情報処理学会論文誌　コンシューマ・デバイス & システム　Vol.5　No.3　11–18 (June 2015)

研究論文

CPU使用率とメモリ帯域使用率を考慮した性能予測手法

若林 昇[1,a)]　吉岡 信和[2]

受付日 2014年7月28日, 採録日 2015年3月26日

概要：デジタルテレビを筆頭にコンシューマ機器は年々高機能化の一途にある．一方で，価格面での競争激化のため，コストダウンは必須となっており，高性能な CPU，潤沢なメモリは期待できない．したがって，現行の性能を限界まで引き出す必要があり，そのためには，定量的な性能測定と性能予測手法が必要になる．本論文では，CPU 使用率とメモリ帯域使用率を考慮した性能予測手法を提案する．

キーワード：性能測定，メモリバンド，同時動作，性能予測

A Performance Prediction Technique in Consideration of CPU Utilization and Memory Band Utilization

Noboru Wakabayashi[1,a)]　Nobukazu Yoshioka[2]

Received: July 28, 2014, Accepted: March 26, 2015

Abstract: Consumer products, such as Digital-TV, have been becoming high functionality year by year. On the other hand, the cost reduction is required because of the competition intensification on the price side, without using rich specification hardware (CPU, memory, etc). Thus, it is needed to draw current performance to the limit. After all, the performance prediction technique is required. This paper proposes a performance prediction technique in consideration of CPU utilization and memory band utilization.

Keywords: performance measurement, memory band, simultaneous behaviour, performance prediction

1. はじめに

コンシューマ機器など組込み機器の多くは，年々高機能化の一途にあり，新機能が追加される．一方で，価格面での競争激化のため，コストダウンは必須となっている．ソフトウェアの観点からは，より高性能な CPU やより大容量なメモリであれば，新機能を追加する際に性能面を考慮する必要がなくなる．しかし，コストアップになるこれらのハードウェアの強化をしないために，複数の新機能と既存機能の同時動作時の性能予測値による見積りが必要になる．また，開発の後工程で性能問題が発覚すると大きな手戻りになるため，性能見積りは新機能開発時の早い段階で必要になる．なお，大規模な組込み機器では，複数の新機能を並行して開発することが多い．このような複数の新機能開発時の早い段階では，それぞれの新機能を同時に動作させることは困難になるため，実際に同時動作させて性能測定することはできない．

従来，同時動作時の性能予測値は各機能の CPU 使用率を加算して算出していた．しかし，デジタルテレビなど映像を扱うような機器では高画質映像などの多量のデータを扱う場合，メモリ帯域を多く使用する．このような映像を扱う機器で，バスアービタなどによるバスの調停機構がある機器の場合，デコーダなど CPU 以外の機器がバスを優先使用することがあり，この場合 CPU はメモリアクセスを待たされるため，CPU 使用率が増えるように見える．すなわち，CPU 使用率とメモリ帯域使用率には因果関係があるといえる．性能予測の際は，メモリ帯域使用率も考慮

[1] 株式会社日立製作所研究開発グループシステムイノベーションセンタ
Hitachi, Ltd., Research & Development Group, Center for Technology Innovation – Systems Engineering, Yokohama, Kanagawa 244–0817, Japan
[2] 国立情報学研究所/総合研究大学院大学
National Institute of Informatics/The Graduate University for Advanced Studies, Chiyoda, Tokyo 101–8430, Japan
[a)] noboru.wakabayashi.mu@hitachi.com

した性能予測が必要になる．

メモリ帯域使用率を考慮したCPU使用率の算出方法として，メモリストール時間を計測し，CPU使用率を算出する方法がある．これはCPU時間を，実行クロック数とメモリストールクロック数の和にクロックサイクル時間を乗じて算出する方法である．このうちメモリストールクロック数はプログラムあたりのメモリアクセス命令件数と1メモリアクセス命令あたりのミス率（キャッシュミス率）とミスペナルティを乗じて算出することができるが，キャッシュミス率を算出するためには，キャッシュミス回数を計測する必要がある．そして，キャッシュミス回数を計測するためには，CPU内部にパフォーマンスカウンタ（ハードウェアカウンタ）が搭載されている必要がある．ここでパフォーマンスカウンタとは，CPUに内蔵した特別なレジスタであり，キャッシュミスや分岐予測ミスといったハードウェアに関する動作の回数を格納する．しかし，低コストが要求されるような組込み機器では，MIPSなどパフォーマンスカウンタが内蔵されていないCPUを使用することがあり，メモリストール時間を用いたこのような従来手法を用いることができない．

本論文では，パフォーマンスカウンタが付いていないCPUを搭載する機器でも，チップセレクト信号の計測からメモリ帯域使用率を測定し，CPU使用率との関係を導き出すことで，同時動作時の性能予測を行う手法について提案する．

