

バス調停の遅延時間見積もりのための確率的数学モデル

柴田 誠也^{†1,†3} 石川 拓也^{†1} 本田 晋也^{†1}
富山 宏之^{†2} 高田 広章^{†1}

マルチプロセッサシステムの性能見積もりを目的として、バス調停による遅延時間を高速に見積もるための確率的数学モデルを提案する。複数のバス調停方式 (FCFS, 固定優先度, ラウンドロビン) に対応したモデルを定義した。提案モデルを用いることで、共有メモリへの複数プロセッサからの同時アクセス時のバス調停遅延時間を短時間で見積もることができる。提案する確率的数学モデルの妥当性および効果を、モンテカルロシミュレーションおよび FPGA 上システムの実行との比較により評価した。

Stochastic Models of Bus Arbitration Delay for Performance Estimation of Multiprocessor Systems

SEIYA SHIBATA,^{†1,†3} TAKUYA ISHIKAWA,^{†1}
SHINYA HONDA,^{†1} HIROYUKI TOMIYAMA^{†2}
and HIROAKI TAKADA^{†1}

This paper presents definitions of stochastic models of bus arbitration delay in order to estimate performance of multiprocessor systems in short time. Three arbitration policies, first-come first-served, fixed-priority and round-robin, are modeled. We verified the model by comparing with Monte-Carlo simulation, and evaluated effectiveness of them by estimating performance of a system on an FPGA.

†1 名古屋大学 大学院情報科学研究科

Graduate School of Information Science, Nagoya University

†2 立命館大学 理工学部

College of Science and Engineering, Ritsumeikan University

†3 日本学術振興会

Japan Society for the Promotion of Science

1. はじめに

近年、組込みシステムは多機能化、マルチプロセッサ化が進んでいる。特に、マルチプロセッサ化に伴うシステムの並列性向上により、共有メモリ型アーキテクチャにおいては、複数のプロセッサがひとつの共有メモリに対してアクセスする場合 (メモリ衝突) が増加しており、バスやメモリインタフェースでの調停遅延が性能に与える影響が大きくなってきている。バス構成の設計においては、調停遅延を考慮して、バス調停方式やメモリ配置、およびメモリアクセス分布を探索する必要がある。

システムの設計早期に性能を見積もることは、アーキテクチャの決定を行うにあたり重要である。見積もり手法が無い場合や、見積もりの精度が低い場合には、ハードウェア製造後に要求の未達成が発覚し大きな手戻りを発生させる恐れがある。マルチプロセッサシステムを設計する場合には、処理量や並列性だけでなく、バス調停遅延を考慮することが、精度の高い見積もりには必要である。

一般的には、設計早期の性能見積もりはシミュレーションを用いて行われる。サイクル精度シミュレーションを用いることで、メモリ衝突を含めた詳細な見積もりが可能となるが、シミュレーション速度は遅い。高速なシミュレーション手法としては、システム内の通信を抽象化した TLM (Transaction Level Model) などを用いる手法がある。TLM は通信を抽象化することで機能のシミュレーションを効率化できるが、メモリ衝突を含めた見積もりのためには、通信についてもサイクル精度と同等に詳細なモデル化が必要となる。FPGA 上プロトタイプ実装を用いた性能評価も行われている。そのために、FPGA 上プロトタイプ実装を自動で行うツールも多く提案されている。しかし FPGA は観測性が低く、性能ボトルネックの所在を分析することが難しい。

性能見積もりのための、高い抽象度でのシミュレーションとして、トレースベースシミュレーションが提案されている¹⁾²⁾。これらは、何らかの方法でシステムを構成するコンポーネント (プロセッサやメモリ、処理内容) の性能パラメータ (トレース) を取得し、トレースを再利用することで内部動作の詳細なシミュレーションなしに、システム全体の性能を見積もるものである。ただし、トレースベースシミュレーションでは、処理を性能パラメータによって抽象化するために、動作の詳細な情報は持たない。そのためメモリアクセスタイミングの詳細な情報が不足し、メモリ衝突を扱うことは難しい。文献 3) では、回帰モデルを用いたメモリ衝突のモデル化を提案している。回帰モデルは定式化の難しい事象に対してフィッティングを行うことができるというメリットがある一方、学習を必要とするため、適

