

ネットワーク性能シミュレータの開発とバスのバッファ評価への適用

桜井 祐市[†] 新保 健一[†] 鳥羽 忠信[†]

組込み大型データ処理装置のハードウェアリソース見積りを目的にネットワークシミュレータを開発した。目標はプロセッサ数 100 個のシステムを実用的な時間内でシミュレーションする事である。既存のハードウェアシミュレータは、プロセッサ部の計算が数日で収束せず利用不可と判明した。そこで、解析時間短縮のためバッファ使用量の計算に特化し、また、DMA バスアクセス時間を用いプロセッサ部のバケット通過時間のみを計算する共有バス時間モデルを提案、MATLAB/Simulink を用いシミュレータを開発し、評価の結果ハードウェアシミュレータに比べ計算時間を 1/12 と短縮化を実現、大型データ処理装置への適用見通しを得た。

Development of the Network Performance Simulator and its Application to Buffer Evaluation of the Bus

Yuichi Sakurai[†], Kenichi Shimbo[†] and Tadanobu Toba[†]

In this work, a network simulator is developed to predict hardware resource of the embedded large size data processing system. The target of this work is to simulate processing system with 100 processors in practical time. It is difficult to simulate with the existing hardware simulator, because the calculation time of processors costs several days. For reducing the analysis time, the calculation of the buffer consumption is specialized, and a "shard bus timing model" is proposed for calculating only packet transit time of the processing part which used by DMA bus access time. And the simulator is developed by using MATLAB/Simulink. Compare with the existing hardware simulator, the calculation time is reduced to 1/12 with the proposed simulator. And it is confirmed that the proposed simulator can be used for simulation of large size data processing system.

1. はじめに

マニピュレータ等制御機器を搭載した製造装置や分析・解析装置などに代表される産業用の大規模組込みデータ処理装置 (HPEC, High-Performance Embedded Computing[1]) においては、ハードウェアリソースを、事前に高精度に見積ることが重要である。産業用 HPEC は、製造装置、スイッチ、プロセッシングと、大規模インタコネクネットワークという構成からなる(図 1)。近年、高精度な制御を目的とした製造装置の性能向上や、扱う物理量の多様化から、処理データは大容量化しており、現在、汎用ネットワークと汎用コンピュータを複数並べた並列処理装置が多く用いられている。

今回対象としたこれらの装置のインタコネクネットワークは、製造装置からプロセッシングまで一方向に連

続した大容量のパケットデータが伝送される点の特徴である。パケットは伝送途中で破棄されず、伝送エラーが発生した際はリトライ処理が行われ、確実に全データを伝送する。このようなインタコネクネットワークを持ったシステムにおいて、従来一時的なバッファ使用量増大によるバッファオーバーフローは、システムの停止などの致命的なエラーを引き起こしていた。

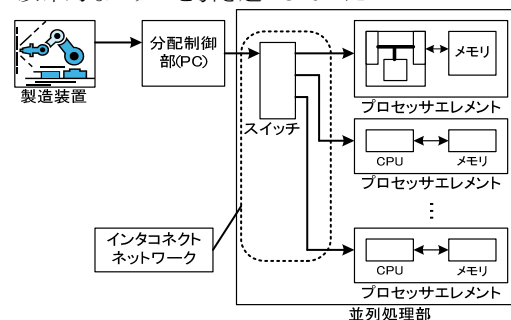


図 1 大規模組込み処理装置

[†]株式会社 日立製作所 横浜研究所
Hitachi, Ltd., Yokohama Research Laboratory

大規模組込み処理装置においては、基本的に製造装置はデータ取得を開始したら、完了まで中断せずデータを生成し続ける。一時的なバッファ使用量増大は、ネットワークの輻輳などが原因の一つである。また、プロセッサアーキテクチャによっては、ある特定の時間に集中して複数のコアから連続的にメモリへアクセスが発生し、共有バスの長時間の占有が起きる。この長時間の共有バス占有により、プロセッサ部(プロセッサエレメント)を先頭にネットワーク上に伝送データが滞留する。製造装置のデータを処理するデータ処理装置では、従来一時的なバッファ使用量増大などでもシステムを安定的に稼働し続けるため、バッファ量やネットワークの伝送容量に余裕を持って設計する手法をとってきた。しかし、これら対処法は根本的な解決策にはならず、装置の高コスト化の要因になっていた。また、製品開発時においても、システム性能検証時に一時的なバッファ使用量増大によるネットワークエラーが発生した場合、システム立ち上げまでの TAT を悪化させる原因となっている。

