Vol.2011-ARC-196 No.17
2011/7/28

発行キューのタグRAMのバンク化と正確なクリティカルパスの遅延時間評価

山口恭平[†1]　甲良祐也[†1,*1]　安藤秀樹[†1]

本論文では，以下の 2 つの点に注目し，発行キューの遅延を回路シミュレータ SPICE を用いて評価した．1 つ目は，発行キューの遅延を短縮するために，発行キューを構成する部品の 1 つであるタグ RAM のバンク化を行った．タグ RAM は，通常の RAM と異なりアドレス・デコーダを持っていない構成であるため，バンク化には特別な設計を要する．2 つ目は，発行キューの正しいクリティカル・パスを見つけることである．従来の研究では，発行キューを構成する各部品のクリティカル・パスの遅延を求め，それらを単純に足し合せることで発行キュー全体の遅延としいたが，それでは発行キューの正しい遅延時間を得られない．なぜなら，発行キューを構成する各部品のクリティカル・パスは論理的にはつながっていないからである．32nm の LSI 技術を仮定し，8 から 128 エントリの発行キューの遅延を測定した結果，タグ RAM のバンク化と正しいクリティカル・パスを求めることによって，これらを行わない場合に比べて，最大 20%短い遅延時間を得た．

Banking Tag RAM of Issue Queue and Evaluation of Correct Critical Path Delay

Kyohei Yamaguchi,[†1]　Yuya Kora[†1,*1]
and　Hideki Ando[†1]

This paper evaluated the issue queue delay, using the circuit simulator, SPICE, focusing on following two features. First, we introduce banking the tag RAM, which is one of the components comprising the issue queue, to reduce the delay. Unlike normal RAM, banking the tag RAM requires a special design, because it does not have an address decoder. Second, we explore and identify a correct critical path in the issue queue. Previous studies summed the critical path of each component in the issue queue to obtain the delay of the issue queue, but this does not provide the correct delay of the issue queue, because the critical path of each component are not connected logically. We evaluated the delay of an issue queue with eight to 128 entries, assuming 32nm LSI technology, and found that banking the tag RAM and identifying the correct critical path reduce the delay by up to 20%, compared without these optimizations.

1. はじめに

命令の動的なスケジューリングを行う発行キューは，より多くの命令レベル並列を利用するために，プロセッサの世代交代とともに拡大されている[2),3)]．一方で，発行キューはプロセッサのクリティカル・パスの 1 つであり，その遅延時間はクロック・サイクル時間を制限する．よって，発行キュー拡大は，IPC の向上とクロック・サイクル時間の増加というトレードオフの下に検討されなければならない．本論文は，このトレードオフ検討に有用な様々なサイズの発行キューの遅延を回路シミュレーションにより求める．

本論文では，発行キューの遅延を求めるにあたり，以下の点を特に検討した．

- 発行キューの構成要素の 1 つであるタグ RAM の遅延を短縮するため，このバンク化を検討した．タグ RAM はアドレス・デコーダがなく，ワード線には，同じく発行キューの 1 構成要素であるセレクト論理からの発行許可信号が直接つながれている．通常の RAM のバンク化においては，アドレスによりバンク選択が行われるが，タグ RAM はそのような方法をとれないため，バンク化にはタグ RAM に特化した方法が必要である．
- 発行キューの遅延を求めたこれまでの研究[1),12)] では，発行キューを構成している部品それぞれのクリティカル・パスの遅延を合計することにより，発行キューの遅延としていた．しかしこの方法は正しくない．なぜなら，それぞれの部品のクリティカル・パスは論理的にはつながっていないからである．本論文では，発行キューの正しいクリティカル z・パスを網羅的なシミュレーションを通して特定し，その遅延を示す．

本論文の残りの部分は，次のような構成となっている．2 節では関連研究について述べる．3 節では発行キューの回路の構成を説明し，4 節ではクリティカル・パスについて考察する．5 節で評価結果を示し，6 節で本論文をまとめる．

†1 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University
∗1 現在，ローム (株)
Presently with Rohm Co., Ltd.