続く2章では，従来手法となる関連研究とその課題について述べ，3章で本研究の対象となる機器構成について説明する．4章では，従来手法の課題を解決する手法について提案し，5章で提案手法を評価する．6章で提案手法に関する議論を行い，最後に7章でまとめる．

2. 関連研究

システムの性能予測に関連する研究として，ハードウェアとソフトウェアの協調シミュレータを用いたハードウェア開発環境の研究 [1], [2] などがある．これらの研究により，ハードウェア実装前にシステム全体の性能予測が可能となった．しかし，コンシューマ機器では，製品開発中にハードウェアが変更されることが多く，このようなシミュレータは最終的な製品に搭載されるハードウェアをシミュレートすることは少なく，最終製品の性能予測ができるとはいいきれない．

また，性能予測に関連する他の研究として，プログラムコード抽象化手法に基づくシステムの性能評価環境PSI-NSIM [3] がある．これはインターコネクトのシミュレーションを行うことで，システム全体の性能を高速かつ精度良く見積もり，大規模システムの効果的な性能解析支援や可視化を実現するものである．しかし，このような大規模システムのシミュレーションは，コンシューマ機器のような低コストが要求される機器には適用が難しい．

モデル検査を用いた性能予測に関する研究として，時間制約の検証ツールや確率的モデル検査ツールを用いた研究がある [4], [5], [6], [7]．これらは，現在最も普及している性能モデル検証ツールであるUPPAAL [8] や確率的振舞いを持つシステムに対するモデル検査ツールPRISM [9] を用いて，実時間ネットワークシステムの詳細なモデルからシステム全体の性能解析を行うものである．しかし，このようなモデル検査ツールを用いた性能予測では，どのようにモデル化するかが重要であり，メモリ帯域使用率とCPU使用率の関係をモデル化する必要があるが，これらの関係については述べていない．

CPU使用率とメモリ帯域使用率の関係に関する研究として，バス調停に関する研究やメモリウォール問題に関する研究などがある．

バス調停に関する研究では，ラウンドロビン方式によるバス調停方法 [10]，静的固定優先度方式によるバス調停方法 [11]，TDMAによるバス調停方式 [12]，TDMAとラウンドロビンを組み合わせた方法によるバス調停方式 [13], [14]，LOTTERYBUS方式によるバス調停方式 [13], [14]，Slack-based Bus方式によるバス調停方式 [12] などがある．しかし，これらの研究では性能予測に関して，メモリ帯域使用率とCPU使用率の関係について考慮されておらず，メモリ使用率とCPU使用率を用いた性能予測方法については提示されていない．

メモリウォール問題に関する研究では，与えられたハードウェア資源制約下においてこれらを最大限に有効活用し，システム全体の性能を向上する提案 [15] がある．また，動的にキャッシュラインサイズを変更しシステムの性能を向上する提案 [16] がある．これらの提案の中ではメモリストール時間を考慮したプログラムの実行時間（CPU使用率）の算出方法が提示されている．メモリストール時間を算出するためには，キャッシュミス回数を計測する必要があり，キャッシュミス回数を計測するためには，CPUにパフォーマンスカウンタが内蔵されている必要がある．

また，パフォーマンスカウンタを用いた性能予測手法の研究 [17] がある．これは，あらかじめ多数のプログラムを実行し，統計的な学習を行い，あるパフォーマンスカウンタの値と性能の間の関係を回帰分析により求める定量的な手法である．しかし，コンシューマ機器のような低コストを要求されるような機器では，CPUにパフォーマンスカウンタが付いていないことがある．この場合，このような従来手法を用いることができない．

3. 対象機器構成

本研究における性能評価の対象となる機器構成の例を**図 1** に示す．

図1で示すとおり，RAMなど外付けの外部メモリとCPU，それ以外のたとえばデコーダやネットワークコン

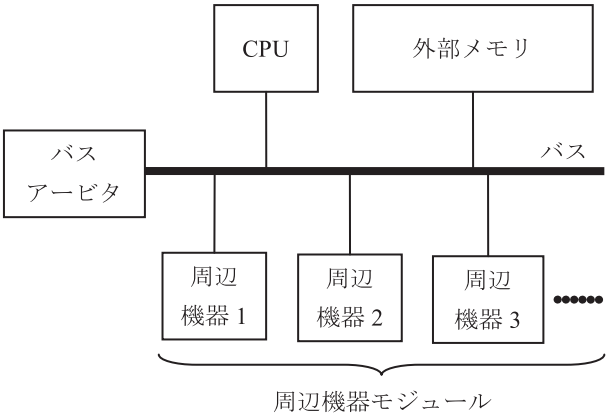


図 1　対象機器構成例
Fig. 1　Target configuration example.