このようなシステムを安定して稼働させるためには、ネットワークのバッファ容量などのハードウェアリソースを事前に予測することが重要である。しかし、一つのインタコネクネットワークにプロセッサエレメントが数十から 100 個以上搭載される産業用 HPEC においては、ネットワークパケットの遷移が複雑であり、机上計算によるハードウェアリソースの予測は困難である。そこで今回、ハードウェアリソースを見積もるための、大規模組込みデータ処理装置のシミュレーション技術について検討した。特に、並列プロセッサエレメント、大規模インタコネクネットワークを含むシステムの評価を念頭とし、シミュレーションが実用的な時間で完了するための検討を行った。

今回、我々は、ハードウェアリソースを評価可能とするネットワークシミュレータを MathWorks 社の MATLAB/Simulink ソフトウェアを用い開発した。特徴は、評価するハードウェアリソースとしてバッファ使用量の計算に特化し、プロセッサエレメントの解析時間短縮のため DMA バスアクセス時間を用いプロセッサエレメントのパケット通過時間のみを計算する共有バス時間モデルを用いた事である。本稿では、大容量データをリアルタイムに処理する大規模組込みデータ処理装置のハードウェアリソース見積りを可能とするシステムシミュレーション技術について述べる。

本稿では 2 節において、ハードウェアリソースのシミュレーションについての関連研究を紹介する。3 節では、

ハードウェアリソース評価として、比較対象とした HDL によるハードウェアリソースの評価と、本研究で採用したネットワーク性能シミュレーション手法について述べる。4 節では、開発したネットワークシミュレータの概要について述べる。このツールを利用したモデリングの例と検証を 5 節に示す。その結果を 6 節で考察する。最後に 7 節でまとめと今後の課題を述べる。

2. 関連研究

ハードウェアの挙動をモデリングするシミュレーション言語としては HDL(Hardware Description Language, ハードウェア記述言語)[2]が多く用いられている。HDL はハードウェアを高精度にシミュレーションできる反面、シミュレーション速度が課題となる。また、大規模組込み処理装置全体の記述をするには記述量が膨大となる恐れがあるなどの課題が存在した。システム全体のシミュレーションを行うには SystemC[3]などの抽象度を上げたモデリング言語も存在し、バスのバッファ評価などへの適用例[4]も存在する。

ネットワークの挙動をモデリングするシミュレーション言語はいくつか代表例が存在する。その一つとして ns(Network Simulator)を挙げる[5]。ns は 1995 年に DARPA の支援で VINT プロジェクトとして LBL, Xerox PARC, UCB, USC/ISI で開発されたもので、研究機関でのネットワークプロトコルや通信手順の研究でデファクトスタンダードである。分散型イベントのネットワークシミュレータであり、オブジェクト指向の設計(Otcl スクリプトによるシナリオ(ネットワークポロジ、パラメタ)設定)や、nam(Network Animation)外部ツールによる結果の可視化といった特徴がある。標準では Ethernet プロトコルのシミュレータとして動作し、パケットロスありが前提として動作する。しかし今回はパケットロス無しが前提のインタコネクネットワークを対象としている。再送制御などによるレイテンシやシステム性能への影響を無視するわけにはいかないため、今回の目的には ns をそのまま適用は出来なかった。また、後述するネットワークシステムからのソフトウェア・ハードウェア協調設計の観点において、MATLAB/Simulink を用いたほうがモデルを拡張し易いと判断した。

MathWorks 社の MATLAB/Simulink ソフトウェアを用いたネットワークシミュレータの関連研究では、Lund 大の TrueTime が挙げられる[6]。これは、プロセスのリアルタイム性の評価に用いるシミュレータであり、ジェネレータ、ネットワークノード、スイッチなど、システムをモデリングする事も可能で、周辺機器との接続など、組込

み機器開発環境との親和性も高い。しかし、本シミュレータは、ハードウェアリソースの評価は不可である。

これらの先行研究をふまえ、本研究では次の手法を提案する。コンピュータバスなどを表現するのに用いられる HDL と同様に、ハードウェアリソースを評価可能とするシミュレータを開発する。評価対象は、パケットロスのないインタコネクタネットワークを用いる大規模システムである。シミュレーションプラットフォームは MATLAB/Simulink を用いる。Simulink は制御系開発の分野では普及しており、開発ノウハウやモデル資産の利用の観点から継続して使いたいという思いがあった。また、Simulink の GUI エディタを用いる事でネットワークポロジ構成を容易化する。加えて、システムのシミュレーションからソフトウェア、ハードウェアの設計、検証までを協調して行う環境としても優れている。HDL Coder[7]などモデルからハードウェアを直接生成する技術を有し、組み込みプロセッサエレメントや FPGA との協調シミュレーションも可能であり、システムベースのソフト・ハード協調設計環境として優れている。シミュレータ開発に際して、シミュレーションモジュール間の同期など、時間の管理は MATLAB が担当する。これは、大規模システム適用時の時間管理が将来、煩雑になることを避けるためである。MATLAB は、離散イベント発生と、モジュールのイベント更新を行うプラットフォームとして機能する。シミュレーションモジュールについては MEX (MATLAB EXecutable) インタフェースを用い、それぞれのモジュールは C++ で記述した。最終的なゴールとしては、シミュレーションモジュールの並列化等により、シミュレーション速度の高速化を狙う。