Vol.2011-ARC-196 No.17
2011/7/28

2. 関連研究

Palacharla らは 800〜180nm の LSI 技術において，プロセッサ内の種々の重要な資源の遅延を評価し[7),8)]，その結果，発行キューがクロック・サイクル時間に影響を与える資源の 1 つであることを指摘した．この研究では，ウェイクアップ論理の構成として CAM を仮定し，その下で，ディープ・サブミクロンにおける LSI 技術では，配線遅延がスケールしないため，タグ・ドライブの遅延が最も深刻となることを指摘した．

彼らはまた，調停回路を使ったセレクト論理の遅延も評価した．一般に，多くの要求を同時に調停するには非常に長い時間を要する．そこで彼らは，この遅延を短縮するために，4 つの要求のうち最も優先度の高い要求に許可を出す小さな調停器を直列に接続する実装を示した．しかし，この方法だけでは単一の許可を出すようにしかできず，複数の許可を出すには，1 つの許可を出す回路を直列に接続しなければならない．これにより，許可する数に比例して遅延が増加することになり，近年の複数の機能ユニット間で共有される発行キュー[2),3)] には適していない．

Palacharla らはウェイクアップ論理とセレクト論理の遅延を評価したが，タグ RAM の遅延は評価していない．そのため発行キュー全体の遅延は得られていない．

五島らはウェイクアップ論理に CAM ではなく RAM を使った構成を提案した[1)]．RAM 構成では，CAM 機構と異なり比較器が必要ない．加えて，RAM は CAM より小さいため，論理を横断する配線の長さを短くできる．これらによって遅延を削減することができる．しかし一方で，発行キューのコンパクションが難しいという欠点を持っている．これは，RAM 構成では，発行キューの中のエントリの位置によって依存関係が表されるからである．

また彼らは，RAM 構成と CAM 構成の場合の遅延を比較したが，CAM 構成における遅延は，各部品のクリティカル・パスの遅延を単純に合計して求められている．4 節で述べるが，各部品のクリティカル・パスは論理的にはつながっておらず，この計算方法では発行キューの正しい遅延は得られない．

タグ RAM に関しては，五島らはモノリシックな RAM を仮定しており，バンク化については考えていない (ただし，彼らは最大 32 エントリまでの発行キューしか評価されておらず，この小さなサイズでは，後に示すようにバンク化は有効でない．また，各部品のクリティカル・パスの遅延を単純に合計して得られる遅延と，正しいクリティカル・パスの遅延の差も，5.4 節で示すように，大差ない．したがって，彼らの仮定と実験の範囲においては，彼らの評価結果は正しいといえる)．

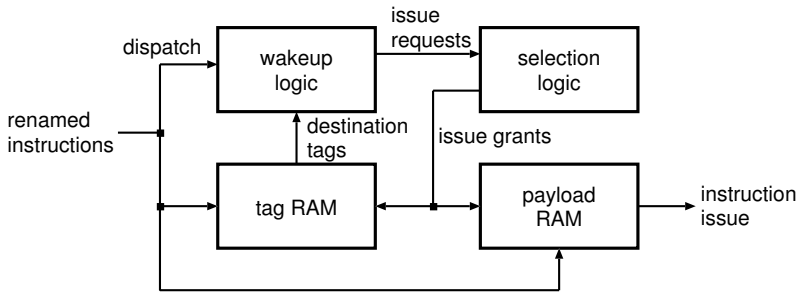


図 1　発行キューの構成
Fig. 1　Organization of issue queue.

セレクト論理に関して，五島はプレフィクス・サム論理を使った回路を提案した[11)]．この論理では，自分の要求より優先度の高い要求の数を数える．その数が発行幅より小さければ，要求は許可される．この論理回路は Palacharla らが提示した調停回路と異なり，許可される要求の数に依存して遅延が大きく増加することはない．この回路の欠点は，加算器の遅延が大きいことであるが，五島は入出力のエンコードを工夫し，これを改善した．

甲良らは，大きなサイズの発行キューの遅延を測定した[12)]．彼らは，遅延時間を大きく支配する配線遅延を短縮するために，最適にリピータを挿入した．その結果，発行キューの遅延は大きなサイズではエントリ数の 2 乗で増加すると考えられていたが，ほぼ 1 乗で増加するにすぎないことを示した．