トローラ（Network Interface Controller：NIC），HDD や SSD といった 2 次記憶装置など数多くの周辺機器モジュールがバス上で接続されており，各モジュールからのメモリアクセス要求をバスアービタが調停する機器構成を対象としている．このように周辺機器がバスに接続する機器構成は一般的であり，またバス調停を行うバスアービタが接続される構成については，映像などの多量のデータがバス上を通信する機器で多く見ることができる．

4. CPU 使用率とメモリ帯域使用率を考慮した同時動作性能予測手法

4.1 概要

本章では，CPU にパフォーマンスカウンタが付いていない機器でも，メモリ帯域使用率を考慮した同時動作時の性能予測ができる手法について述べる．

CPU にパフォーマンスカウンタが付いていない機器でも，図 1 で示した対象機器構成のとおり，DDR2-SDRAM など外部メモリは接続されている．

提案手法では，まず，SDRAM のチップセレクト信号の単位時間あたりのパルス数（周波数）を計測し，理論的な最大値との比率からメモリ帯域使用率を算出する．また，メモリ帯域への負荷を変えて，CPU 使用率を計測し，関連性を算出する．さらに，各単体機能の CPU 使用率およびメモリ帯域使用率の測定値と，基準点の CPU 使用率およびメモリ帯域使用率の測定値から，算出した CPU 使用率とメモリ帯域使用率の関連性を考慮する．これにより，各機能が同時動作した際の予測値を算出する式を提案する．

なお，図 1 で示した対象機器構成では，周辺機器モジュールは画像情報などを外部メモリに転送することを想定しており，本手法もこのような環境での予測を対象としている．このような環境では CPU のキャッシュを介することがないので，データキャッシュの影響はほとんどなく，また命令実行についてもメモリ待ちの影響の方が大きくなるため，命令キャッシュの影響もほとんどない．

4.2 チップセレクト信号を用いたメモリ帯域使用率測定方法

4.1 節で述べたように，性能予測する際は，メモリ帯域使用率まで考慮する必要がある．本節では，メモリ帯域使用率の測定方法について述べる．

本論文の対象機器構成において，DDR2-SDRAM など外部メモリにつながるバス上には，CPU のほかにもデコーダやネットワークコントローラ（Network Interface Controller：NIC），HDD や SSD といった 2 次記憶装置など数多くのモジュールが接続されており，各モジュールからのメモリアクセス要求をバスアービタが調停する構成になっている．しかし，どのモジュールがどれだけバスを占有しているのかリアルタイムに測定する仕組みは用意されていないことが多い．このため，従来は机上で見積もった理論的な最大値などを積み重ねることでワーストケースでの性能を予測していた．

しかしこの方法では，アービタの調停動作などを含めて実際にどう動いているのかの挙動把握をすることはできず，したがって，平均値を知ることもできなかった．

そこで，リアルタイムにメモリ帯域使用率を測定するために，本手法では，DDR2-SDRAM へ出力しているチップセレクト信号の単位時間あたりのパルス数（周波数）によってメモリの負荷（使用率）を近似する．なお，チップセレクトはコンピュータシステムにおける大半のデバイスに存在するピンであり，CPU がどのデバイスに対して読み出しや書き込みを行うのかを指定する際に利用する．チップセレクト信号がアクティブになっている間，そのデバイスは動作可能状態になる．DDR2-SDRAM のチップセレクト信号は DDR2-SDRAM にアクセスする際に必ずアクティブになるので，本手法ではこの点に着眼し，メモリの負荷の近似としてチップセレクト信号を利用することにした．

本手法は，チップセレクト信号を周波数ドメインアナライザなどで観測することにより，周波数の動的な変化と平均値をリアルタイムに計測するとともに，チップセレクト信号の周波数の理論的な最大値との比率からメモリバンド幅の使用率を算出する手法である．**図 2** にデジタルテレビにおける視聴機能と録画機能を同時に動作させたときのチップセレクト信号パルスの周波数の変化の一例を示す．横軸は時間 (ms)，縦軸は周波数 (MHz) である．図 2 のチップセレクト信号パルスが High の状態の間，メモリアクセスがあったということになる．また図 2 の例では約 3 ms 周期でメモリアクセスしていることが分かる．

DDR メモリでは，メモリ内容（コンデンサに蓄えられた電荷）を保持するために，周期的なリフレッシュ（再び電荷を注入する動作）が必要となる．このリフレッシュの影響として，リフレッシュ期間は他のメモリアクセスは実行できず実質的にはバンド幅負荷となる．しかしながら，チップセレクトの周波数にはこのリフレッシュの影響は

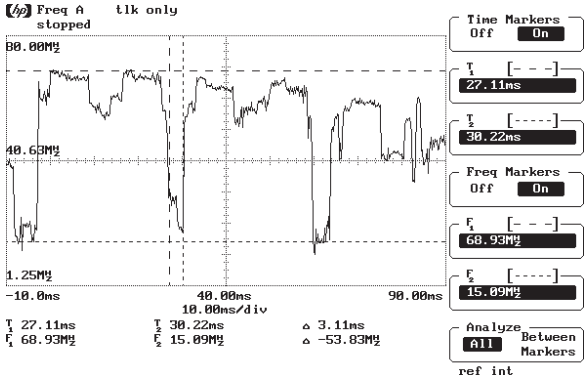


図 2　デジタルテレビにおける視聴と録画の同時動作時のチップセレクト信号パルス周波数変化

Fig. 2　Chip select pulse of displaying and recoding on DTV.