3. ハードウェアリソース評価手法

3.1. ネットワークバッファ使用量評価

産業用 HPEC システムで評価すべきハードウェアリソースは、ネットワークの実効伝送容量、レイテンシなど、代表的なものでいくつか存在する。今回はそのうち、ネットワークバッファの使用量を評価することとした。産業用途などリアルタイム性を要求されるシステムにおいてネットワークバッファ使用量の評価は、システムの安定稼動のために重要である。システムが破綻しないようにするには、バッファ使用量の上限値をあらかじめ評価し求め、この上限値以上のバッファを具備するシステムを構築する必要がある。ネットワークバッファの使用量は、ポートの入力容量と出力容量の関係から決まる。概要を図 2 に示す。

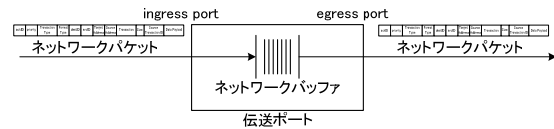


図 2 ポート入出力容量によるバッファ使用量評価

この、ネットワークバッファ使用量の評価を、先ず、HDL を用いて実施した。以下、結果について説明する。

3.2. HDL シミュレータによるバッファ使用量評価

HDL を用い、1 対 1 の単純なノード間でバッファ使用量を評価する。モデルとして、FPGA を対向させ SRIO (Serial RapidIO) プロトコルネットワークを構築した。図 3 に、評価用に試作した HDL モデルの構成、表 1 にモデル概要、表 2 にシミュレーション環境を示す。FPGA TX から一定容量のデータを連続し伝送させ、FPGA RX の輻輳が 100us 続いた場合の FPGA TX のバッファ使用量を評価した。

結果、FPGA TX のバッファ使用量は 9768Bytes (約 40 パケット) に上り、シミュレーション時間は 5.8 分であった。実際のシステムシミュレーションはこのように単純ではなく、例えば FPGA RX の代わりに図 4 に示すプロセッサモジュール (MPU) モデルを受信側として用いシステムをモデリングする。プロセッサモジュールは CPU コアや共有バスなど多数のコンポーネントから構成され、HDL によるシミュレーション時間はさらに増大する。また、実際のシステムのシミュレーション評価では 1 万パケット程度を用いることから、プロセッサモデルを HDL でモデリングしシミュレーション評価するには、実計算機の計算量が発散し、実用的な計算時間内では収まらない問題があった。

そこで、システムのバッファ使用量の変化について検討を行い、高速処理を可能とするネットワークシミュレータを提供可能とする手法を提案する。以下、検討結果について述べる。

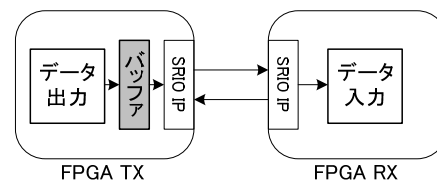


図 3 HDL バッファ評価モデル

表 1 HDL バッファ評価モデル 概要

#	項目	値
1	SRIO IP	Altera RapidIO V10.0
2	シミュレータ	Mentor Graphics ModelSim SE 10.0b
3	言語	Verilog-HDL

表 2 シミュレーション環境

#	項目	値
1	OS	Windows XP 32bit
2	メモリ	4GBytes
3	CPU	Intel Core i7 870 2.93GHz

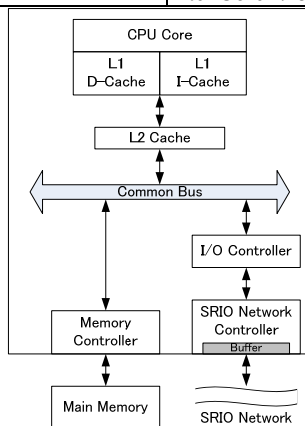


図 4 プロセッサモジュール(MPU)モデル

3.3. 提案手法

産業用 HPEC システムにおいてバッファ使用量が変動する要因は主に 2 つ存在する。説明のため、複数の製造装置から、ネットワークパケットがスイッチを経由し、単一のプロセッサエレメントで処理するようなネットワークポロジを図 5 に示す。