3. 発行キューの構成と回路

発行キューはリネームされた命令を保持し，発行する命令を決定する部品である．その構成は，**図 1** に示すように，ウェイクアップ論理，セレクト論理，タグ RAM，ペイロード RAM からなる．一般に，ウェイクアップ論理は 1 次元の配列であり，各エントリは 2 つのソース・オペランドのタグと，対応する命令のデータ依存の状態（依存が解決されているかどうか）を示すレディ・フラグを持つ．もし 2 つのソース・オペランドのデータ依存が解決されているなら，発行要求をセレクト論理に送る．セレクト論理は，資源制約を考慮していくつかの要求に対して発行許可の信号を出力する．許可信号はペイロード RAM に送られ，発行する命令に関する情報が出力される．許可信号は，タグ RAM にも送られ，デスティネーション・タグが読み出される．それらのタグはウェイクアップ論理に放送され，レ

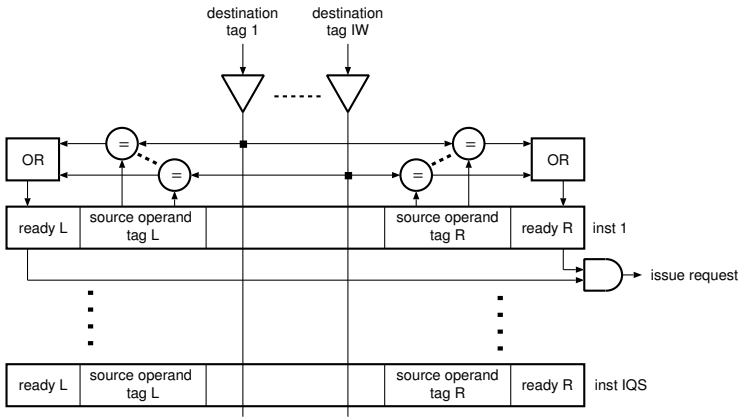


図 2 ウェイクアップ論理
Fig. 2 Wakeup logic.

ディ・フラグを更新する．

発行キューのクリティカル・パスはウェイクアップ論理→セレクト論理→タグ RAM と行き，ウェイクアップ論理に戻ってくるパスである．本論文はこのクリティカル・パスの遅延を評価する．

2 節で述べたように，ウェイクアップ論理とセレクト論理には様々な回路が提案されている．本研究ではウェイクアップ論理として，CAM を用いた回路を仮定する．RAM 構成を仮定しなかった理由は，我々の知る限り，RAM 構成では，発行キューのコンパクションが難しいという欠点がある点である (この欠点を避けるための簡単な方法は，循環バッファにより発行キューを実装することであるが，発行要求の誤った優先度をセレクト論理に与えてしまう．より精巧な方法として，セレクト論理でエイジ・マトリクス[9] を使うことがあげられるが，この回路は，複数の許可を出せるように拡張することが難しい)．セレクト論理としては，プレフィクス・サムで構成された回路を仮定する．これは，調停回路に比べて発行許可の数の増加による遅延の増加が少ないからである．

3.1 ウェイクアップ論理

図 2 に，CAM で構成されたウェイクアップ論理を示す．タグ RAM から読み出された IW 個のデスティネーション・タグが，ウェイクアップ論理の IQS 個の全てのエントリに放送される．ここで，IW と IQS は，それぞれ，発行幅と発行キュー・サイズを示す．そ

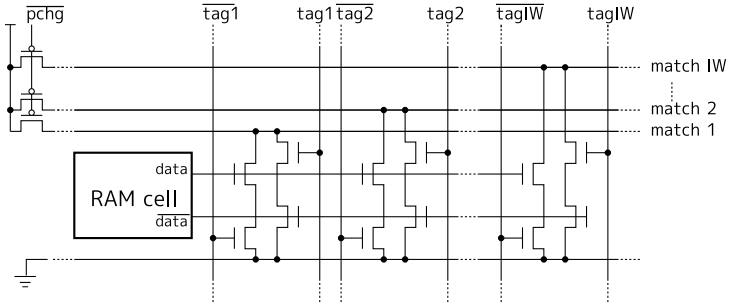


図 3 タグ比較のための CAM セルの回路
Fig. 3 CAM cell circuit for tag comparison.

