社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2005－SLDM－122 (27)
2005／12／2

シナリオを用いたタスク及びバス転送へのサイクル割り当ての一手法

山口 聖二† 谷本 匡亮† 中田 明夫† 東野 輝夫†

† 大阪大学 大学院情報科学研究科
〒 565-0871 大阪府吹田市山田丘 1-5
E-mail: †{s-yamagt,tanimoto,nakata,higashino}@ist.osaka-u.ac.jp

あらまし バスシステムを設計する際，システム全体のスループットは仕様から導出可能であるが，バスに接続する各モジュールについてのスループットやバスサイクルの割り当ての導出は容易ではない．提案手法では，シナリオをバスシステム内で実行されるタスクとバス転送の実行系列として定義する．そして，与えられたモジュール構成，バストポロジ，及びシナリオから，各タスクの演算強度や転送データ量を考慮した，スループット/レイテンシ制約を満たす各タスク及びバス転送へのサイクル割り当てを導出する．また，提案手法を MPEG2 デコーダバスシステムに適用し，制約を満たすサイクル割り当てを導出することで，本手法の有効性を確認する．

キーワード バスシステム，実時間処理，パイプライン処理，画像音声処理，サイクル割り当て

A Cycle Budgeting Method for Tasks and Bus Transfers of Bus Systems Using Scenarios

Seiji YAMAGUCHI†, Tadaaki TANIMOTO†, Akio NAKATA†, and Teruo HIGASHINO†

† Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, JAPAN
E-mail: †{s-yamagt,tanimoto,nakata,higashino}@ist.osaka-u.ac.jp

Abstract In designing a bus system, it is important to derive a real-time constraint (the number of available cycles) for each task module of the bus system while satisfying the given end-to-end real-time constraint of the entire system such as throughput and/or latency constraints. In this paper, we define a scenario as execution sequences of tasks and bus transfers executed in a bus system. Then we propose a method to derive real-time constraints for each task and bus transfer from a given bus system configuration (including bus topology) and a scenario. In deriving the real-time constraints, we consider operation strengths of tasks, the amount of bus transfers, and bus scheduling policies (either fixed priority or round robin). We show that our method is effective for a MPEG2 decoder bus system.

Key words Bus systems, Real-time systems, Pipelined processing, Multimedia processing, Cycle budgeting

1. まえがき

製造プロセス技術の進歩により，SoC へ搭載可能な機能が飛躍的に増加しており，それ故，実装モジュール間でのオンチップ通信が増加し，オンチップ通信が SoC の性能を決定する主要因となりつつある [1]．要求仕様として与えられた実時間制約を満たすようバスシステムを設計するためには，バス上に接続された各モジュールの処理時間とモジュール間のバス転送への時間を考慮する必要がある．

並行システムをモデル化し，システムのスループット制約などから各処理の実時間制約を導出する既存研究として，文献 [2], [3] がある．文献 [2] では，シングル CPU 上でバッファを介して非同期通信を行う周期タスク群を横取り可能な固定優先度スケジューラによって実行する実時間システムを対象として，入出力間の遅延，関連する入力間の時間差，および出力レートに関する実時間制約を満たし，かつ CPU 利用率を最小化するような，各タスクの周期やデッドラインを求めるアルゴリズムを提案している．また，文献 [3] では，文献 [2] の手法を EDF スケジューラに対応可能なように拡張し，各周期タスクの実時間制約とコードサイズのトレードオフを解析することで，仕様として与えた対象システムへの実時間制約を満たしつつ，コードサイズを削減する手法が提案されている．但し，文献 [2], [3]

が対象とするモデルでは，バッファによるタスク間での非同期通信が前提であり，各通信バッファに書き込み可能なタスクが一つのみに限定されているため，バス調停を伴うバスシステムをモデル化することができない．

本論文では，一定間隔のデータ投入を受けてパイプライン的な動作を行うバスシステムを対象として，バスシステム全体が仕様として与えたレイテンシ (データの入力から出力までの時間) 及びスループット (データの投入時間) 制約を満たす各モジュールの実時間制約を自動導出する一手法を提案する．提案手法が対象とするシステムは，さまざまな機能モジュールがバスで接続されたバスシステムである．ただし，画像や音声などの入力データが一定の時間間隔で到着し，システム内のあるモジュールがそれを受け取り，何らかの加工を行い，その結果のデータを次のモジュールに渡していくなど，最終的なデータを出力するまで，パイプライン的にデータを次々と受け渡していくシステムを対象とする．また，一つのモジュールの実行終了後には必ず他のモジュールへのデータ転送があるものとする．そのとき，モジュール間が専用線で接続されていればそれを用い，さもなければ，バスを用いてデータ転送を行うと仮定する．さらに，データ転送はブロッキング転送，すなわち，転送先モジュールの処理が終了していなければ，終了までデータ送信は待たされるものとする．