第一の要因は、ネットワークポロジに起因する。図 5 のシステムにおいて、複数の製造装置からのパケットデータがスイッチの同一出力ポートに集中した場合、ポートの輻輳が発生する。そして、スイッチの伝送容量が減少し、スイッチのバッファの消費速度が減少する。しかし、スイッチからのデータは連続して入力され続けるため、バッファの使用量が増加する。

第二の要因は、プロセッサエレメントの CPU core の処理負荷によるネットワーク伝送容量の変化である。プロセッサエレメントの CPU core が高負荷で、プロセッサエレメントの共有バス利用可能率が低い場合、プロセッサエレメントのネットワークインタフェースの伝送容量が減少する事となり、このプロセッサエレメントにデータを供給しているポートのバッファの利用量が増加する。

以上から、システムのハードウェアリソースとしてネットワークバッファ容量を評価するには、ネットワークポロジの輻輳によるものと、プロセッサエレメントの処理負荷によるものの 2 つの伝送容量変化をモデリング出来ればよい。ネットワークポロジ起因の輻輳による伝送容量変化のモデリングは、Simulink モデルとして動作するネットワークスイッチモデルを開発し用いる事で実現する。実際のネットワークスイッチ同様の処理を実装し、実際の輻輳状態を再現し、伝送容量変化を観察する。また、ポート間のネットワークパケットのやり取りは、Simulink オブジェクトの遷移で表現することとし、HDL に比べ処理の簡略化を狙う。

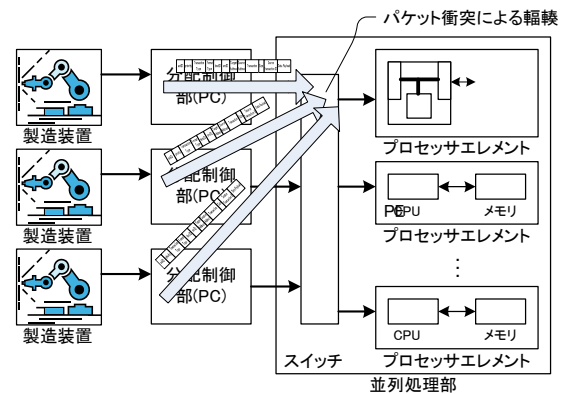


図 5 複数の製造装置のパケット衝突による輻輳

プロセッサエレメントの処理負荷による伝送容量変化のモデリングは、共有バス時間モデルを用い実現する。以下、前節の図 4 に示すプロセッサモジュール(MPU)モデルの例を用い、共有バス時間モデルについて説明する。製造装置で生成したネットワークパケットがプロセッサエレメントに対し行う主なアクセスは、プロセッサエレメントの共有バスを通じた、メモリへの Read、Write 処理である。ここで、高負荷の CPU コアがメモリへの大量の Read、Write を行った場合、プロセッサエレメントの共有バスが長時間占有され、ネットワークインタフェースは共有バスの利用権が獲得できず、パケットはネットワークインタフェースの内部のバッファに待機する。このパケットの待機状態はレイテンシとしてカウントされ、システムのネットワーク性能を悪化させる。以上から、CPU コアによるプロセッサエレメントの共有バスの占有の有無が、プロセッサエレメントにおける伝送容量を変化させるパラメタとなり得る事が分かる。今回、この共有バス占有状態を時間で表現したものを共有バス時間モデルと呼ぶ事とし、プロセッサエレメントのモデリングに用いた。概要を図 6 に示す。プロセッサエレメントは、バス利

用可能状態とバス利用不可能状態の2状態どちらかをバス利用可能率に基づき選択する。ネットワークパケットがプロセッサエレメントに到着時、バス利用可能状態であればメモリ格納レイテンシをサービス時間として与え、バス利用不可能の場合は、ネットワークインタフェース内のバッファにパケットデータを保持し、所定のレイテンシに基づきサービス時間を与える、という計算を行う。プロセッサエレメントモジュールのシミュレーションとして、ネットワークパケットに対し、バス利用可能と不可能状態の2状態どちらのサービス時間をネットワークパケットに対し与えるか、その選択のみをすることが、この計算方式の利点である。

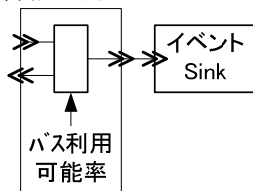


図 6 プロセッサエレメントモジュール
(共有バス時間モデル)

以上から、実利用可能な計算時間内で、システムのネットワーク性能とハードウェアリソースを評価可能とする。次に、本節で示したネットワーク性能シミュレーション方式に基づいたネットワークシミュレータの開発について述べる。