れぞれのエントリは 2 つのソース・オペランド・タグを持っており，それらは放送されたデスティネーション・タグと比較される．もしタグが一致したら，レディ・フラグがセットされる．レディ・フラグが両方セットされたら，発行要求が出力される．

図 3 に，タグの比較を行う CAM セルの回路を示す．ウェイクアップ論理の 1 つのエントリは，タグ・ビット分の CAM セルからなる．図の左にある SRAM のセルはソース・オペランド・タグの 1 ビットを保持している．上部の水平の線はマッチ線と呼ばれ，タグが一致したことを示す．縦に 2 つ並べられたトランジスタは，タグ比較の結果にしたがってマッチ線をディスチャージするものである．

回路は次のように動作する．最初に，マッチ線がプリチャージされ，そしてデスティネーション・タグが放送される．もしソース・オペランド・タグとデスティネーション・タグが一致しなかったら，対応するマッチ線は縦に 2 つ並べられたトランジスタによりディスチャージされる．逆に，全てのビットが一致していたら，マッチ線は H を維持する．

3.2 セレクト論理

本研究では，セレクト論理は，3 節の最初で説明したようなプレフィクス・サム論理で実装することを仮定する．一般に，プレフィクス・サム論理は，入力，出力ともに N 個あり，i 番目の出力 $(0 \leq i \leq N-1)$ は，0 番目から i 番目までの入力の値の合計である．**図 4** に，例として $N = 16$ の時のプレフィクス・サム論理の回路図を示す[11]．クリティカル・パス上の加算器の数は，$\log_2 N$ 個になる．

この論理をセレクト論理として使う時，それぞれの入力は発行要求のブール値とし，プレフィクス・サム論理は入力の値を算術的に加える．もし $(i-1)$ 番目の出力値が発行幅より

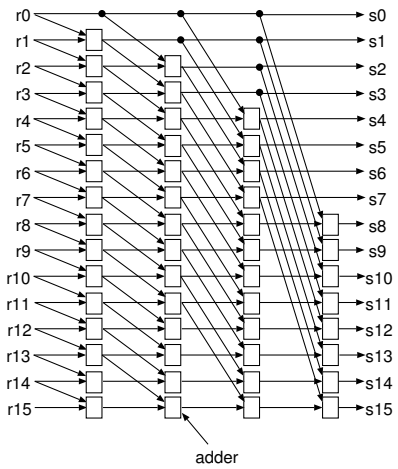


図 4 プレフィクスサムの回路 ($N = 16$)
Fig. 4 Circuit of prefix-sum ($N = 16$).

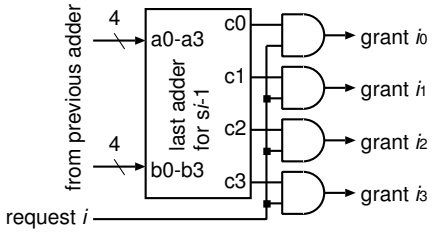


図 5 発行許可信号を出力する回路
Fig. 5 Grant output circuit ($IW = 4$).

小さく，そして i 番目の発行要求が真の場合，要求は許可される．**図 5** に，発行許可信号を出力する回路を示す．図に示されている加算器の入出力のエンコーディングには，後に述べるワンホット・エンコーディングを使用している．図の granti_u 信号は，タグ RAM の i 番目のエントリの u 番目のワード線につながっている．

遅延を短くするための新しい加算器の回路が，五島によって提案されている[11]．例えば，$IW = 4$ のセレクト論理において，加算器の入出力の値は，次の 5 つの値で十分である: “0”，“1”，“2”，“3”，“≥ 4”．そこで五島は，入出力を**表 1** に示すように 4 ビットでワンホット・エンコーディングし，**図 6** に示す加算器を提案した．加算器は，4 ビットの入力 a と入力 b を加え，和 c を出力する．直感的に分かるように，この加算器は従来の加算器より高速である．本研究でもこの加算器を用いた．