そのようなバスシステムの実時間制約を考慮したパイプライン動作のシナリオ記述モデルとして，時間とマークの情報を持つことができ，データの入力，出力に対応するノードを持つ，閉路の無い有向グラフ (動作シナリオと呼ぶ) を用いる．動作シナリオを用いて，システムに入力されたデータに対して，各モジュール内で処理を加えているのか，バス転送を行っているのかを表現する．また，複数ある処理の中でいずれかを選択する場合 (choice) や複数の処理を並列に実行する場合 (par) なども動作シナリオを用いて記述する．システムに入力されたデータと，それに対する処理の進み具合は，このグラフの中に加えられるマークと，その付近のノードが持つ情報を用いて表現する．

本研究ではバス調停方式に固定優先度または時分割ラウンドロビンが指定できるものとする．その下で各バス転送について最も競合したと仮定した場合の最悪転送時間 (WCET) を見積もり，モデル上の入力から出力までの各処理系列毎に，関係する全てのバス転送 WCET の合計を求め，レイテンシ制約から見積もり引いた時間を，処理系列上の各処理にあらかじめ与えられた重み付けに従い比例配分する．

次に，求めた各モジュール動作の時間制約およびバス転送の最悪時間を動作シナリオ中の対応するノードに付加した上で，動作シナリオを等価な動作意味を持つ時間ペトリネット (Time Petri Nets:TPN) [4] に変換し，デッドロックやライブロックが存在しない (これを活性と呼ぶ) か否か，および，レイテンシ/スループット制約を満たすか否かの解析を行う．得られた TPN が活性であるか否かのチェックを TPN 解析ツールである TINA [5] を用いて解析する．また，スループット制約を満たすか否かは，データ入力に対応するトランジションから接続するプレースに，二つ以上のトークンが蓄積しないことを TINA を用いて確認する．さらに，レイテンシ制約を満たすか否かは，TINA でシミュレーションを行い，得られたトランジションの発火系列を解析することにより，バス転送時間を含む全体処理の最悪実行時間を求めることで確認する．

動作シナリオ中に同じモジュールによる動作が複数現れる場合，モジュール動作を実行するハードウェアは通常 1 つしか存在しないため，前のデータの処理を終えるまで次のデータを処理することができない．そのような場合 (一般に複数の処理間でリソース競合が存在する場合)，各処理の実行が待たされる場合が生じるため，単純にレイテンシ制約からバス転送の最悪時間の合計を引いた残り時間をモジュール動作に配分しただけではレイテンシ制約を満たせない場合がある．そこで，各処理のリソース競合が高々j 回しか生じないという前提の下で，競合によって生じる実行時間のオーバヘッドの最悪値を見積もり，レイテンシ制約からバス転送の最悪値とリソース競合のオーバヘッドの最悪値を引いた時間を各モジュールへ処理の重みに応じて比例配分する．次に，得られた各モジュールの時間制約を代入した TPN が活性かつレイテンシ/スループット制約を満たすか否かをチェックする．これを $j=1$ から順に増やしながら，活性かつレイテンシ/スループット制約を満たす j が見つかるまで繰り返す．リソース競合が高々j 回のとき制約を満たした場合，レイテンシ/スループット制約を満たすより緩い解は j と $j-1$ の場合の時間制約の中間のどこかに存在すると考えられる．そのような解を二分探索法にて探索することにより，出来るだけ緩い各モジュール動作への時間制約を導出する．

提案手法の適用実験として，共有メモリ方式の MPEG2 デコーダシステム [6]〜[9] の設計に適用し，バスシステム全体への時間制約からの個々のモジュールへの実時間制約の導出結果を示す．

2. 動作シナリオによるモデル化手法

対象とするバスシステムのモデル化は次のように行う．まず，**動作シナリオ**とは一つの入力データに対するバスを構成するモジュールの動作実行系列およびそれらの逐次実行 (seq)，選択実行 (choice)，並列実行 (par) のことであると定義する．バスシステムの処理実行順は以下の例のようにバスシステムの動作シナリオに基づいて記述する．

[例 1] 図 1(a) のバスシステム構成を考える．バスにはモジュール A，C，E のみが接続されており，B は A のみに，D は C のみに接続されている．このとき，以下のような動作シナリオが考えられる (図 1(b)，(c) 参照)．

シナリオ 1： A が E からデータをフェッチし加工後，B か C へ転送．C に転送された場合，C がデータを加工後，D に転送．

シナリオ 2： C が E からデータをフェッチし加工後，A および D へ転送．A がデータを加工後，B へ転送． □

各動作シナリオについて，以下の通りグラフを作成することで，バスの動作を記述する．バスシステムのモジュール名はノードのラベルで記述する．モジュール間のデータ転送がバス転送である場合，該当するアークに明示する．また，データ投入間隔はグラフの根に当たるノードへアークが入力されるよう