切な学習用入力を用意する必要があるというデメリットがある。

本論文では、トレースベースシミュレーションを前提とした、確率的数学モデルによるメモリアクセス時のバス調停遅延時間の見積もり手法について述べる。確率的数学モデルは、メモリアクセスが時刻的にランダムに発生するシステムを想定し、プロセッサ間でメモリ衝突が起きうる時間と、その時間内でのメモリアクセス回数のみを入力として、バス調停遅延時間の期待値を出力する。対応するバス調停方式は、FCFS、固定優先度、ラウンドロビンの3つである。提案する確率的数学モデルの妥当性および効果を、シミュレーションおよびFPGA上システムの実行との比較により評価した。

2. システムレベルでの設計探索フレームワーク

本章では、本論文で想定する設計探索の目的および探索フレームワークについて述べる。目的は、設計早期における、実現したいシステムの機能の、ハードウェアアーキテクチャへのマッピングの決定である。探索要素としては、機能、ハードウェアアーキテクチャ、およびマッピングがある。機能は、すでに複数の並列動作可能な動作ブロック（プロセス）として適切に分割されているとし、機能の分割は探索対象ではない。ただし、プロセス間の通信については、FIFOチャンネルで表されるとし、FIFO中のデータバッファの数は可変で探索対象である。ハードウェアアーキテクチャとしては、プロセッサ数、専用ハードウェアの使用有無、メモリ種類や数が主な探索対象である。マッピングは、機能が持つプロセスおよびチャンネルの、ハードウェアアーキテクチャ上への割り当てが探索対象である。

これらの探索を効率的に行うためのフレームワークとして、我々はSystemBuilderと高速性能見積もりツールという2つのツールを開発してきた。SystemBuilderは、プロセスとチャンネルを自動合成し、FPGA上実装を生成するツールである⁴⁾。FPGA上プロトタイプ実装を自動合成することで、システムの評価が短時間で可能になる。SystemBuilderによる自動合成フローを図1に示す。さらに、SystemBuilderにより自動合成されたFPGA上プロトタイプ実装は、プロファイルを追加することで、FPGA上でのプロセスの並列動作状況をトレースとして取得することが可能である⁵⁾。高速性能見積もりツールは、FPGA上実装の結果として取得されるトレースを利用し、ホスト計算機上で、複数のマッピングの性能を見積もるトレースベースシミュレーションを行うツールである⁶⁾。

高速性能見積もりツールが行うトレースベースシミュレーションでは、事前にFPGA上にて取得しておいた、あるマッピングのトレースをもとにして、別のマッピング時（異なるハードウェアアーキテクチャを含む）のシステムの実行時間を見積もる（図2）。このと

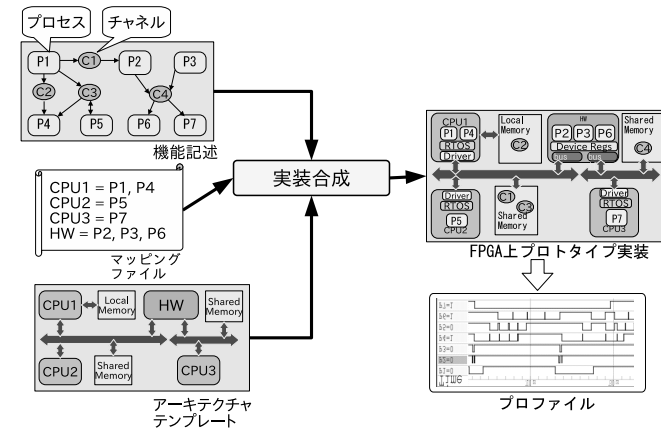


図1 SystemBuilderによる自動合成フロー
Fig.1 Automatic synthesis flow of SystemBuilder.