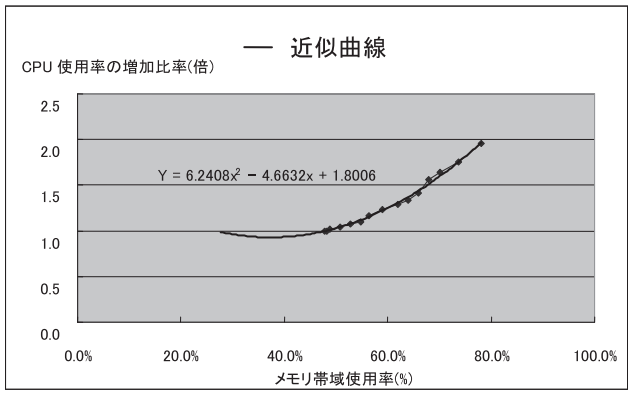


図 3　バンド幅の増加分が CPU 負荷率に与える影響

Fig. 3　The influence that increase of the band width gives in CPU load.

ほとんど反映されないため，その分を補正する必要がある．これらを考慮し，チップセレクトの周波数の平均値を F_{mean} [MHz]，理論的な最大値を F_{max}，単位時間あたりのリフレッシュ期間を R [%] としたときのメモリ帯域使用率 U_m [%] を（式 (1)）で算出する．

$$U_m[\%]=(F_{mean}[\mathrm{MHz}]+F_{max}[\mathrm{MHz}]\times R[\%])/F_{max}[\mathrm{MHz}] \times 100 \quad (1)$$

4.3　CPU 使用率とメモリ帯域使用率の関係

本節ではメモリ帯域使用率が CPU 使用率に及ぼす影響について述べる．4.2 節でも述べたように，DDR はリフレッシュ動作を行っている．この周期は設定により変更することができるので，リフレッシュ周期を意図的に変えることにより，ダミーのメモリバンド負荷として利用できる．

このようにしてダミー負荷の量を変えながら CPU 使用率を測定することで，メモリ帯域使用率と CPU 使用率の関係を導き出すことが可能である．なお CPU 使用率の測定は，各種ツール[*1]があるのでそのツールを用いればよい．なお，これらのツールは，一定時間内に CPU がアイドル状態になった時間を測定して CPU 使用率を測定しており，実際の CPU 負荷（CPU 計算時間）を測定しているものではない．詳細には，どのタスクも実行されていない状態に呼ばれる関数内でカウンタ値をインクリメントし，一定時間内に CPU アイドル状態になった割合をカウンタ値から算出するものである．そのため，ダミー負荷を与えると，アイドル状態のカウンタ値がインクリメントされず CPU 負荷が高くなると想定できる．

デジタルテレビに対して，リフレッシュ周期を意図的に変えてメモリ帯域に負荷を与えたときの CPU 使用率の測定結果のグラフを**図 3** に例示する．

図 3 のグラフの縦軸は単位時間あたりのリフレッシュ期間 1%のときの CPU 使用率を 1 としたときの CPU 使用率の比率であり，横軸はリフレッシュ頻度を変化させたときに式 (1) で求められるメモリ帯域使用率である．

本手法では，測定した値をグラフにプロットし，近似曲線を求める．図 3 の例では 2 次曲線で近似でき，CPU 使用率の増加比率 Y はメモリ帯域使用率 x [%] から以下の式 (2) の近似式が得られる．式 (2) においてそれぞれの係数は，a = 6.2408，b = −4.6632，c = 1.8006 となる．なお，式 (2) および a，b，c の係数は例である．プロットした値によっては a，b，c の係数は変わり，また 2 次曲線ではなく他の曲線で近似することもありうる．

$$Y = f(x) = ax^2 + bx + c \quad (2)$$

図 3 のグラフから分かるように，同じソフトウェアを同じ CPU で実行していても，メモリ帯域が約 70%の使用率になると CPU 使用率が約 1.5 倍になり，メモリ帯域が約 80%になると CPU 使用率が 2 倍となる．これはメモリ帯域に負荷がかかるため，CPU がメモリへのアクセスを待たされる結果，CPU が動けない，すなわち，ある処理を行うのに時間がかかってしまい，メモリ帯域の負荷が高いほど，CPU 使用率が高くなるということである．

このように，CPU 使用率とメモリ帯域使用率は依存関係にある．そのため，純粋な CPU 使用率がどれくらいあるのかは，CPU 使用率測定ツールから得られた値よりメモリ帯域負荷による影響部分を排除しなければならない．