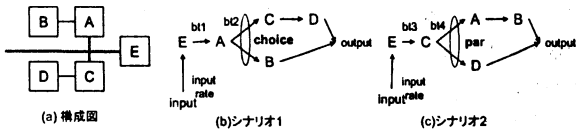

図 1　例 1 のバスシステム構成図および動作シナリオ
Fig. 1　Bus System Configuration and Scenarios of ex.1

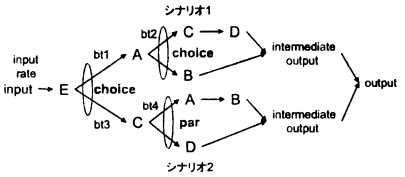

図 2　例 2 の動作シナリオ
Fig. 2　Scenario of ex.2

に，ダミーのノード，アークを追加し，このアークへ時間を明示する．そして，分岐が存在する場合は実行方式を choice，par で指定する．

[例 2]　図 2 は例 1 にある二つの動作シナリオからいずれかが選択実行されるという動作シナリオを記述したものである．□

バスシステムで起こり得る競合としては，1) バス競合，2) リソース競合，の二つが考えられるため，これらをモデル化する必要がある．

各バス転送に対しては，1. で既に述べたとおり固定優先度または時分割ラウンドロビンのバス調停方針を定め，その下でバス転送にかかる最悪時間を見積もり，バス転送動作を表すアークの時間制約とする．

一方，異なる動作シナリオを表す二つ以上の部分グラフに配置された同一モジュールを表すノードに対して，リソース競合の指定を行う．すなわち，同じラベルが記されたシナリオ上の複数ノードのうち，高々一つのノードのみが入力データを受けつけ，処理を行うものとする．

3.　実時間制約割り当て問題

本論文では，実時間制約割り当て問題を以下のように定義する．

[定義 1]　バスシステムへの時間制約割り当て問題とは，(1) バスシステムの動作を記述した動作シナリオ，(2) バス競合が一切発生しないと仮定した場合の各モジュールからのバス転送時間，(3) 動作シナリオ上の各モジュールに与えられた処理の重み，(4) バスシステム全体のスループット/レイテンシ制約，および (5) バス調停方式 (固定優先度方式，または時分割ラウンドロビン方式) を入力とし，スループット/レイテンシ制約を満たすための，バスシステムを構成する各モジュールに対する時間制約を導出する問題である．□

ただし，提案手法で時間制約が導出不能となるのは，(1) 入力モデルの動作意味を表す動作シナリオが活性でない場合，(2) 調停方式を考慮したバス転送の最悪実行時間の見積もり値がレイテンシ制約以上である場合，および (3) 指定したリソース競合に対してリソース競合解消にかかる最悪実行時間の見積もり値がレイテンシ制約以上である場合，の何れかの場合である．また，時間制約の導出では出来うる限り最大の時間制約の導出を行う．

4.　時間制約見積もり手法

提案手法では，先ず，バス転送の最悪ケースの実行時間解析を動作シナリオ上のパス毎に行い，レイテンシ制約からバス転送の最悪実行時間を引いた値を動作シナリオ上のパス毎の新たなレイテンシ制約として，リソース競合を考慮した各機能モジュールの時間制約の導出を行う．

以降の説明では，動作シナリオ上の入力から出力までの各有向パスを Path(i)(i は添え字)，モジュール動作ノードの集合を P，Path(i) 上のモジュール動作ノードの集合を P(Path(i)) とする．

4.1　バス転送の最悪実行時間解析

提案手法では固定優先度方式，時分割ラウンドロビン方式に対して下記を行う．

(1)　動作シナリオ上にバス転送が指定された n 個のデータ転送 $\{t_1, t_2, \ldots, t_n\}$ が存在し，バス競合がない場合の各バス転送の転送時間を $\{x_1, x_2, \ldots, x_n\}$ とする．

固定優先度方式の場合　データ転送 t_i の優先度を $\sigma(i)$ $(1 \leq \sigma(i) \leq n, \sigma : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$ は置換) とすると，t_i のバス転送の最悪実行時間 x'_i は，x_i に優先度 $\sigma(i)$ 未満のバス転送時間の総和を加えることから，$x_i + \sum_{\sigma(j)<\sigma(i)} x_j$ となる．