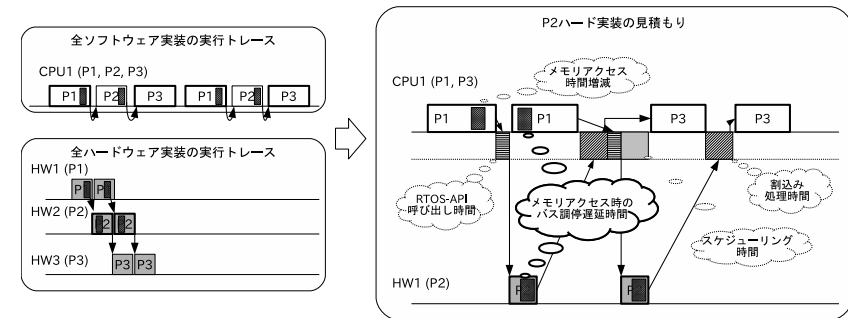


図2 高速性能見積もり手法の概念図
Fig.2 Concept of fast performance estimation method.

き、各プロセスの処理時間は、マッピング先 PE(Processing Element: プロセッサや専用ハードウェア) によってのみ変化すると仮定する。メモリアクセス時間については、マッピング変更による処理の並列性の変化に伴い、バス調停遅延によって増加する可能性がある。しかし、文献6)の時点では、このバス調停遅延時間を考慮していなかった。

本研究では、高速な性能見積もりを実現しつつ、共有メモリアクセスにおけるバス調停遅

延の影響を精度良く反映するために、確率的数学モデルを提案する。

3. バス調停遅延時間の確率的数学モデル

トレースは、FPGA 上にてシステムを並列性のない状態で実行させて取得する。つまり、バス調停による遅延は発生しない状況でのトレースを取得する（例えば、1CPU 上で実行した場合のトレースを用いる）。結果として、トレースには、プロセスがある入力を得てから出力するまでの時間と、その間のメモリアクセス回数が含まれ、サイクル精度でのメモリアクセス時刻は含まれない。

3.1 モデル定義

メモリアクセスにおけるバス調停遅延時間の確率的数学モデルを構築するに当たっての前提について述べる。モデルでは、ある時間範囲についてのみ考える。メモリアクセスの主体は、複数の PE である。メモリアクセスレイテンシは、一定であり、1 単位時間である。時間範囲内において、各 PE からは 1 回ずつのみメモリアクセスが発生する。時間範囲内の時刻 t におけるメモリアクセス確率密度関数は、全 PE で同じであり、 $f(t)$ で表される。バス調停遅延時間とは、「ある PE（以降、注目 PE）がメモリアクセスを要求してから実際にアクセス開始するまでの時間」である。以上の前提をもとに、全 PE 数と、注目 PE がメモリアクセスを要求する時刻 t をもとに、注目 PE がバス調停により被る遅延時間が x となる確率を算出する数学モデルを構築した。

確率的数学モデルが想定するメモリ衝突の状況およびバス調停遅延時間の計算方法を図 3 により説明する。図 3 では、全プロセッサ数が 4 である場合の、メモリ衝突の状況を示している。注目するプロセッサからのメモリアクセス（注目アクセス）要求が時刻 t に発生するとし、それ以前に他の 3 つのプロセッサからもメモリアクセス要求がある。時刻範囲 $[t-3, t-2]$ 内に発生するアクセス要求のうち、先に要求が発生した“アクセス 1”が、まず最初にメモリアクセスを行い、次にアクセス要求のあった“アクセス 2”は調停遅延によりアクセス 1 の完了まで待たされる。アクセス 1 が完了すると、アクセス 2 が続いてメモリアクセスを行う。各メモリアクセスのレイテンシは 1 単位時間であると仮定しているため、アクセス 2 の完了は時刻範囲 $[t-1, t]$ 内である。さらに、“アクセス 3”の要求がアクセス 2 中に発生し、アクセス 2 に続いてアクセスを行う。アクセス 3 は $[t-1, t]$ 内にメモリアクセスを開始するため、注目アクセスの要求時刻 t をまたぐ。結果として、注目アクセスはアクセス 3 と衝突し、バス調停によりアクセス 3 の完了まで時間 x だけ待たされる。