4.4　CPU 使用率予測値とメモリ帯域使用率予測値の算出方法

本節では，複数機能の同時動作時における CPU 使用率およびメモリ帯域使用率の予測値を算出する方法について述べる．

4.3 節でも述べたように，CPU 使用率測定ツールから得られた値よりメモリ帯域負荷による影響部分を排除しなければならない．まずは，このメモリ帯域負荷の影響を取り除いた CPU 使用率の計算方法について述べる．

多くの CPU 使用率測定ツールは，システム全体あるいは

[*1] vmstat や top といったオープンソースソフトウェアのコマンドがある．

プロセス単位の CPU 使用率が測定される．そのため，ある単体機能の純粋な CPU 使用率を測定したい場合，まずは基準となる機能の CPU 使用率およびバンド帯域使用率を測定する必要がある．本論文では，デジタルテレビに対して，デジタル番組視聴中でかつ，バスのアービタの設定を CPU 最優先とした場合の値をデジタルテレビの例における基準点とするようにした．ここで，基準点の CPU 使用率を U_{cb} [%] とし，ある単体機能の CPU 使用率を U_{cx} [%]，メモリバンド幅使用率を U_m [%] とすると，該単体機能の純粋な CPU 使用率 U_{cp} [%] は式 (3) で算出する．第 1 項は CPU 使用率測定ツールから得られた CPU 使用率からメモリ帯域使用率の影響を反映した CPU 使用率になり，これに第 2 項である基準機能の CPU 使用率を差し引く．

$$U_{cp}\,[\%] = U_{cx}/f(U_m) - U_{cb} \tag{3}$$

また，基準点のメモリバンド幅使用率を U_{mb} [%] とすると，この単体機能のメモリバンド幅使用率 U_{mp} [%] は，CPU 使用率からの影響はないため，式 (4) のように単純に差分をとることで算出する．

$$U_{mp}\,[\%] = U_m - U_{mb} \tag{4}$$

次に，これらの数式を用いた複数機能の同時動作時の性能予測値算出方法について述べる．本手法では，以下の手順で算出する．

(1)　**単体機能の CPU 使用率およびメモリ帯域使用率を測定（CPU 使用率測定ツールおよび 4.2 節のメモリ帯域使用率測定方法で計測）．**
(2)　**基準点の CPU 使用率およびメモリ帯域使用率を測定（CPU 使用率測定ツールおよび 4.2 節のメモリ帯域使用率測定方法で計測）．**
(3)　**(1)，(2) で得られた値を，式 (3)，式 (4) に代入して，単体機能の純粋な CPU 使用率およびメモリ帯域使用率を算出．**
(4)　**(1) の機能の組合せについて，(3) で算出した値から，同時動作時の CPU 使用率およびメモリ帯域使用率を算出．**

(4) において，ある機能 1 の純粋 CPU 使用率を U_{cp1} [%]，単純メモリ帯域使用率を U_{mp1} [%]，別の機能 2 の純粋 CPU 使用率を U_{cp2} [%]，単純メモリ帯域使用率を U_{mp2} [%]，基準点の CPU 使用率を U_{cb} [%]，メモリ帯域使用率を U_{mb} [%] とすると，機能 1 と機能 2 の同時動作時のメモリ帯域使用率 U_{mt} [%] および CPU 使用率 U_{ct} [%] は，それぞれ以下の式 (5)，式 (6) のように算出する．

$$U_{mt}\,[\%] = U_{mb} + U_{mp1} + U_{mp2} \tag{5}$$

$$U_{ct}\,[\%] = (U_{cb} + U_{cp1} + U_{cp2}) \times f(U_{mt}) \tag{6}$$

5.　評価

4 章で提案した同時動作性能予測手法について，妥当性を評価する．HDD への録画機能が搭載されたデジタルテレビの実機に対して，提案手法を用いた同時動作時の性能予測を行った．対象となるデジタルテレビのブロック図の概要を**図 4** に示す．

DDR2-SDRAM など外部メモリにつながるバス上には，CPU のほかに，MPEG2 や H.264 の放送波圧縮画像データをデコードするデコーダ，画質を高画質化する高画質化エンジン，録画データなどを記憶する HDD など，主に画像処理に関わるハードウェアモジュールが接続されており，各モジュールからのメモリアクセス要求をバスアービタが調停する構成になっている．なお，バス上に存在するバスマスタがバスを使用できるのは，通常同一バス上で 1 つであり，同時に複数のバスマスタがバスを使用することはできない．バスアービタとは，バス上に存在する機構であり，バスマスタからのバス使用権要求を優先度などによって調停する機能を持つ．