時分割ラウンドロビン方式の場合　異なる n 個のモジュールからバスアクセスが存在するとすると，バス転送 t_i でのバス転送の最悪実行時間 x'_i は，$x_i \times n$ となる．

t_i の実行時間を x'_i に修正する．これを動作シナリオ上で指定された全てのバス転送に対して行う．

(2)　動作シナリオ上の全てのパス毎に，パス上に存在する全てのバス転送に渡って，(1) で求めたバス転送時間の合計を求め，レイテンシ制約時間から引いたものを，各パス $Path(i)$ に対するレイテンシ制約時間 $Lnew_i$ とする．

4.2　リソース競合を考慮した時間制約導出

前節で求めた動作シナリオ上の各パスへの新たなレイテンシ制約 $\overrightarrow{Lnew} = (Lnew_1, \ldots, Lnew_n)$ に対して，リソース競合を考慮したパス上の各バス転送への時間制約導出手法の手順概要を下記に示す．

Step1　(リソース競合が全くない場合の時間制約導出)

動作シナリオ上の各パス $Path(i)(i \in \{1, \ldots, n\})$ 上のバス転送に対して，モジュール動作の重みによるレイテンシ制約 $Lnew_i$ の加重平均を求め，それをモジュール動作の時間制約とする．また，任意の $i \in \{1, \ldots, N\}$ に対して $Lw_i(0) = Lnew_i$ とする．

レイテンシ制約を $Lnew_i$ としたとき，モジュール動作ノード $p \in P(Path(i))$ に対する，$Path(i)$ における時間制約 $Val_i(Lnew_i, p)$ は下記で与えられる．

$$Val_i(Lnew_i, p) \stackrel{\text{def}}{=} Lnew_i \times Weight(p)/WS(i).$$

ここで，$Weight(p)$ はモジュール動作 $p \in P$ に関連付けられた処理の重み，$WS(i)$ はパス $Path(i)$ 上のモジュール動作の重みの総和，すなわち，$WS(i) \stackrel{\text{def}}{=} \sum_{p' \in P(Path(i))} Weight(p')$ とする．

さらに，レイテンシ制約 $\overrightarrow{Lnew} = (Lnew_1, \ldots, Lnew_n)$ が与えられたときのモジュール動作 p の時間制約 $Val(\overrightarrow{Lnew}, p)$ は下記で与えられる．

$Val(\overrightarrow{Lnew}, p) \stackrel{\text{def}}{=} \min_{i \in \{1,\ldots,n\}} \{Val_i(Lnew_i, p)\}$.

Step2 (リソース競合が高々j回存在する場合の時間制約導出)

まず，$j := 1$ とし，パス $Path(i)$ 上で同じラベルを持つモジュール動作ノードが必ず競合を起こし，他のパスのモジュール動作が全て j 回ずつリソースを取得して開放するまでの時間 $TConf(i,j)$ を Step1 で求めたバス転送の値を用いて算出する．

$TConf(i,j) \stackrel{\text{def}}{=} \sum_{p \in P(Path(i))} \sum_{k \in \{1,\ldots,n\} \setminus \{i\}} \sum_{p' \in P(Path(k)) \cap ConfResource(p)} (Val_i(Lw_i(j-1), p') \times j)$.

ただし，$ConfResource(p)$ はモジュール動作ノード p と同じラベルを持つモジュール動作ノードの集合 (存在しなければ空集合) を返す関数とする．

次に，パス $Path(i)$ のレイテンシ制約 $Lw_i(j-1)$ から $TConf(i,j)$ を差し引いた値を新たなレイテンシ制約 $Lw_i(j)$ として，$Lw_i(j)$ に対して Step1 を実施し，各バス転送の時間制約を導出する．

$Lw_i(j) \stackrel{\text{def}}{=} Lw_i(j-1) - TConf(i,j)$.

Step3 (Step2 で算出した値の正当性チェック (5. 参照))

動作シナリオに対して，$\{Lw_i(j)\}_{i \in \{1,\ldots,n\}}$ で算出したモジュール動作の時間制約を与えた動作シナリオの動きを表す TPN を，TINA を用いて，活性，スループット制約，初期レイテンシ制約 T を満たすか否かの判定を行い，全て満たされた場合，Step4 へ，そうでない場合，j に 1 を加え Step2 へと処理を移行する．

Step4 (線形計画法を用いたリソース競合が高々j回存在する場合の時間制約導出)