基本的な考え方は、以下のとおりである。図 3 中の注目アクセスは、直接的にはアクセス

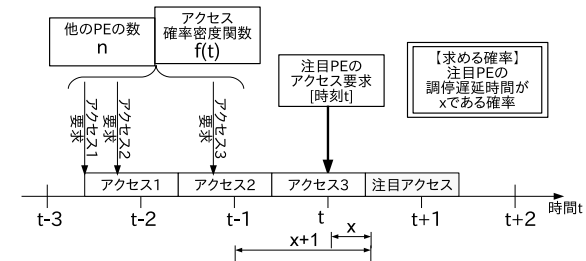


図 3 想定するメモリ衝突の状況

Fig. 3 Assumed situation of memory conflicts.

3 に x だけ待たされている。注目アクセスがアクセス 3 に待たされるのは、図 3 のように時刻 $t-1$ 以前から要求があり、かつ時刻 t をアクセス 3 がまたぐ場合か、もしくは時刻範囲 $[t-1, t]$ 内にアクセス 3 の要求が発生する場合である。時刻範囲 $[t-1, t]$ 内にアクセス 3 の要求が発生する場合の確率は、アクセス 3 のアクセス確率 $\int_{t-1}^t f(t)dt$ に等しい。一方、時刻 $t-1$ 以前から要求があり、かつ時刻 t をアクセス 3 がまたぐ場合は、新たに仮想的な注目アクセス（仮想注目アクセス）を時刻 $t-1$ に想定すると、仮想注目アクセスが $x+1$ だけ待たされる場合に等しい。つまり、仮想注目アクセスを 1 単位時間ずつ過去にずらしながら、再帰的に待ち時間を計算していけば、求める確率が得られる。

以下では、FCFS (First-Come First-Served)、固定優先度 (FP: Fixed Priority)、ラウンドロビン (RR: Round Robin) の 3 つの調停方式それぞれについて、確率的数学モデルの定義を示す。数学モデルは、FCFS のモデルを基本とし、それを拡張することで FP, RR に対応させている。

まず、FCFS のモデル $I_{FCFS}(f, n, k, t)$ および $W_{FCFS}(f, n, k, t, z)$ を示す。式の引数はそれぞれ、 f がメモリアクセス率の確率密度関数、 t が注目する時刻、 n が注目 PE 以外の、時刻 t 以前にアクセス要求を行う PE 数、そして k が時刻 t をまたぐ (t 以前に要求され、 $t-1$ 以降に処理される) メモリアクセスの数を表す。 z は、 k 個のメモリアクセスがある時刻をまたぐ場合に、注目アクセスが待たされる時間（バス調停遅延時間）を表す。ここで、 z は、 $k > 0$ のとき、 k と図 3 中に示した x を用いて $k-1+x$ と表すことができる。 $k = 0$ のときは、 $z = 0$ である。 $W_{FCFS}(f, n, k, t, z)$ が、バス調停遅延時間が z となる確率を与える。 $I_{FCFS}(f, n, k, t)$ は、 k 個のメモリアクセスがある時刻 t をまたぐ確率を与え、 $W_{FCFS}(f, n, k, t, z)$ において $z=0$ となる確率を求めるための補助関数として用いられる。

$$I_{FCFS}(f, n, k, t) = \begin{cases} \left(\int_{t-1}^t f(\tau) d\tau \right)^n, & n = k \wedge 0 < k \\ 1, & n = k \wedge k = 0 \\ \left(\int_{-\infty}^t f(\tau) d\tau \right)^n - \sum_{i=1}^n (I_{FCFS}(f, n, i, t)), & k < n \wedge k = 0 \\ I_{FCFS}(f, n, k+1, t-1) \\ + \sum_{i=1}^k \binom{n}{i} \times I_{FCFS}(f, i, i, t) \times I_{FCFS}(f, n-i, k-i+1, t-1) \\ + \binom{n}{k} \times I_{FCFS}(f, k, k, t) \times I_{FCFS}(f, n-k, 0, t-1), & k < n \wedge 0 < k \end{cases}$$