このような構成のデジタルテレビにおいて，同時動作を行う対象機能を下記に示す．

- 放送波表示
- 放送番組の HDD 録画
- 放送番組の HDD 録画（TS→XP ダウンコンバート）
- 録画番組の HDD 再生

これらの各単体機能に対して，4.4 節で述べた手順 (1)〜(3) の結果得られる基準点から比較した純粋 CPU 使用率とメモリ帯域使用率を**表 1** に示す．なお，基準点は，4.4 節でも示したとおり，デジタル番組表示中でかつ，バスのアービタの設定を CPU 最優先（通常はデコーダ優先）とした場合の値を基準点とするようにした．

これらの機能について，下記の組合せで同時動作させた場合の，提案手法による予測値と，実際に同時動作させて得られた測定値を比較することで，提案手法の妥当性を検証する．

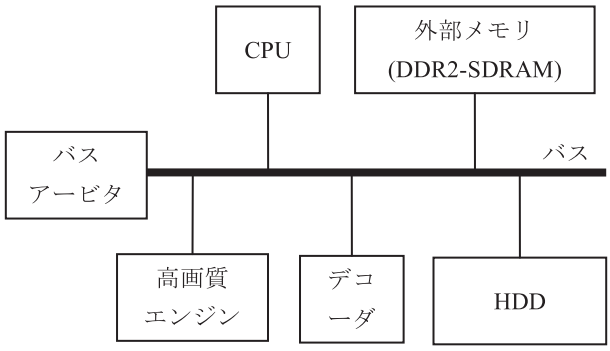


図 4　評価対象となるデジタルテレビのブロック図（概要）
Fig. 4　Block diagram of the DTV targeted for an evaluation.

表 1　対象機能の CPU 使用率およびメモリ帯域使用率（基準点からの差分）
Table 1　The CPU utilization and memory band utilization of the target function.

機能	CPU 使用率 [%]	メモリ帯域使用率 [%]
放送波表示	3.9	3.0
HDD 録画	17.7	0.0
HDD 録画（ダウンコンバート）	12.1	0.0
HDD 再生	8.1	1.8

表 2　同時動作予測方法の検証
Table 2　Verification of suggestion method.

同時動作	提案手法(予測値)		測定値	
	メモリ帯域使用率 [%]	CPU 使用率 [%]	メモリ帯域使用率 [%]	CPU 使用率 [%]
放送波表示＋HDD 録画	49.6	68.4	49.6	65.6
放送波表示＋HDD 録画（ダウンコンバート）	49.6	62.7	48.1	64.6
HDD 録画＋HDD 再生	48.4	71.5	48.1	72.9
放送波表示＋放送波表示	52.6	57.0	55.6	59.3

表 3　同時動作予測方法の検証（誤差）
Table 3　Verification of suggestion method (Error).

同時動作	絶対誤差		相対誤差	
	メモリ帯域使用率 [%]	CPU 使用率 [%]	メモリ帯域使用率 [%]	CPU 使用率 [%]
放送波表示＋HDD 録画	0	2.8	0	4.1
放送波表示＋HDD 録画（ダウンコンバート）	1.5	1.9	3.1	2.9
HDD 録画＋HDD 再生	0.3	1.4	0.6	2.0
放送波表示＋放送波表示	3.0	2.3	5.4	3.9

- 同時動作（1）:「放送波表示」＋「HDD 録画」
- 同時動作（2）:「放送波表示」＋「HDD 録画（ダウンコンバート）」
- 同時動作（3）:「HDD 録画」＋「HDD 再生」
- 同時動作（4）: 2 画面再生（「放送波表示」＋「放送波表示」

上記同時動作（1）～（4）に対する提案手法（4.4 節で述べた手順 (4)）による予測値と，実機による測定値を**表 2** に示す．

また，提案手法による予測値と実際の測定値との誤差（絶対誤差および相対誤差）を**表 3** に示す．

表 3 に示したとおり，絶対誤差で見ると，メモリ帯域使用率および CPU 使用率ともにほぼ同じ値になっており，相対誤差で見た場合でも，0～5.4%であり，性能的にはほぼ同等といえる．なお，測定ツールを用いて 20 秒間の「放送波表示 + HDD 録画」の CPU 使用率を計測したところ 10.6%の計測幅があり，同様に「放送波表示 + HDD 録画」のメモリ帯域使用率では 10.2%の測定幅があることから，提案手法による予測誤差は十分に小さいといえる．これにより，提案する同時動作の性能予測手法は妥当であると考える．

6. 議論

6.1 提案手法の一般性

5 章の評価では，同時動作時の実機による測定値をとる必要があったため，すでに同時動作ができる実装済みの機能で評価したが，本手法を用いると，新機能開発時などの新機能など，実際に同時動作させることができなくても，すでにある環境の測定値と，新機能単体の動作が可能な環境での測定値があれば，4.4 節で述べた手順により，同時動作時の予測値を得ることができる．

また，本論文では，デジタルテレビを例にしたが，図 1 で示した対象機器構成で，下記の特徴を持つ機器に対しては，提案手法は有効であると考える．

- バスアービタなどによるバスの調停機構がある．
- バス上で多量のデータを扱う（バスの負荷が高い）．
- パフォーマンスカウンタがない．
- SDRAM メモリ（チップセレクト信号）を使用．