表 1 $IW = 4$ の場合の加算器の入出力のためのワンホット・エンコーディング
Table 1 One-hot encoding for adder input and output ($IW = 4$).

value	0	1	2	3	≥ 4
encoding	1000	0100	0010	0001	0000

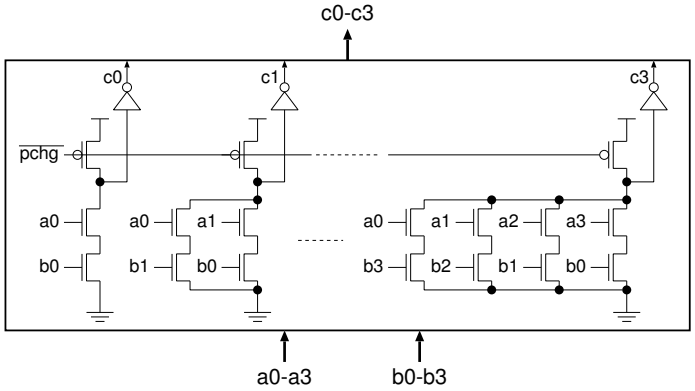
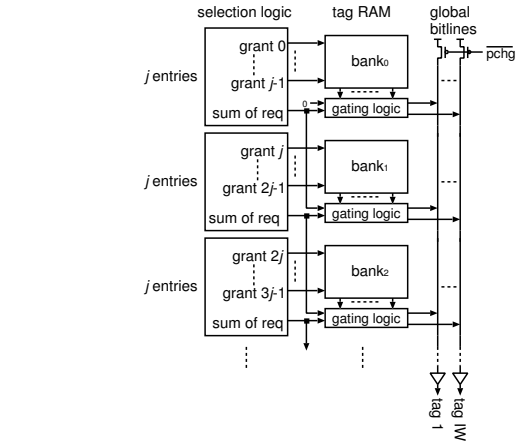


図 6 セレクト論理で用いる加算器 ($IW = 4$)
Fig. 6 Adder for selection logic ($IW = 4$).

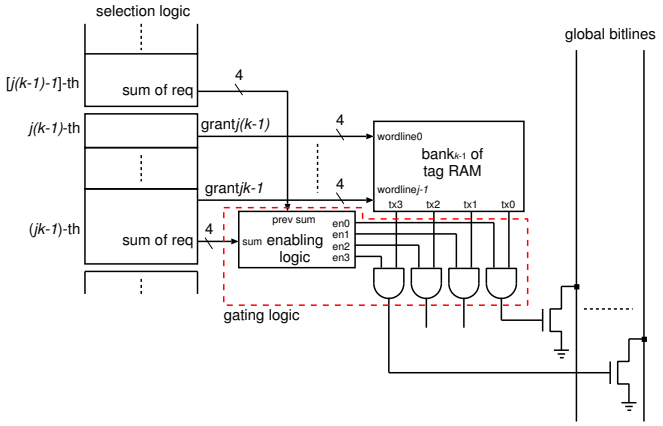
3.3 タグ RAM

タグ RAM はアドレス・デコーダのない SRAM からなる．IW 個のポートを持ち，1 エントリにつき，セレクト論理からの IW 本の許可信号が，IW 本のワード線に直接接続される．モノリシック構成では，アサートされたワード線に接続されたセルが保持する最大 IW 個のデスティネーション・タグがビット線に出力され，センスアンプで増幅され，出力される．

タグ RAM は，通常の RAM と同様，バンク化によってアクセス時間を減少させることが可能である．しかし，タグ RAM の場合，アクセスはアドレスによってなされないため，バンク化は単純ではない (通常の RAM では，バンクの選択はアドレス・ビットの一部を使って行われる)．**図 7**((a) は概要，(b) は詳細) に，j エントリのバンクで $IW = 4$ の場合のタグ RAM の構成を示す．バンクの出力 tx_0–tx_3 を有効にするゲーティング論理を追加して



(a) 概要



(b) 詳細 ($IW = 4$)

図 7 バンク化されたタグ RAM の構成
Fig. 7 Organization of banked tag RAM.