4. ネットワークシミュレータの開発

ネットワークシミュレーションの目的は、ハードウェアリソースの動きを観察し、見積りを実施する事である。また、ネットワークバッファの使用量が一時的に高くなるなど、定常状態以外の状態も把握したかったため、パケット遷移を基準とした Discrete-Event Network Simulator[8]として開発を行う。

開発は、標準的な大容量組み込み情報処理装置をモデリング出来るよう進めた。プロトコルとしては、インタコネクトネットワークで一般的に用いられる Serial RapidIO プロトコル[9]を参考にした。図 7 に、シリアルパケット形状を示す。パケットを生成するジェネレータのブロック図を図 8 に示す。入力されるパケット発生パターンに従って間隔を空けながらパケットを生成する。パケット出力パターンは、固定長の時間と量と行き先を繰り返し実行するもの、一定容量(CBR、Constant Bit Rate)、ランダムに出力するもの、MATLAB/Simulink の外部トリガ SimEvent モジュールを用いるものから選択可能とした。伝送線路モジュールのブロック図を図 9 に示す。

伝送容量[Gbps]をパラメタとして持ち、シリアルパケットオブジェクトを入力後、ペイロードサイズを解析し、伝送線路の通過に掛かる時間を計算し、サービス時間として与える。スイッチモジュールは、シリアルパケットオブジェクトを入力、スイッチ設定表に従って、出力ポート先を切替え、出力する。ブロック図を図 10 に示す。サービス時間は、バッファ通過にかかるレイテンシをns単位で、設定値は入力、出力ポート共通とし、入力、出力バッファ通過時にそれぞれ加算、バッファに複数サイクル滞留した場合はそのサイクル分加算される。プロセッシングモジュールは、インタフェースとしてはネットワークパケットの IN/OUT が可能なポートを持つ。前節で示したプロセッサエレメントモジュールのモデリング方式に従い、ネットワークインタフェースから見たプロセッサエレメントの共有バスの利用可能率の設定が可能なブロックとして開発した。

ack ID	priority	Transaction Type	Format Type	dest ID	src ID	Target Address	Source Address	Transaction	Size	Source Transaction ID	Data Payload
--------	----------	------------------	-------------	---------	--------	----------------	----------------	-------------	------	-----------------------	--------------

図 7 シリアルパケット形状

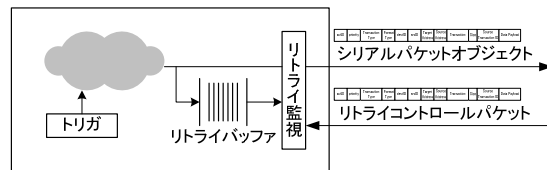


図 8 パケットジェネレータ ブロック図

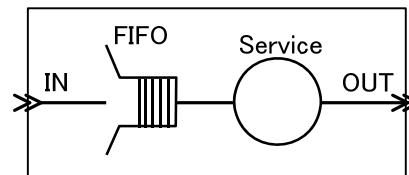


図 9 伝送線路モジュール ブロック図

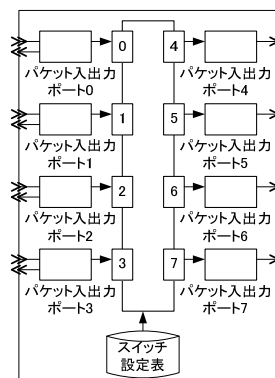


図 10 スイッチモジュール ブロック図

バッファ使用量については、各モジュールのバッファに、最大使用量を記録する記録部を設ける。システムシミュレーション完了後、全モジュールの最大使用量を、性能評価値として GUI、もしくはテキスト情報として出力する。以上開発したシミュレーションソフトウェアの構成を図に示す。

本開発ソフトウェアにより、ネットワークの輻輳等によるボトルネックとシステム全体への影響と、プロセッサエレメントの処理による共有バスの輻輳とシステム全体への影響とを統合して確認可能とする。また、組込みシステムでのリアルタイム伝送を確保するために、プロセッサエレメントの共有バスを含む超高速大容量システムにて重要となる輻輳制御の方式を検証可能となり、輻輳状態発生の有無を事前にシミュレートすることにより必要なバッファサイズを事前に予測可能とする。