このような特徴を持つ機器としては，映像を扱う機器で多く見ることができる．たとえば，評価対象としたデジタルテレビ以外でも，ビデオや液晶プロジェクタ，スキャン機能を有するマルチファンクションプリンタなどのコンシューマ機器に適用できる．また，コンシューマ機器以外でも，CT スキャンや MRI などの高画質データを扱う医療系機器でも適用可能であると考える．また，今後 M2M (Machine to Machine) や IoT (Internet Of Things) といった世界で用いられる組込み機器でも，ビックデータを扱う場合は多量のデータがバスを使用するため，本提案手法を適用できると考えており，このような組込み機器は今後増えてくると予想している．

図 1 で示した対象機器構成例では，周辺機器が 3 つの場合であったが，本手法では，メモリ帯域を実測するため，3 つ以上の場合であっても適用できると考えている．また，図 1 で示した対象機器構成例では，CPU が 1 つの場合で

あったが，2 個以上のマルチコア構成でも本手法の基本的な考え方は適用できると考えている．ただし，この場合，各コアからのチップセレクト信号をどのように考慮するかを検討する必要がある．

6.2　提案手法の限界

本提案手法による性能予測値の使い方には留意が必要である．性能予測値が 100%を超えないと問題ないように思えるが，適用対象の機器によっては，100%に達していなくても，ある閾値を超えると，操作性や応答性に影響が出る場合がある．

たとえば，デジタルテレビの場合，CPU 使用率が高くなると，リモコンによるチャネル切替えや音量調整操作に対する反応が遅くなることがある．これは，チャネル切替えや音量調整操作の分 CPU 使用率が増え，瞬間的に CPU 使用率が 100%を超えることが起こるためである．また，メモリ帯域使用率が高くなると，ブロックノイズなどの映像破綻が発生する可能性がある．これも映像データの転送が間に合わず，バッファーアンダフローを起こしてしまうことが原因である．

そのため，同時動作の可否を判断する際は，これらの不具合が出ないレベルの閾値を設け，その閾値を超えないことで，判断する必要があると考える．なお，この閾値の設定方法については現状過去の経験によるところが大きく，人手によって設定する必要がある．そのため，今後はこの設定を自動化することが望ましい．

7.　おわりに

7.1　結果

CPU にパフォーマンスカウンタが付いていない機器でも，メモリ帯域使用率を考慮した同時動作時の性能予測ができるチップセレクト信号を用いた性能予測手法について提案した．

提案手法では，チップセレクト信号からメモリ帯域使用率と CPU 使用率の関連性を算出し，さらに，各単体機能の CPU 使用率およびメモリ帯域使用率の測定値と，基準点の CPU 使用率およびメモリ帯域使用率の測定値から，各機能が同時動作した際の予測値を算出する式を提案した．

また，HDD への録画機能が搭載されたデジタルテレビの実機に対して，提案手法を用いた同時動作時の性能予測を行った．提案手法による性能予測値と，実際に測定した値と比較し，ほぼ同じ値になることを確認し，本提案手法の有効性を確認した．これにより，以下の結果を得た．

- チップセレクト信号によるメモリ帯域使用率の算出方法は複数機能同時動作時の性能予測に有効である．
- リフレッシュ周期を変えたときのメモリ使用率と CPU 使用率の実測値から得られる近似式で関係性を導き出し，複数機能同時動作時の性能予測に用いることは有効である．
- 基準点の CPU 使用率およびメモリ帯域使用率を計測し，各機能の純粋な CPU 使用率およびメモリ帯域使用率を算出し，複数機能同時動作時の性能予測に用いることは有効である．
- CPU にパフォーマンスカウンタが付いていない機器でも，チップセレクト信号を用いることで，メモリ帯域使用率を考慮した同時動作時の性能予測が可能である．

7.2　今後の課題

6.2 節で述べたとおり，本提案手法による性能予測値の使い方には留意が必要であり，操作性や応答性に影響がないかを判断する閾値を設けることが実際には必要であることを述べた．この閾値は，現状，人手によって設定する必要がある．経験に基づく人手の設定では，属人的になるため，自動化することが望ましいと考えており，閾値設定の自動化が今後の課題である．

デジタルテレビの例では，リモコンの操作キューの蓄積具合などから応答性や操作性を数値化し，性能予測方法に反映させることや，バッファーオーバフローやアンダフローのエラー回数などから映像破綻について数値化し，性能予測方法に反映させることが今後の課題である．

また他の方法として，経験による閾値をデータベース化し，類似機能に対する閾値をデータベースから取得するなどの，閾値取得の自動化が今後の課題になる．

参考文献

[1] Juurlink, B., Antochi, I., Crisu, D., Cotofana, S. and Vassiliadis, S.: GRAAL: A framework for low-power 3D graphics accelerators, *IEEE Comput. Graph. Appl.*, Vol.28, No.4, pp.63–73 (2008).

