado;
- A distribuição temporal dos roteadores em cada tempo de simulação;
- A avaliação em relação às otimizações realizadas; e
- A seleção da arquitetura com o melhor desempenho para a aplicação analisada.

Para a validação dos algoritmos desenvolvidos para a análise das redes de interconexão, foram configuradas duas variações temporais para a aplicação e as redes.

Na primeira configuração foi definido um tempo de transmissão igual a 1 ($tempo_rede[m] = 1$) para cada canal de comunicação e tamanho do canal igual a 5 ($tamanho_rede[m] = 5$). O tempo permitido para cada

mensagem transmitida foi definido como sendo igual a 100 *u.t.* e o tamanho das mensagens igual a 10 (*tamanho_msg* = 10). Portanto, cada mensagem é transmitida através de 2 palavras sobre cada canal de comunicação. Durante a sua execução, o algoritmo FFT realiza três vezes uma comunicação do tipo troca completa (todos com todos). Portanto são realizadas 12 (3×4) trocas de mensagens simultâneas em cada tempo de simulação. Como citado anteriormente, a estrutura de troca de mensagens do tipo borboleta implementada pelo algoritmo FFT permite a utilização de todos os roteadores durante a execução da aplicação, o que vem a caracterizar melhor a avaliação temporal realizada pelos algoritmos de análise.

A tabela 2 mostra os resultados obtidos em termos do tempo necessário para a execução de todas as trocas de mensagens da aplicação. Esse tempo corresponde portanto, ao tempo total de todos os roteadores das redes.

Tabela 2. Tempos totais de transmissão para cada rede avaliada

<i>Arquitetura de Interconexão</i>	<i>Tempo total de execução</i>
Barramento	12
Árvore	34
Grelha (circuito)	134
Grelha (pacotes)	36

De acordo com os resultados da Tabela 2, pode-se verificar que o chaveamento por circuito empregado na rede em Grelha, consome muito tempo devido a sua característica de, para cada mensagem transmitida, alocar todos os roteadores, desde o fonte até o destino da mensagem. Como o algoritmo FFT realiza a troca de mensagens de todos processadores para todos, o tempo necessário para alocar todos os roteadores no caminho da mensagem para cada mensagem, cresce exponencialmente. Portanto, para essa aplicação, o chaveamento por circuito mostra-se desfavorável. Por outro lado, o roteamento por pacotes possui um resultado bastante satisfatório.

A rede em *Árvore* possui um bom desempenho para essa aplicação por que as trocas de mensagens ocorrem principalmente entre processadores conectados ao mesmo roteador, o que vem a contornar o gargalo desse tipo de rede que é o roteamento através do nível 2.

Finalmente, verifica-se que o tempo do Barramento corresponde exatamente à execução seqüencial nas trocas de mensagens, uma vez que são trocadas 12 mensagens com tempo de transmissão unitário (*tempo_rede[m]* = 1, para todo *m*) e o Barramento possui apenas um canal de comunicação. O fato de o Barramento, mesmo trocando as mensagens seqüencialmente possuir o melhor desempenho (e portanto, o escolhido para esta aplicação) significa que, quando há muitas trocas de mensagens entre todos os roteadores da rede, o tempo dos roteadores pode ser mais significativo do que enviar as mensagens seqüencialmente sobre um único canal de comunicação. Isso ocorre porque, em muitas situações, cada mensagem deve passar sobre vários roteadores até chegar ao seu destino e esse tempo pode ser maior que o tempo que cada processador conectado ao barramento espera pela licença de escrita por parte do mestre.

Na segunda configuração temporal, aumentou-se o tempo de transmissão de cada canal de comunicação da rede Grelha para 10 *u.t.* (*tempo_rede[m]* = 10, sendo *m* = *grelha*), de modo a verificar se o algoritmo detectasse a capacidade de transmissão dessa rede para o algoritmo FFT. De fato, pôde-se comprovar que, com esse tempo de transmissão para cada canal de comunicação da rede Grelha, esta não pôde mais ser utilizada para a transmissão das mensagens com restrição temporal de 100 *u.t.*, para cada tempo de simulação. O algoritmo de análise detectou o estouro do tempo limite para os dois tipos de chaveamento. Já no primeiro tempo de simulação, a implementação com chaveamento por circuito consumiu 440 *u.t.* para a transmissão simultânea das quatro mensagens, enquanto que a implementação com chaveamento por pacotes consumiu 140 *u.t.* para a transmissão.

Para verificar a otimização realizada na *Árvore*, foi realizada uma segunda experiência, onde proposadamente as trocas de mensagens no passo 1 de execução do algoritmo ocorrem apenas entre os processadores 2 e 3 que estão conectados à roteadores diferentes. Portanto, mesmo descaracterizando a configuração original das trocas de mensagens do algoritmo FFT, pôde-se verificar o impacto causado pela utilização do nível 2 na *Árvore* e pela otimização realizada, que permuta os nodos sempre que o nível 2 é utilizado mais do que duas vezes em cada tempo de simulação. Nesse caso, como ocorrem quatro trocas de mensagens em cada tempo de simulação e essas trocas são realizadas somente entre os processadores 2 e 3, a

otimização é efetivamente utilizada pelo algoritmo de análise. Os resultados obtidos podem ser verificados na Tabela 3.

Tabela 3. Tempos totais de transmissão com otimização para a Rede Árvore

<i>Arquitetura de Interconexão</i>	<i>Tempo total de execução</i>
Barramento	16
Árvore	40
Grelha (circuito)	76
Grelha (pacotes)	28

Como pode-se verificar na Tabela 3, aconteceram mudanças em todos os tempos totais de transmissão, uma vez que foi criado um novo padrão de comunicação. É interessante notar como diminui o tempo de transmissão da rede Grelha com chaveamento por circuito. Isso ocorreu por que os processadores 2 e 3 estão diretamente conectados (nodos vizinhos), o que diminui o tempo para as trocas de mensagens, já que o "circuito" entre eles corresponde à apenas um canal de comunicação.