いる．ゲーティング論理の中のイネーブル論理は，発行キューの最初のエントリから対応するバンクの最後のエントリまでの発行要求の数と，1 つ前のバンクの最後のエントリまでの発行要求の数を観測することにより，バンクのどのポートを有効とすべきかを特定する．例えば，$j = 8$ で，$\sum_{i=0}^{7} \mathrm{request}i = 1$，$\sum_{i=0}^{15} \mathrm{request}i = 3$ のとき，tx_1 と tx_2 の出力を許可する．**表 2** に，イネーブル論理の真理値表を示す．ここで， c0–c3 はプレフィクス・サムの最後の加算器の出力である．前述の例は，真理値表の 2 番目のセクションの 3 行目の場合である．通常の RAM のバンク化と違い，この方法によるタグ RAM のアクセスには，潜在的に 2 つのクリティカル・パスが考えられることに注意されたい．1 つは，発行許可信号がバンク化された RAM を通ってタグを出力するまでのパスであり，もう 1 つは，発行要求の和信号がゲーティング論理を通ってタグを出力するまでのパスである．

表 2 イネーブル論理の真理値表
Table 2 Truth table of enabling logic.

prev sum c0–c3	sum c0–c3	enable en0–en3
1 0 0 0	1 0 0 0	0 0 0 0
1 0 0 0	0 1 0 0	1 0 0 0
1 0 0 0	0 0 1 0	1 1 0 0
1 0 0 0	0 0 0 1	1 1 1 0
1 0 0 0	0 0 0 0	1 1 1 1
0 1 0 0	0 1 0 0	0 0 0 0
0 1 0 0	0 0 1 0	0 1 0 0
0 1 0 0	0 0 0 1	0 1 1 0
0 1 0 0	0 0 0 0	0 1 1 1
0 0 1 0	0 0 1 0	0 0 0 0
0 0 1 0	0 0 0 1	0 0 1 0
0 0 1 0	0 0 0 0	0 0 1 1
0 0 0 1	0 0 0 1	0 0 0 0
0 0 0 1	0 0 0 0	0 0 0 1
0 0 0 0	0 0 0 0	0 0 0 0

3.4 レイアウト

図 8 に，仮定したウェイクアップ論理，セレクト論理，タグ RAM の配置を示す．各部品内のレイアウトは，MOSIS のデザイン・ルール[5] を仮定し，マニュアルで作成した．その結果，各部品の 1 エントリの高さはほぼ等しくなったので（$IW = 4$ の場合，セレクト論理，タグ RAM のウェイクアップ論理に対する高さの割合はそれぞれ 0.98 と 1.00），本研究では，それらのうち最も大きな値を発行キューのエントリの高さとした．このため，発行要求と許可信号の線は曲がることなく，水平に配置される．

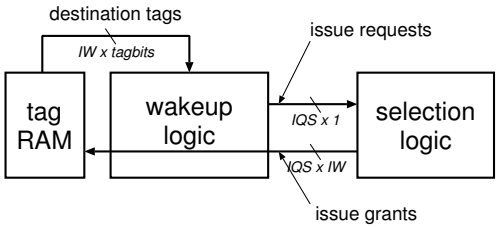


図 8 ウェイクアップ論理，セレクト論理，タグ RAM のレイアウト
Fig. 8 Layout of wakeup logic, selection logic, and tag RAM.

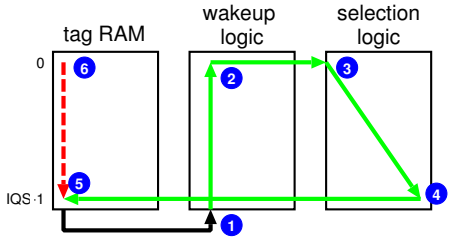
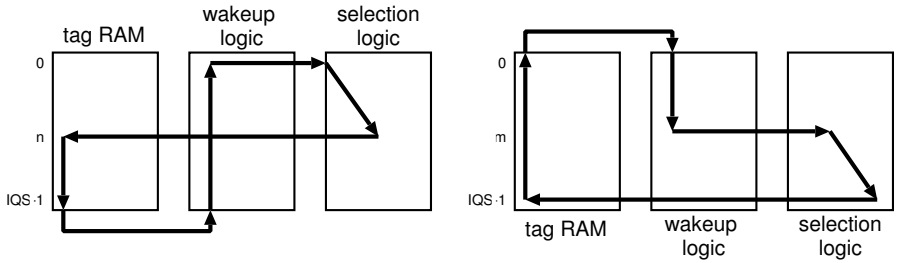