[2] Crisu, D., Cotofana, S. and Vassiliadis, S.: A hardware/software co-simulation environment for graphics accelerator development in ARM-based SOCs, *Proc. 13th Annual Workshop on Circuits, Systems and Signal Processing, ProRISC* (2002).

[3] 柴村英智，薄田竜太郎，本田宏明，稲富雄一，于　雲青，井上弘士，青柳　睦：PSI-NSIM：大規模並列システムの性能解析に向けた並列相互結合網シミュレータ，信学技報，CPSY2007-32 (2007).

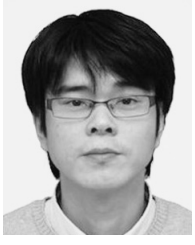
[4] Behrmann, G., David, A., Larsen, K.G., Petterson, P. and Yi, W.: Developping UPPAAL over 15 years, *Software: Practice and Experience*, Vol.41, Issue 2, pp.133–142 (2011).

[5] Kupferschmid, S., Wehrle, M., Nebel, B. and Podelski, A.: Faster Than Uppaal?, *CAV 2008*, pp.552–555 (2008).

[6] Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T. and Maler, O.: SpaceEx: Scalable Verification of Hybrid Systems, *CAV 2011*, pp.379–395 (2011).

[7] 伊藤明彦，長岡武志，岡野浩三，楠本真二：確率的モデル

検査ツールを用いた実時間ネットワークシステムの検証手法の提案およびネットワークシミュレータ NS-2 との比較，電子情報通信学会技術研究報告．SS，ソフトウェアサイエンス，Vol.109, No.170, pp.37–42 (2009).
[8] David, A., Larsen, K.G., Legay, A., Mikucionis, M. and Wang, Z.: Time for Statistical Model Checking of Real-time Systems, *Proc. 23rd International Conference on Computer Aided Verification* (*CAV'11*) (2011).
[9] Kwiatkowska, M., Norman, G. and Parker, D.: PRISM 4.0: Verification of Probabilistic Real-Time Systems, *CAV*, pp.585–591 (2011).
[10] タネンバウム，A.S.：OS の基礎と応用—設計からの実装，DOS から分散 OS Amoeda まで，ピアソン・エデュケーション，東京 (1995).
[11] Liu, C.L. and Layland, J.W.: Scheduling algorithms for multiprogramming in a hard-real-time environment, *J. Association for Computing Machinery*, Vol.20, No.1, pp.46–61 (1973).
[12] Jun, M., Bang, K., Lee, H., Chang, N. and Chung, E.: Slack-based bus arbitration scheme for soft real-time constrained embedded system, *Proc. 12th Asia and South Pacific Design Automation Conference* (*ASPDAC2007*), pp.159–164 (2007).
[13] Lahiri, K., Raghunathan, A. and Lakshminarayana, G.: LOTTERYBUS: A new high-performance communication architecture for system-on-chip designs, *Proc. 38th Design Automation Conference* (*DAC2001*), pp.15–20 (2001).
[14] Pyoun, C.H., Lin, C.H., Kim, H.S. and Chong, J.W.: The efficient bus arbitration scheme in SoC environment, *Proc. 3rd IEEE International Workshop*, pp.311–315 (June-July 2003).
[15] 林　徹生，今里賢一，井上弘士，村上和彰：演算/メモリ性能バランスを考慮した CMP 向けオンチップ・メモリ貸与法の提案，情報処理学会研究報告．組込みシステム，Vol.1, pp.59–64 (2008).
[16] 井上弘士，甲斐康司，村上和彰：DRAM/ロジック混載 LSI の高オンチップ・メモリバンド巾を活用する動的可変ラインサイズ・キャッシュ方式の提案，信学技報，pp.109–116 (1998).
[17] 金井　遵，佐々木広，近藤正章，中村　宏，天野英晴，宇佐美公良，並木美太郎：性能予測モデルの学習と実行時性能最適化機構を有する省電力化スケジューラ，情報処理学会論文誌，コンピューティングシステム，No.49, pp.20–36 (2008).



若林 昇

2001 年神戸大学大学院自然科学研究科修士課程修了．同年（株）日立製作所入社．2002 年より組込みソフトの開発効率および性能の向上研究に従事，現在に至る．

吉岡 信和　（正会員）

1998 年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（情報科学）．同年（株）東芝入社．2002 年より国立情報学研究所に勤務，現在，同研究所准教授，2007 年より総合研究大学院大学准教授を兼務，セキュリティ・プライバシー技術，ソフトウェア工学，学術クラウドの研究開発に従事．2011 年より日本ソフトウェア科学会理事．電子情報通信学会，日本ソフトウェア科学会，人工知能学会，IEEE CS 各会員．