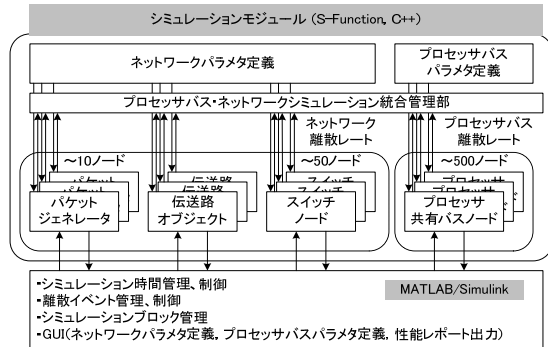


図 11 開発ソフトウェア構成

5. 実験

5.1. 概要

開発したネットワークシミュレータの、ハードウェアリソースの見積もり精度を検証する。実験は、ネットワークをモデリングしてシミュレーションを実施し、ネットワークバッファの使用量について3.2節の HDL の結果と比較する。作成したシミュレーションモデルを図 12 に、表 3 にシミュレーション条件を示す。シミュレーション環境は先に示した HDL の環境(表 2)と同一である。シミュレーションモデルは、単一のパケットジェネレータ、スイッチ、8 プロセッサエレメントモジュールで構成した。パケットはジェネレータにて一定容量(CBR, Constant Bit Rate)で生成し、スイッチモジュールを経て各プロセッサエレメントモジュールへ分配する。本実験は先に示した HDL の結果との比較を主眼とし、プロセッサエレメントモジュール負荷に起因するバッファ使用量増大のみをモデリングし評価した。なお、パケットジェネレータを単一ではなく複数用いる事などにより、ネットワークポロ

ジの輻輳によるネットワークバッファの使用量評価も、本ネットワークシミュレータでは実施可能である。

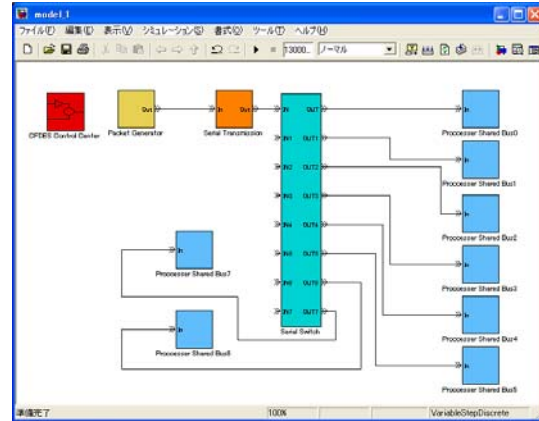


図 12 シミュレーションモデル

表 3 シミュレーション条件

#	項目	値
1	プロセッサエレメント	8
2	シリアル伝送路	2.5Gbps
3	ネットワークバッファ	最大 80 フレーム(20KBytes)
4	バス占有/開放時間	100 μ s / 100 μ s
5	CFDES	Ver 2.0.1
6	MATLAB/Simulink	7.11.0.584 (R2010b)

プロセッサエレメントモジュールについては、前節で示した、プロセッサエレメントの共有バスの占有状態をベースとしたモデリングを行う。今回は、簡易的な評価パラメタとして、CPU コアのメモリへのアクセス時間と非アクセス時間の 2 つを用いた。この 2 つのパラメタを用い、CPU コアが共有バスを長時間占有する事によるネットワークスループットへの影響を簡易的に求める。図 13 に、今回用いた共有バス時間モデルのブロック図を示す。ネットワークパケットがプロセッサエレメントモジュールにてサービスを受けられる期間(bus available)とサービスを受けられない期間(bus blocking)をそれぞれ連続した時間として定義し、バス占有時間は3.2節の HDL の実験条件と等しく100 μ sとした。それぞれの時間はネットワークパケットの到着とは無関係に、周期的に繰り返すものとする。プロセッサエレメントモジュールを今回このようにモデリングした狙いを以下挙げる。第一に、本実験は3.2節の HDL との比較を目的とした。一定期間のバス占有によるネットワークバッファ使用量の変化のみを評価すべく、パケット到着に連動したプロセッサ負荷の変動のモデリングは省略した。また、パケット蓄積によるネットワークバッファ使用量増大と共にパケ

ット放出によるバッファ使用量減少動作の確認もすべく、バス占有とバス開放を行うモデリングを行った。そして、バス占有とバス開放を行うシミュレーションを長期間実施した場合、実機におけるネットワークバッファ使用量の最大値が確認出来るものと考えた。