$Lw_i(j)$ を各モジュールに比例配分した時間制約 $\{Val_i(Lw_i(j), p)\}_{p \in P(Path(i))}$ は，実時間制約割り当て問題の一つの解になっているが，一般に厳しすぎる可能性があるため，より緩い解の候補を以下のヒューリスティックで求める．直観的には各パス $Path(i)$ 上のモジュールの時間制約 $\{X_{p,i}\}_{p \in P(Path(i))}$ の総和 Lid_i を，全体の処理時間が $Lw_i(j-1)$ からリソース競合のオーバヘッドを引いたものと一致するまで，各 $X_{p,i}$ を伸ばした時間制約が求めるべき解の候補になると考えられる．そこで，各パスに対して，Lid_i が $Lw_i(j-1)$ 以上 $Lw_i(j)$ 以下となる不等式，Lid_i が，$Lw_i(j-1)$ からリソース競合が j 回生じるとした場合の各モジュール処理のオーバヘッドを引いた値以下となる不等式，及び，$Path(i)$ 上の各バス転送の時間変数が，$Lw_i(j)$ で算出した値以上 $Lw_i(j-1)$ で算出した値以下とする不等式，同じ処理の異なるパス上で見積もった時間制約の値は等しいとする等式を構成し，Lid_i の総和が最大となるよう線形計画法で解を求め，得られた $\overrightarrow{Lid} = (Lid_1, \ldots, Lid_n)$ に対して Step1 を実施し，各モジュール動作の時間制約 $\{Val(\overrightarrow{Lid}, p)\}_{p \in P}$ を導出する．なお，線形計画法への入力となる式は下記となる．

変数 (実数): $\{Lid_i\}_{i \in \{1,\ldots,n\}}$, $\{X_{p,i}\}_{p \in P, i \in \{1,\ldots,n\}}$

目的関数: $\sum_{i \in \{1,\ldots,n\}} Lid_i$ を最大化

線形制約式:

$\forall i \in \{1,\ldots,n\}\ [Lid_i = \sum_{p \in P(Path(i))} X_{p,i}]$,

$\forall i \in \{1,\ldots,n\}\ [Lw_i(j) \leqq Lid_i \leqq Lw_i(j-1)]$,

$\forall i \in \{1,\ldots,n\}\ [Lid_i \leqq Lw_i(j-1) - (\sum_{p \in P(Path(i))} \sum_{k \in \{1,\ldots,n\} \setminus \{i\}} \sum_{p' \in P(Path(k)) \cap ConfResource(p)} X_{p,k})]$,

$\forall i \in \{1,\ldots,n\}\ \forall p \in P(Path(i))\ [Val_i(Lw_i(j-1), p) \leqq X_{p,i} \leqq Val_i(Lw_i(j), p)]$

$\forall p \in P\ [X_{p,1} = X_{p,2} = \cdots = X_{p,n}]$

Step5 (3 点の二次曲線補間)

Step3 で求めた解 $\{Val(\overrightarrow{Lw_i(j)}, p)\}_{p \in P}$ よりもよい解は，直観的には $\{Val(\overrightarrow{Lw_i(j)}, p)\}_{p \in P}$, $\{Val(\overrightarrow{Lid}, p)\}_{p \in P}$ $\{Val(\overrightarrow{Lw_i(j-1)}, p)\}_{p \in P}$ の三点の間のどこかにあると考えられる．そこで，以下のヒューリスティックにより解候補の集合を求める．

パス $Path(i)$ で求めた $Lw_i(j-1)$, Lid_i, $Lw_i(j)$ の三点を二次元座標で，$(1, Lw_i(j-1))$, $(2, Lid_i)$, $(3, Lw_i(j))$ としてプロットし，$(1, Lw_i(j-1))$, $(2, Lid_i)$ 間，$(2, Lid_i)$, $(3, Lw_i(j))$ 間を K 点でそれぞれ補間する (K は任意に定める)．

Step6 (二分探索法による解探索)

Step5 で求めた解候補の集合から最も良い解を効率よく探索するために，Lid からスタートして，レイテンシ制約を変更しながら Step3 を実施し，活性でかつスループット及びレイテンシが満たされる範囲で，初期に与えたレイテンシ制約 L を満たす最大解を，二分探索法に従って，補間した $2 \times K$ 点から探索する．

5. TINA による解析

本章では，動作シナリオの動作意味を表す TPN，及び，それに 4. で求めた時間制約を代入したモデルが活性であるかの解析，及びスループット及びレイテンシ制約の TINA による検査手法について述べる．

5.1 活性及びスループットの解析

動作シナリオ A に対して，4. で求めたバス転送見積もりで算出した時間制約，及びその他のデータ転送への時間制約を付加した動作シナリオ B を構成する．B がスループット制約を満たすか否かは確認しない限り定かでないため，動作シナリオを等価な動作を行う TPN に変換し，TPN 上で，まず，デッドロック/ライブロックが起こらないこと (活性であること) を TINA で検証する．この検証で活性であることが確認された場合，次に，入力データがノンブロッキングで入力されるとした意味のモデルに変形し，その下で入力データが入力バッファを表すプレースに蓄積しないか否かを TINA を用いて検証する．

5.2 レイテンシの解析

与えられた動作シナリオの動作意味を表す TPN に対して，バス転送見積もりで算出した時間制約，及び前節で求めた時間制約を対応するトランジションに付加し，TINA を用いて網羅的にシミュレーションを実施し発火系列を解析することで，