図 9 各部品のクリティカル・パスを単純につなげたパス
Fig. 9 Path simply connectiong the critical path of each component

4. 正しいクリティカルパス

それぞれの部品のクリティカル・パスを単純につないだパスは，論理的にはつながっていない．例えば，**図 9** に示すように，タグ RAM の出力とウェイクアップ論理のタグ・ドライバが最後のエントリで接続される配置を考える．図のマーク (1) から発し，(2)〜(5) と通り，(1) に帰る信号を考える．この場合，信号はタグ RAM のクリティカル・パスの一部である (6)〜(5) を通らない．

正しいクリティカル・パスは，**図 10** に示すように 2 つ考えられる．1 つは，タグ RAM とウェイクアップ論理のタグ・ドライブの出力が最後のエントリでつながっている場合であり，これを構成 A と呼ぶこととする．もう 1 つは反対に，最初のエントリでつながっている場合であり，これを構成 B と呼ぶこととする．構成 A では，パスはセレクト論理の n 番目のエントリで折り返している．一方，構成 B では，ウェイクアップ論理の m 番目のエントリで折り返している．本研究では，最初に構成 A と構成 B それぞれの最も遅延の長いパスを見つける．そして 2 つのパスの遅延時間を比較し，遅延時間の短い方のパスを持つ構成をより良い構成とし，そのパスをクリティカル・パスと決定する．



(a) 構成 A　　(b) 構成 B

図 10 考えられる正しいクリティカル・パス
Fig. 10 Possibly correct critical paths.

5. 評　　価

SPICE によるシミュレーションを行い，$IW = 4$ で様々なサイズの発行キューの遅延時間を評価した．ここで，タグのサイズによる遅延時間の変化は小さいので，タグのビット長は 8 で固定した．

32nm の LSI 技術を仮定した．また，SPICE のトランジスタ・モデルとして，アリゾナ州立大学の Nanoscale Integration and Modeling group による予測モデルである predictive transistor model（PTM）を用いた[4),10)]．単位長さあたりの配線抵抗と配線容量は，ITRS によって予測されたものを使用した[6)]．また，遅延を減少させるため，長い配線にはリピータを挿入した．リピータの挿入間隔は，実験により最適化した．

5.1 ウェイクアップ論理の遅延

図 11 に，様々な発行キュー・サイズでのウェイクアップ論理の遅延の測定結果を示す．各棒グラフは，タグ・ドライブ，タグ・マッチ（タグの比較），OR（比較結果の OR），レディ（レディ・フラグを保持する SR ラッチへの書き込み），AND（2 つの OR の AND）の遅延で分けられている．図に示すように，小さなキューではタグ・マッチとレディが遅延の多くを占めている．しかし，発行キュー・サイズが大きくなると，タグ・ドライブの遅延

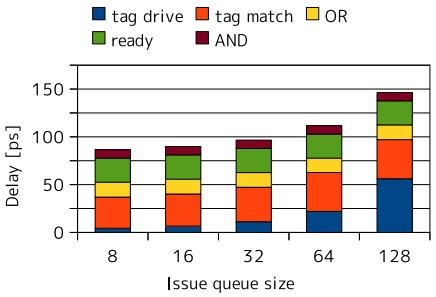


図 11　ウェイクアップ論理の遅延
Fig. 11　Delay of wakeup logic.

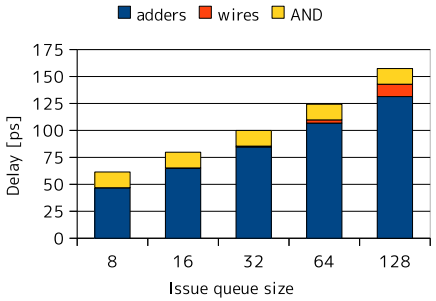


図 12　セレクト論理の遅延
Fig. 12　Delay of selection logic.