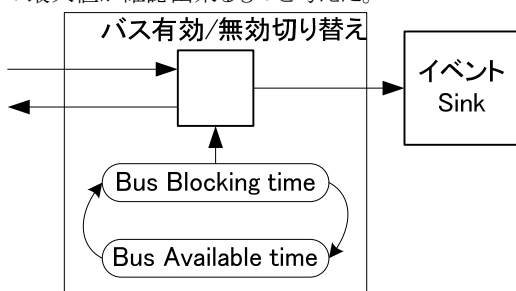


図 13 共有バス占有時間モデル

5.2. 結果

シミュレーション結果を、HDL によるハードウェアモデルの結果と合わせ表 4に示す。実行時間は、HDL が 8.7 秒/パケット伝送に対し、本開発シミュレータは 0.75 秒/パケット伝送となり、実行時間を 1/12 に短縮できた。約 1 万パケットの伝送を 2 時間で完了する。また、バッファ使用量は最大 8704Bytes であった。バッファ使用量グラフを一部拡大し図 14に示す。bus blocking の時間帯に、ネットワークバッファにパケットが蓄積されている事が分かる。HDL のシミュレーションでのバッファ使用量は最大 9768Bytes で、本開発シミュレータとの誤差は-1064Bytes、約 4 パケット分であった。今回開発したネットワークシミュレータの目的はネットワークバッファ使用量を予測する事である。実験結果は HDL との誤差が約 1 割程度となり、本開発シミュレータは実設計に適用可能であると考えている。

表 4 シミュレーション結果と実行時間の比較

	本開発 シミュレーション	HDL
ネットワークバッファ 使用量	8704Bytes	9768Bytes
実行時間	30 秒	5.8 分

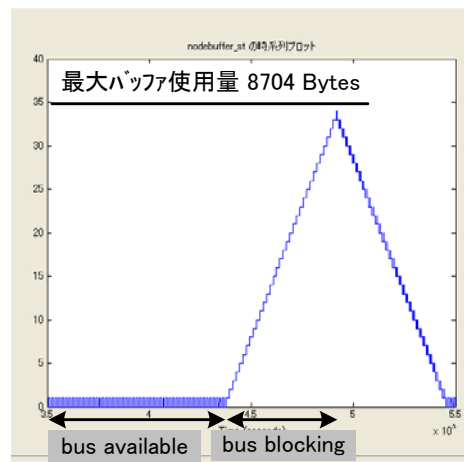


図 14 バッファ使用量の時間変化

6. 議論

本開発のシミュレータの速度は、HDL よりは早いですが、一般的なネットワークシミュレータから比べると遅い。例えば、MATLAB/Simulink 付属のサンプルモデル (Ethernet ネットワークシミュレータ) に比べると 1/6 の性能に留まった。実際のシステムのシミュレーション評価では 100 万パケット程度を用いるため、実機適用時に問題となる可能性がある。解決策は現在検討中であるが、現状、精度を高めるために ns 単位でシミュレーション計算を行っている箇所がある。これをパケット単位の離散シミュレーションに変更する、というのは解決法の一つである。また、C++モジュールの並列化による PDES(Parallel DES)も解決法の一つである。

また、本開発のシミュレータは、HDL と比較してシミュレーション結果の精度が 4 パケット程度に収まった。この要因の一つは、システム構成が単純であることが挙げられる。ネットワークパケットを生成するシミュレーションブロック 1 個を先頭に、スイッチを介しプロセッサエレメントモジュールが 8 個つながる単純ツリー構成であり、ポート間では HDL とおおよそ同一のネットワーク処理が行われていたものと想定される。しかし、今後適用を検討する大型の組込み情報処理装置においては、例えばファットツリーなどの複雑なトポロジとなり、シミュレーション精度の悪化が懸念される。ネットワークトポロジに起因する輻輳の発生がバッファ使用量の変動要因の一つであることは、3.3節で言及している。今後、解析するシステムネットワークが複雑化した際に高精度なシミュレーションを実現するには、ネットワークの双方向化技術と、ネットワークドライの実装の 2 つの技術開発が

必要と考えている。図 15 に双方向化技術の例として双方向対応スイッチノードの機能概略を示す。従来は単方向だったポートを双方向に拡張し、入出力パケットによる輻輳の発生など、より実機に近い処理のモデリングを実現する。図 16 に、ネットワークリトライ処理の概念図を示す。SRIO プロトコルでは、受信側バッファが Full の場合に、リトライコントロールパケットが受信側から送られる。送信側においては、このコントロールパケットの要求に応じ、内部バッファ内を順序に従って送出する。この、パケットリトライ機能を実装する。リトライ制御には以上示したリトライコントロールパケットと呼ぶパケットオブジェクトを用いるため、このパケットを伝送するための時間的なレイテンシが発生する。Wu らにより、SRIO プロトコルのリトライ処理によりレイテンシがほぼ倍になる報告[10]がある。リトライ処理によってパケットバッファの出力伝送容量は 1/2 に減少するという事であり、このレイテンシをシミュレーションモデルにて再現することにより、より実機に近いシステムのモデリングが可能になると考えている。また HDL と比較しシミュレーション結果が4パケット程度に収まったもう一つの要因は、プロセッサエレメントの共有バスの占有状態モデリングを簡易的に行ったことも挙げられる。今回の実験では共有バス性能を表すパラメタを単一のバス占有時間 100 μ sのみとし、バス占有、開放を周期的に繰り返すモデリングを行った。これにより HDL とおおよそ同一のプロセッサエレメントモジュール負荷に起因するバッファ使用量増大が発生したものと想定される。実際の情報処理装置のプロセッサエレメントの共有バス状態は、プロセッサエレメントのソフトウェア処理に起因して複雑に変化する。これはインタコネクネットワークの性能に強く影響する因子である。今後、解析するシステムの処理内容に応じた高精度なシミュレーションを実現するには、実プログラムのプロファイリングによるプロセッサエレメントの共有バス時間モデリング技術の開発が必要と考えている。プロファイル結果から共有バス時間モデルを生成する事で、実機に近い処理負荷でのシステム性能解析が実現可能になると考えている。