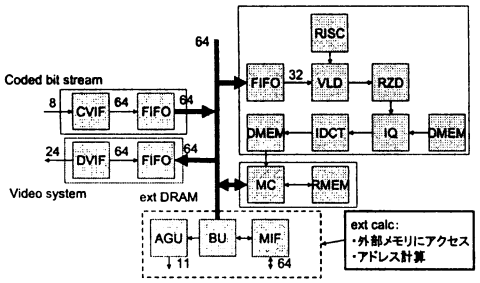

図 3 MPEG2 デコーダ LSI のブロック図

Fig. 3 Block Diagram of MPEG Decoder LSI

ソーストランジション発火からシンクトランジション発火に至る発火系列の中で最大の時間経過を求め，それがレイテンシを満たしているかをチェックすることにより行う．

6. 実 験 例

提案手法を図 3 に示す MPEG デコーダのバスシステム [6] に適用した結果を示す．

6.1 MPEG デコーダ 処理概要

復号化処理は符号化データが CVIF から入力され，入力されたデータは外部メモリに一度蓄積された (ext DRAM) 後，VLD に読み込まれる．VLD では入力データの種類により，分岐が発生する．入力データのみで復号化を行い，出力画像を生成できる I フレームと前方向または双方向のフレームを参照し差分を用いて画像を生成する P，B フレームとなる．I フレームでは入力画像に対して，可変長符号化 (VLD)，逆量子化 (IQ)，逆 DCT 変換 (IDCT)，フレーム内動き補償 (MC) を施し，画像を復号する．P,B フレームでは入力データから動きベクトルなどのパラメータを算出し (VLD,RISC)，参照画像を取得後 (RMEM)，動き補償 (MC) を行い画像を復号する．復号された画像は一度外部メモリに蓄積された (ext DRAM) 後，バッファに書き込まれ (DispFIFO)，フォーマット変換を施した (DVIF) 後，出力される．

6.2 動作シナリオによるモデル化

前節で説明した図 3 のモジュールレベルにおける動作シナリオを図 4 に示す．図 4 で，t_k^i は $Path(i)$ 上で k 番目のモジュール動作への時間制約とする．また，図 4 に示すように，各バス転送に対して $bt0 < bt1 < \cdots < bt8$ となるように固定優先度を付加する．

各モジュール動作の重みは，文献 [7] に記載されている MPEG デコーダの各処理で行われる演算量 (MOPS:Million Operation Per Seconds) の概算値を実装での並列度で割った値を用いる．なお，実装での並列度は文献 [8], [9] での実装方式を参考に，MC 及び IDCT の処理能力を 1pixel/cycle とし，実装並列度をそれぞれ 128，12 とした．特に，IDCT は Low と Column 処理それぞれを並行して処理可能な構成としてモデル化を行った．各モジュール動作の重みを表 1 に示す．

バス競合が一切ないという前提における各バス転送のデータ転送時間と，各バス転送の最悪実行時間を表 2 に示す．

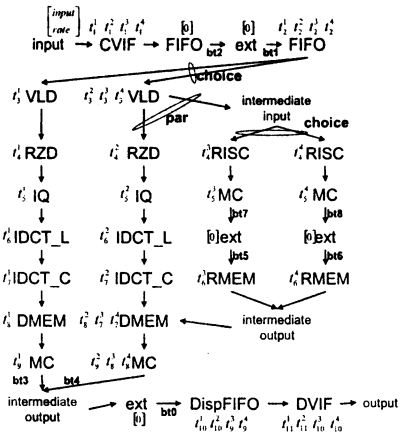

図 4 MPEG2 デコーダの動作シナリオ

Fig. 4 Scenario of MPEG2 Decoder

バスシステム全体として行うパイプライン動作への制約であるレイテンシとスループットに関しては，[8] での実装方式を参考に，パイプラインの並列度を 4 と仮定し，スループット制約を 1/30[sec.]，レイテンシ制約を 4/30[sec.] としている．

6.3 実 験 結 果

動作シナリオの各パスにおける $Lw(0)$，Lid，$Lw(1)$ の値を表 3 に示す．

リソース競合が多いにも関わらず，$j = 1$ で活性かつスループット/レイテンシ制約が満たされた．なお，線形計画問題ソルバ LP-Solve に与えた Lid の線形計画式を図 5 に，$Lw(0)$，Lid，$Lw(1)$ を補間したグラフを図 6 に示す．二分探索法によ