$$W_{FCFS}(f, n, k, t, z) = \begin{cases} I_{FCFS}(f, n, 0, t), & k = 0 \\ n \times f(t-1+x) \times \left(\int_{t-1+x}^t f(\tau) d\tau \right)^{n-1}, & 0 < k = n \\ W_{FCFS}(f, n, k+1, t-1, k+x) \\ + \sum_{i=1}^{k-1} \binom{n}{i} \times W_{FCFS}(f, n-i, k-i+1, t-1, k-i+x) \times \left(\int_{t-1}^t f(\tau) d\tau \right)^i \\ + \binom{n}{k} \times W_{FCFS}(f, n-k, 1, t-1, x) \times \\ \sum_{j=1}^k \binom{k}{j} \times \left(\int_{t-1}^{t-1+x} f(\tau) d\tau \right)^j \times \left(\int_{t-1+x}^t f(\tau) d\tau \right)^{k-j} \\ + \binom{n}{k} \times \left(\int_0^x W_{FCFS}(f, n-k, 1, t-1, \tau) d\tau + I_{FCFS}(f, n-k, 0, t-1) \right) \\ \times k \times f(t-1+x) \times \left(\int_{t-1+x}^t f(\tau) d\tau \right)^{k-1}, & 0 < k < n \end{cases}$$

次に、FP のモデル $W_{FP}(f, p, n, k, t, z)$ を示す。p は注目 PE のバス上優先度を表す。 $p \in [0, n]$ であり、 $p = 0$ が最高優先度を示す。モデル構築のため、補助関数として I_{FP} , I_{FP_f} , I_{FP_l} , W_{FP_l} の 4 つを定義した。 I_{FP_l} および W_{FP_l} は、注目アクセスが最低優先度の場合のモデルである。 I_{FP_f} は、時刻 t_{pre} 以降に発生するメモリアクセス要求の影響を考慮するモデルである。FP のモデル I_{FP} は、 I_{FP_l} を基本とし、アクセスする並び順の場合分けにより、注目アクセスの優先度が最低でない場合の確率を求める。

$$I_{FP_f}(f, n, k, t_{pre}, t) = \begin{cases} 1, & n = k = 0 \\ \int_t^\infty (f(\tau) d\tau)^n, & k = 0 \\ \sum_{i=1}^n \binom{n}{i} \times \int_{t_{pre}}^t (f(\tau) d\tau)^i \times I_{FP_f}(f, n-i, k-i, t, t+i), & k > 1 \end{cases}$$

$$I_{FP_l}(f, n, k, t) = \begin{cases} I_{FCFS}(f, n, 0, t) + \sum_{i=0}^{n-1} \binom{n}{i} \times I_{FCFS}(f, i, 0, t) \times \int_t^\infty (f(\tau) d\tau)^{n-i}, & k = 0 \\ I_{FCFS}(f, n, k, t) + \sum_{l=1}^k \sum_{i=l}^{n-1} \binom{n}{i} \times I_{FCFS}(f, i, l, t) \times \int_0^1 I_{FP_f}(f, n-i, k-l, t, t+l-1+x) dr, & k > 1 \end{cases}$$

$$W_{FP_l}(f, n, k, t, z) = \begin{cases} I_{FP_l}(f, n, 0, t), & k = 0 \\ W_{FCFS}(f, n, k, t, k-1+x) \\ + \sum_{l=1}^k \sum_{i=l}^{n-1} \binom{n}{i} \times W_{FCFS}(f, i, l, t, t+l-1+x) \\ \times I_{FP_l}(f, n-i, k-l, t, t+l-1+x), & k > 1 \end{cases}$$

$$I_{FP}(f, p, n, k, t) = \begin{cases} I_{FP_l}(f, n, k, t), & p = n \\ 1 - \sum_{i=1}^{p+1} (I_{FP}(f, p, n, i, t)), & p < n \wedge k = 0 \\ 0, & p < n \wedge k > p+1 \\ \sum_{i=p+1}^n (I_{FP_l}(f, n, i, t) \times \frac{(n-p) \times \binom{(n-p)-1}{i-(p+1)}}{\binom{n}{i} \times i}), & p < n \wedge k = p+1 \\ \sum_{i=k}^{(n-p)+k-1} (I_{FP_l}(f, n, i, t) \times \frac{\binom{p}{k-1} \times \binom{n-p}{i-k} \times (n-(i-1))}{\binom{n}{i} \times i}) \\ + I_{FP_l}(f, n, n-p+k, t) \times \frac{p \times \binom{p-1}{k-1}}{\binom{n}{n-p+k} \times (n-p+k)}, & p < n \wedge k < p+1 \end{cases}$$