Finalmente pode-se perceber que a rede Árvore teve um acréscimo de apenas 6 *u.t.* no tempo total de transmissão, sendo que esperava-se muito mais, já que quatro mensagens são agora trocadas simultaneamente utilizando o roteador no nível 2. Como quatro mensagens com tempo 1 *u.t.* passam por três roteadores simultaneamente, deveria haver um aumento de 12 *u.t.* no tempo de transmissão total em relação à execução do algoritmo original. No entanto, houve um aumento de apenas 6 *u.t.*, o que comprova a eficiência da otimização proposta, já que as duas últimas trocas de mensagens ocorrem em nodos vizinhos, com tempo unitário, e não 3 *u.t.*, como ocorreria sem a otimização.

6. Conclusões e Trabalhos Futuros

Em Sistemas Embarcados, a escolha de uma Arquitetura de Comunicação que melhor atenda às especificações da aplicação pode constituir-se um fator essencial para a obtenção de uma arquitetura otimizada que implemente o sistema completo. Isso ocorre por que, nesse tipo de sistema, as funções são distribuídas entre diferentes elementos de processamentos que necessitam trocar informações eficientemente.

Acredita-se que os objetivos propostos para este trabalho foram atingidos, uma vez que foi implementado e validado, um algoritmo para análise e seleção de arquiteturas de redes de interconexão com base em restrições temporais. Os resultados foram validados em três implementações de redes com características diferenciadas, a fim de comprovar a abrangência semântica da análise e seleção proposta.

Outra característica interessante diz respeito à implementação dos algoritmos para análise que, por serem realizados dentro do paradigma de orientação a objetos, puderam ser projetados para serem facilmente extensíveis à análise em diferentes arquiteturas e níveis de abstração.

Como trabalhos futuros, pretende-se ampliar a análise para a realização da seleção com base em outras variáveis de restrição de projeto, tais como área e potência. Além disso pretende-se avaliar outros exemplos para verificar-se a generalidade dos algoritmos de análise, bem como realizar testes com outras configurações de arquiteturas de interconexão, tais como o Barramento CoreConnect [IBM00a] [IBM00b] e SPIN [GUE00]. O passo seguinte trata de estender os algoritmos para a síntese das arquiteturas de comunicação, tanto em HW como em SW.

Finalmente, dentro do escopo geral do projeto, pretende-se integrar a estratégia de análise e seleção aqui proposta com o processo de síntese implementado no ambiente S³E²S para a implementação de um sistema completo.

Referências

- [AH98] ASHRAF, F. et. al. "Introduction to Routing in Multicomputer Networks". *ACM Computer Architecture News*, vol. 26, n. 5, December 1998, p. 14-21.

- [CAR00] CARRO, L., WAGNER, F., KREUTZ, M., OYAMADA, M. “A Design Methodology for Embedded Systems based on Multiple Processors”. DIPES2000.
- [DAV97] DAVEAU, J. et. al. “Protocol Selection and Interface Generation for HW/SW CoDesign”. *IEEE Transactions on VLSI Systems*, vol. 5, n. 1, March, 1997, p. 136-144.
- [GUE00] GUERRIER, Pierre; GREINER, Alain. “A generic architecture for on-chip packet-switched interconnections”. In: DESIGN AUTOMATION AND TEST IN EUROPEAN CONFERENCE (DATE2000), 2000, Paris *Proceedings...* Disponível por WWW em <http://www-asim.lip6.fr/~pg/SPIN/index.html> (13 de setembro de 2000).
- [HAM96] HAMACHER, C. *Computer Organization*. McGraw-Hill, New York, 4th ed., 1996.
- [IBM00a] IBM Microelectronics. *The CoreConnect™ bus architecture*. IBM Microelectronics, 2000. Disponível por WWW em http://www.chips.ibm.com/products/coreconnect/docs/crcon_wp.pdf (13 de setembro de 2000).
- [IBM00b] IBM Microelectronics. *CoreConnect Architecture: A 32-, 64-, 128-bit core on-chip bus structure*. IBM Microelectronics, 2000. Disponível por WWW em <http://www.chips.ibm.com/products/coreconnect> (13 de setembro de 2000).
- [KNU99] KNUDSEN, P. V.; MADSEN, J. “Integrating Communication Protocol Selection with Hardware/Software CoDesign”. *IEEE Transactions on CAD of Integrated Circuits and Systems*, vol.18, n. 8, August, 1999, p. 1077-1095.
- [NI93] NI, L. M.; MCKINLEY, P. K. “A Survey of Wormhole Routing Techniques in Direct Networks”. *IEEE Computer Magazine*, vol. 26, n. 2, February 1993, p. 62-76.
- [QUI94] QUINN, M. J. *Parallel Computing: Theory and Practice*. New York : McGraw-Hill, 1994.
- [TAN96] TANENBAUM, A. *Computer Networks*. Prentice-Hall, New Jersey, 3rd ed., 1996.
- [ZEF99] ZEFERINO, C.A. “Redes de Interconexão para Multiprocessadores”. Porto Alegre : PPGC UFRGS, Exame de Qualificação 57, 1999.
- [WAG99] WAGNER, F.; OYAMADA, M.; CARRO, L.; KREUTZ, M. (1999). “Object-oriented Modeling and Co-simulation of Embedded Systems”. In: IFIP X International Conference on VLSI, Lisbon, 1999, **proceedings ...** ... IEEE Computer Society Press, Los Alamitos, California, 1999.