表 1 各モジュール動作の重み

Table 1 Weights for Each Module Operation

	Path A	Path B		Path C	Path D
CVIF	1.944	1.944	CVIF	1.944	1.944
FIFO	1.944	1.944	FIFO	1.944	1.944
VLD	1.2	1.2	VLD	1.2	1.2
RZD	2.4	2.4	RISC	1.2	2.4
IQ	4	4	MC	0.030375	0.06075
IDCT_L	7.7916	7.7916	RMEM	1.944	3.888
IDCT_C	7.7916	7.7916	DMEM	1.944	1.944
DMEM	1.944	1.944	MC	1.09375	1.09375
MC	0.030375	1.09375	DispFIFO	1.944	1.944
DispFIFO	1.944	1.944	DVIF	10	10
DVIF	10	10			
計	40.989575	42.05295	計	23.244125	26.4185

表 2 バス転送最悪実行時間

Table 2 Worst Case Execution Time of Each Bus Data Transfer

バス転送 (優先度順)	転送時間 [msec.]	WCET [msec.]	バス転送 (優先度順)	転送時間 [msec.]	WCET [msec.]
bt0	1.08	1.08	bt5	1.08	6.48
bt1	1.08	2.16	bt6	2.16	8.64
bt2	1.08	3.24	bt7	0.030375	8.670375
bt3	1.08	4.32	bt8	0.06075	8.731125
bt4	1.08	5.4	-	-	-

表 3 各パスにおける $Lw(0), Lid, Lw(1)$ の値

Table 3 $Lw(0), Lid, and Lw(1)$ for Each Path

パス	Lw(0)[msec.]	Lid[msec.]	Lw(1)[msec.]
Path A	122.533	78.4366	49.8306
Path B	121.453	92.1496	49.3955
Path C	106.311	74.6023	73.344
Path D	104.128	84.0779	83.6145

目的関数 $\max\{\sum_{1\le i\le 4} Lid_i\}$

$Lid_k = \sum_{1\le i\le 11} t_i^k \quad k=1,2, \quad Lid_k = \sum_{1\le i\le 10} t_i^k \quad k=3,4$

$Lid_1 + t_4^1 + t_5^1 + t_6^1 + t_7^1 + t_8^1 + t_9^1 + t_5^3 + t_5^4 \le Lw_1(0)$

$Lid_2 + t_4^1 + t_5^1 + t_6^1 + t_7^1 + t_8^1 + t_9^1 + t_5^3 + t_5^4 \le Lw_2(0)$

$Lid_3 + t_4^4 + t_5^4 + t_6^4 + t_7^4 + t_8^4 + t_9^4 + t_5^3 + t_5^4 \le Lw_3(0)$

$Lid_4 + t_4^3 + t_5^3 + t_6^4 + t_7^4 + t_8^4 + t_9^4 + t_5^3 + t_5^4 \le Lw_4(0)$

$Lw_i(1) \le Lid_i \le Lw_i(0) \quad 1 \le i \le 4$

$t_1^1 = t_1^2 = t_1^3 = t_1^4, \quad t_2^1 = t_2^2 = t_2^3 = t_2^4, \quad t_3^2 = t_3^1 = t_3^4, \quad t_8^2 = t_7^3 = t_7^4,$

$t_9^2 = t_8^3 = t_8^4, \quad t_{10}^1 = t_{10}^2 = t_9^3 = t_9^4, \quad t_{11}^1 = t_{11}^2 = t_{10}^3 = t_{10}^4$

$[Val(Lw(1), x(i-1)) < t_i^1 < Val(Lw(0), x(i-1))] \quad 1 \le i \le 11$

$[Val(Lw(1), x(i+8)) < t_i^2 < Val(Lw(0), x(i+8))] \quad 3 \le i \le 9$

$[Val(Lw(1), x(i+14)) < t_i^3 < Val(Lw(0), x(i+14))] \quad 4 \le i \le 6$

$[Val(Lw(1), x(i+17)) < t_i^4 < Val(Lw(0), x(i+17))] \quad 4 \le i \le 6$

図 5 *Lid* を得る線形計画式

Fig. 5 ILP Equations to Calucurate *Lid*

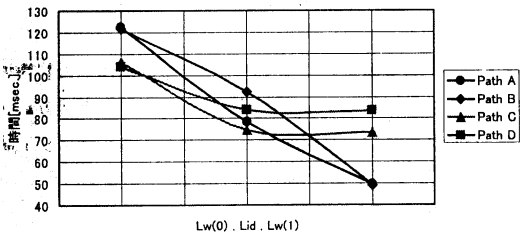

図 6 Lw(0),Lid,Lw(1) による補間

Fig. 6 Interpolation Using Lw(0), Lid, and Lw(1)