$$W_{FP}(f, p, n, k, t, z) = \begin{cases} W_{FP_l}(f, n, k, t, z), & p = n \\ icPri(f, p, n, 0, t), & p < n \wedge k = 0 \\ 0, & p < n \wedge k > p+1 \\ \sum_{i=p+1}^n (W_{FP_l}(f, n, i, t, x + (i-1)) \times \frac{(n-p) \times \binom{(n-p)-1}{i-(p+1)}}{\binom{n}{i} \times i}), & p < n \wedge k = p+1 \\ \sum_{i=k}^{n-p+k-1} (W_{FP_l}(f, n, i, t, x + (i-1)) \times \frac{\binom{p}{k-1} \times \binom{n-p}{i-k} \times (n-i+1)}{\binom{n}{i} \times i}) \\ + W_{FP_l}(f, n, n-p+k, t, x + ((n-p+k)-1)) \times \frac{p \times \binom{p-1}{k-1}}{\binom{n}{n-p+k} \times (n-p+k)}, & p < n \wedge k < p+1 \end{cases}$$

最後に、RR のモデル $W_{RR}(f, n, k, t, z)$ を示す。RR は、FP において優先度が実行時に切り替わっていくことと同等と考え、FP のモデルにより各優先度について求めた確率の平均値としてモデル化した。

$$W_{RR}(f, n, k, t, z) = \sum_{p=0}^n \frac{W_{FP}(f, p, n, k, t, z)}{n+1}$$

3.2 数学モデルの妥当性検証

構築した確率的数学モデルの妥当性を検証するため、モデルと同様の前提に基づくモンテカルロシミュレーションプログラムを作成した。シミュレーションプログラムは、複数 PE からのランダムなメモリアクセスにより注目 PE の 1 回のメモリアクセスが待たされる、という状況を 1 回の試行とする。シミュレーションプログラムを一定回数試行した結果の確率分布と、提案するモデルにより計算された確率分布を比較した。比較は、バス調停遅延時間を 0.01 刻みで増加させたときの累積分布関数 (CDF) のプロットにより行った。ここで、累積分布はバス調停遅延時間の増加に伴い、1 に急速に近づいていき、そのままプロットしては変化が分かりにくいので、以降に示す図では、 $\log_{10}(1 - CDF)$ により示す。

図 4 に、各調停方式の、シミュレーション結果とモデル計算結果の累積分布の比較を示す。図 4 中の全グラフは、各 PE のメモリアクセス確率を 0.1 とした場合の結果を示している。まず、図 4(a) および図 4(b) に、FCFS による調停時の待ち時間の累積分布を示す。図 4(a) は全 PE 数が 3 ($n=2$)、図 4(b) は全 PE 数が 4 ($n=3$) の場合である。次に、図 4(c)、図 4(d)、および図 4(e) に、固定優先度方式による調停遅延の、優先度がそれぞれ 0, 1, 2 の場合の累積分布を示す。PE 数は 3 ($n=2$) である。最後に、図 4(f) に、ラウンドロビン方式による調停遅延の累積分布を示す。PE 数は 3 ($n=2$) である。

結果、シミュレーションにより得られた累積分布とほぼ一致する累積分布をモデル計算により得られたことが見て取れた。よって、妥当性を確認することができたと言える。

3.3 高速性能見積もりツールでの使用方法

高速性能見積もりにおいては、プロセスの実行時間 E と、実行中のメモリアクセス回数 N だけが既知である。アクセス確率については、1 回の実行中において一様分布であると仮定し、上記 2 要素を用いて $f(t) = N/E$ とする。複数の PE からのメモリアクセス確率は、モデル上では一律同じ確率であると仮定していたが、実際には一律ではない場合が多いため、注目 PE 以外のプロセス間でアクセス確率を平均して扱うこととする。