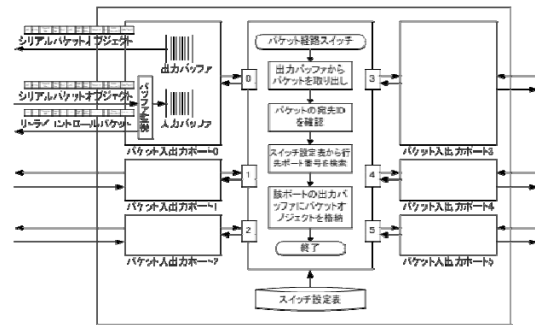


図 15 双方向対応スイッチモジュール

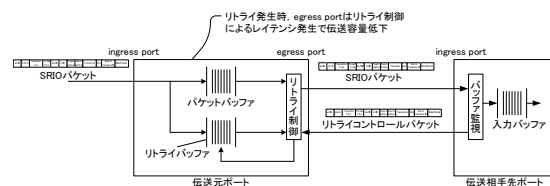


図 16 ポート間リトライ制御

7. まとめと今後の課題

本研究では、ハードウェアリソースを評価可能とするネットワークシミュレータを MathWorks 社の MATLAB/Simulink ソフトウェアを用い開発した。ハードウェアリソースとしてバッファ使用量の計算に特化し、プロセッサ部の解析時間短縮のため DMA バスアクセス時間を用いプロセッサ部のパケット通過時間のみを計算する共有バス時間モデルを用いた事を特徴とする。ネットワークバッファ使用量の評価においては、HDL によるネットワークモデルとの比較で解析時間を 1/12 と短縮化を実現し、大型データ処理装置への適用見通しを得た。

シミュレーション速度の更なる高速化や、双方向ネットワークへの対応、パケットリトライ処理への対応、および他のハードウェアリソース評価への対応などは今後の課題である。

商標

MATLAB、及び Simulink は米国 The MathWorks, Inc.の登録商標。

SystemC は Open SystemC Initiative の登録商標。

Verilog は Cadence Design Systems の登録商標。

RapidIO は RapidIO Trade Association の登録商標。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

参考文献

- [1] Wayne, W.: High-Performance Embedded Computing: Architectures, Applications, and Methodologies , Morgan Kaufmann, 2006
- [2] IEEE Standards Association: IEEE Standard Verilog Hardware Description Language , IEEE Std 1364-2001, 2001
- [3] IEEE Standards Association: IEEE Standard for Standard SystemC Language Reference Manual, IEEE Std 1666-2011, 2012
- [4] 石田 耕三, 小野 みどり, 遠山 治, 影本 哲哉, 小山 雅行, 細谷 史郎: メモリバス SystemC モデルを用いたバスのバッファ評価, 電子情報通信学会技術研究報告, SIP 105(635), pp. 117-121, 2006
- [5] The Network Simulator - ns-2,
<http://www.isi.edu/nsnam/ns/>
- [6] Anton, C., Dan, H., Bo, L., Johan, E., and Karl-Erik, A.: How Does Control Timing Affect Performance? Analysis and Simulation of Timing Using Jitterbug and TrueTime, IEEE Control Systems Magazine, 23:3, pp. 16-30, 2003
- [7] The MathWorks, Inc.: HDL Coder,
<http://www.mathworks.co.jp/products/hdl-coder/>
- [8] Mohsen, G., Ammar, R., Bilal, K., Ala, A.: Network Modeling and Simulation: A Practical Perspective, John Wiley & Sons, 2010
- [9] RapidIO Trade Association: RapidIO™ Interconnect Specification, 2011
- [10] Wu, F., Jia, S., Li, W., Wang, Y.: A serial physical layer design in RapidIO, Electron Devices and Solid-State Circuits(EDSSC), pp. 1-4 , 2010