表 4 時間制約解探索結果

Table 4 Search Result for Real Time Budget

分割 index	Sim. 実行順	throughput const. (≦ 1/30?)	latency [msec.]	latency const. (≦ 4/30?)
0(Lw(0))	2	NG	-	-
512	3	NG	-	-
768	4	NG	-	-
896	5	NG	-	-
928	7	OK	184.11	NG
944	8	OK	137.68	NG
952	9	OK	142.02	NG
954	11	OK	147.89	NG
955	12	OK	130.80	OK
956	10	OK	128.11	OK
960	6	OK	129.04	OK
1024(Lid)	1	OK	108.32	OK

る解探索における分割数 K を本実験では 1024 とした．二分探索法による時間制約解探索結果を表 4 に示す．

MPEG デコーダへの入力フレームが I,B,P の三種類であり，1[sec.]に 30 フレームの処理が行われることから，3^3[sec.] 以上のフレーム数を評価する必要があると考え，10,000 フレーム分の PN シミュレーションを TINA を用いて行った．

シミュレーション結果を表 4 に示す．二分探索の結果，$Lw(0)$ と *Lid* を補間し，1024 分割した中で 955 番目の結果が 130.80[msec.] となり，レイテンシ/スループット制約を満たす最も緩い解を与える時間制約となった．

レイテンシ/スループット制約を満たす各モジュールの最も緩い時間制約を表 5 に示す．各動作シナリオ上で競合を指定されたモジュールが高々1 回競合する中での最悪実行時間よりは良い結果が得られた．

表 5 導出された各モジュール動作への時間制約

Table 5 Derived Real Time Budget for Each Module Operation

	Path A [nsec.]	Path B [nsec.]		Path C [nsec.]	Path D [nsec.]
CVIF	3837813	3837813	CVIF	3837813	3837813
FIFO	3837813	3837813	FIFO	3837813	3837813
VLD	2369020	2697926	VLD	2697926	2697926
RZD	4738040	5395853	RISC	4177634	8589143
IQ	7896734	8993089	MC	105746	217412
IDCT_L	15382181	17517788	RMEM	6767767	13914411
IDCT_C	15382181	17517788	DMEM	4370641	4370641
DMEM	3837813	4370641	MC	2459047	2459047
MC	59965	2459047	DispFIFO	3837813	3837813
DispFIFO	3837813	3837813	DVIF	19741836	19741836
DVIF	19741836	19741836			

7. あとがき

本論文では，バスシステムのタスクとバス転送の実行系列を記述するための記法として動作シナリオを定義し、データ投入間隔が一定で全体としてパイプライン処理を行うバスシステムを動作シナリオを用いてモデル化し、バスデータ転送量とバス上の各モジュールの負荷を重みとして与え，バス調停方式の情報を与えることで，スループット/レイテンシ制約を満たす各モジュールでの実時間制約の導出を行う手法を提案した．また，提案手法を MPEG デコーダバスシステムに適用することで，提案手法の有効性を確認した．

今後の課題としては，他のパイプライン・バスシステムへの適用や，バスシステム内で実際に行われているバス転送を解析することで最悪見積もり値の緩和を行い，モジュールへの実時間制約を緩和する手法の検討などが挙げられる．

文　献

[1] D. Sylvester and K. Kutzer, "Getting to the bottom of deep sub-micron", In Proc. of ICCAD'98, 1998.

[2] R. Gerber, H. Seongsoo, M. Saksena, "Guaranteeing Real-Time Requirements with Resource-Based Calibration of Periodic Processes" IEEE Trans. on Software Engineering, Vol. 21, No. 7, 1995.

[3] I. Shin, I. Lee, and S. L. Min, "Embedded System Design Framework for Minimizing Code Size and Guaranteeing Real-Time Requirements", Proc. of the 23rd IEEE Real-Time Systems Symposium (RTSS 2002), pp.201-211, 2002.

[4] 村田忠夫,「ペトリネットの解析と応用」, 近代科学社

[5] B. Berthomieu, P.-O. Ribet, F. Vernadat, "The tool TINA – Construction of Abstract State Spaces for Petri Nets and Time Petri Nets", International Journal of Production Research, Vol. 42, No 14, 2004.

[6] T. Demura et. al., "A Single-Chip MPEG2 Video Decoder LSI", Proc. of ISSCC'94, 1994.

[7] 映像情報メディア学会,「総合情報マルチメディア選書 MPEG」オーム社,1996.

[8] T. Onoye, T. Masaki, Y. Morimoto, Y. Sato, I. Shirakawa, and K. Matsumura, "Single Chip Implementation of MPEG2 Decoder for HDTV Level Pictures", IEICE Trans. Fundamentals, Vol. E79-A, No. 3, 1996.

[9] T. Onoye, G. Fujita, M. Takatsu, I. Shirakawa, and N. Yamai, "Single Chip Implementation of MPEG2 Motion Estimator Dedicated to MPEG2 MP@HL", IEICE Trans. Fundamentals, Vol. E79-A, No. 8, 1996.