高速性能見積もりでのモデル使用にあたり、まず、提案するモデルから、バス調停遅延時

間の期待値 D をアクセス率ごとに事前に計算しておく。次に、見積もり実行時に、複数のプロセスの実行が並列に行われる時点で、プロセスごとに、他プロセスのアクセス率を計算する。バス調停遅延時間を加算した新しいプロセス実行時間を E_{new} とすると、 E_{new} はアクセス率に対応した D とメモリアクセスレイテンシ L 、およびメモリアクセス回数 N を使用して、 $E_{new} = E + D \times L \times N$ により得られる。

4. 評価実験

確率的数学モデルを高速性能見積もりツールに対して適用した効果を評価するため、調停遅延が発生するシステムを FPGA 上で動作させ、性能見積もりを行った。アクセス衝突が起きうる共有メモリは、アクセスレイテンシが 16 サイクルで一定のメモリを用いた。このメモリは、本来 1 サイクルで読み書き可能なメモリ (FPGA の BlockRAM) のレイテンシを 16 サイクルに変更して作成した、モデルの前提に合致する理想的なメモリである。FPGA 上では、プロセッサとメモリは Point-to-Point で接続される。そのためバス調停は発生しないが、共有メモリ側に調停回路が挿入される (スレーブ側調停)。本論文では、バス調停とスレーブ側調停は同じとみなした。調停回路は、ラウンドロビン方式で調停を行う。

調停遅延が発生するシステムとして、メモリアクセスのみを行うプロセスを 4 つ持つシステムを作成した。4 つのプロセスはそれぞれ、指定されたメモリアクセス率に従い、メモリアクセスをランダムなタイミングで行う。作成したシステムは 2 プロセッサ構成である。図 5 に作成したシステムの動作を示す。CPU1 上のプロセス (図 5 中、reader1, writer1) は 1 回の起動につきメモリアクセスを 100 回行う。CPU2 上のプロセス (reader2, writer2) は 1000 回メモリアクセスを行う。CPU1 からの 100 回のメモリアクセスが、CPU2 からのメモリアクセスと衝突しうる状況を作成した。各プロセッサからのメモリアクセス率は、20%, 25%, 33%, 50%, 66%, 80% の 6 種類の指定で評価を行った。

図 6 に、提案する確率的数学モデル (ラウンドロビン版) の適用有無による実行時間見積もり誤差の変化を示す。ここでの誤差とは、プロセス間のメモリアクセス衝突ごと (“reader1” と “reader2” の衝突など) のバス調停遅延時間の見積もり誤差の、平均値である。図中、“w/o model” が確率的数学モデルを適用しない場合、“w/ model” が適用した場合の結果を示している。確率的数学モデルを適用しない場合は、各プロセスを並列性なしに実行したトレースをそのまま用いた、バス調停遅延を無視した見積もり結果である。

図 6 から、確率的数学モデルを適用することで誤差が削減され、見積もり誤差が 0% に近づいていることがわかる。特に、プロセッサからのメモリアクセス率が低い場合には、1%未

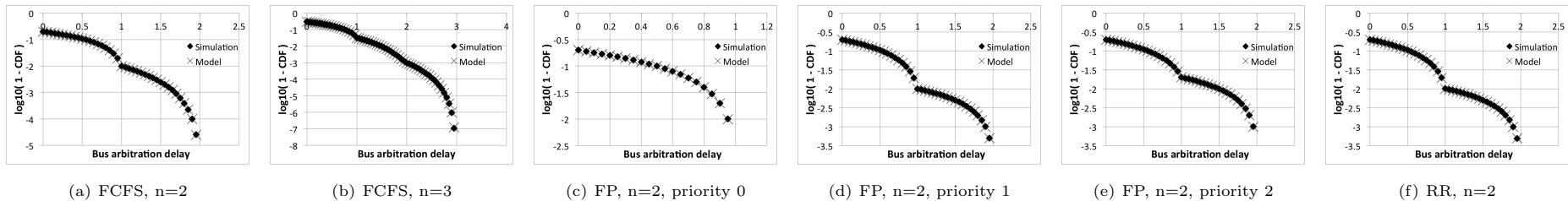


図 4 確率的数学モデルとモンテカルロシミュレーションにより計算された各調停方式における調停遅延時間の $\log_{10}(1-CDF)$

Fig. 4 CDF of bus arbitration delay for three arbitration policies obtained by our models and Monte-Carlo simulation.

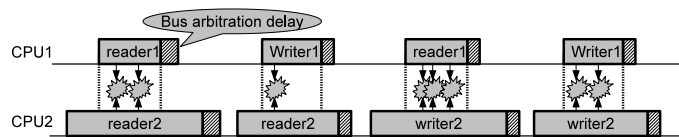


図 5 評価用システムの動作

Fig. 5 Behavior of a system for evaluation.

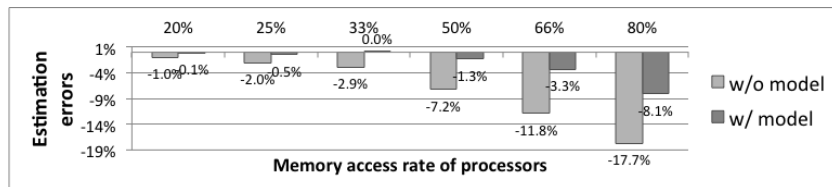


図 6 確率的数学モデル適用有無による見積もり誤差の変化

Fig. 6 Estimation errors of with and without bus arbitration delay models.

満の誤差となり、効果が確かに表れている。一方、アクセス率が 50% を超えた場合には、アクセス率が上がるに連れてモデル適用を行っても誤差が効果的に削減されなくなっている。これは、アクセス率が高まったことで、複数回のメモリアクセス間での影響が現れてきたためであると考えられる。アクセス率が高い場合、共有メモリへのアクセスが隙間なく処理されるようになり、遅延時間に対するアクセスのランダム性の影響が小さくなっているためであると考えられる。

5. おわりに

本論文では、トレースベースシミュレーションにおいてメモリ衝突時のバス調停遅延時間を見積もるための確率的数学モデルを提案した。モンテカルロシミュレーションとの比較に

より妥当性を確認し、さらに、FPGA 上で動作するシステムで発生するバス調停遅延時間を見積もった。FPGA 上のシステムの見積もりでは、アクセス率が 50% 以下の場合に確率的数学モデルを用いることで性能見積もり誤差を 1% 未満に抑えられることを確認した。さらに、アクセス率が 50% を超える場合には誤差が大きくなることを確認した。

今後は、アクセス率が高い場合、およびアクセスタイミングにランダム性が無い（規則的な）場合に対応し、多様なシステムに適用可能なバス調停遅延時間見積もりを実現することを考えている。また、メモリ特性を考慮に入れたバス調停遅延時間の見積もりも課題である。

謝辞 本研究の一部は半導体理工学研究センター（STARC）との共同研究による。

参考文献

- 1) Mahadevan, S., et al.: ARTS: A systemc-based framework for multiprocessor systems-on-chip modelling, Design Automation for Embedded Systems, (2007).
- 2) Wild, T., et al.: TAPES - trace-based architecture performance evaluation with SystemC, Design Automation for Embedded Systems, vol.10, no.2-3, pp.157-179 (2005).
- 3) Bobrek, A., et al.: Stochastic Contention Level Simulation for Single-Chip Heterogeneous Multiprocessors, IEEE Trans. on Computers, vol. 59, no. 10, pp. 1402-1418 (2010).
- 4) 本田晋也ら: システムレベル設計環境: SystemBuilder, 電子情報通信学会論文誌, vol.J88-D-I, no.2, pp.163-174 (2005).
- 5) Shibata, S., et al.: Efficient Design Space Exploration at System Level with Automatic Profiler Instrumentation, IPSJ Trans. on System LSI Design Methodology, vol. 3, pp.179-193 (2010).
- 6) 柴田誠也ら: システムレベル設計探索のための高速性能見積もり手法, DA シンポジウム 2010, Vol.2010, No.7 (2010).