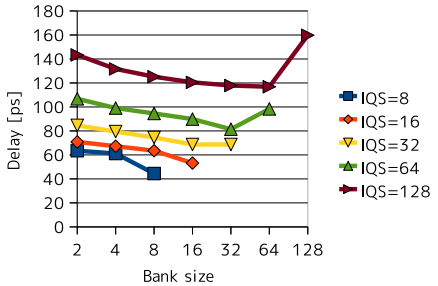


図 13　様々なバンク・サイズでのタグ RAM の遅延
Fig. 13　Tag RAM delay for various bank sizes.

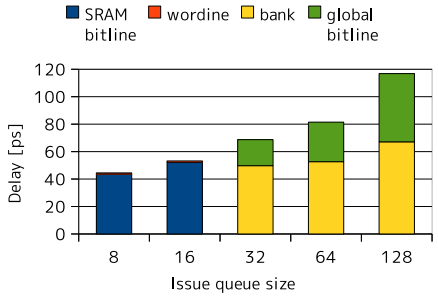


図 14　最適にバンク化されたタグ RAM の遅延
Fig. 14　Delay of tag RAM with best bank configuration.

が大幅に増えている．

5.2　セレクト論理の遅延

図 12 に，様々な発行キュー・サイズでのセレクト論理の遅延の測定結果を示す．各棒グラフは，加算器の遅延の和，配線遅延の和，発行要求信号とプレフィクス・サムの出力の AND の遅延に分けられている．図に示すように，加算器のゲート遅延が支配的である．遅延は，3.2 節で述べたように，クリティカル・パス上の加算器の数が $\log_2 IQS$ で増えるので，遅延も $O(\log_2 IQS)$ で増加している．

5.3　タグ RAM の遅延

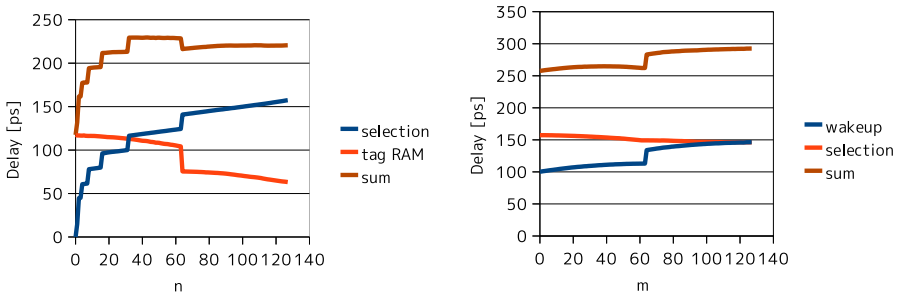
図 13 に，様々な発行キュー・サイズとバンク・サイズでのタグ RAM の遅延の評価結果を示す．ここで，バンク・サイズが発行キュー・サイズ (IQS) が等しい点では，バンク化は行われていないことを注意しておく．図からわかるように，小さなキュー ($IQS \leq 16$) では，ビット線が短く，また，ビット線につながっている RAM のセルの数が少ないので，バンク化は効果的ではない．これに対して，大きなキューでは，逆の理由でバンク化は効果的である．3.3 節で述べたように，バンク化されたタグ RAM には 2 つのクリティカル・パスが考えられる．実験の結果，32 エントリの場合ではゲーティング論理を通ったパスの方が長く，64 エントリと 128 エントリの場合ではバンク化された RAM を通る方のパスが長いことが分かった．

図 14 に，最適にバンク化された場合の様々な発行キュー・サイズにおけるタグ RAM の遅延を示す．バンク化されていないタグ RAM の棒グラフは，ビット線とセンスアンプの遅延の和と，ワード線の遅延に分けられている．バンク化されているタグ RAM の棒グラフは，バンクの遅延と，グローバルなビット線の遅延に分けられている．$IQS \leq 16$ の時はタグ RAM はバンク化はされておらず，$IQS \geq 32$ の時はバンク化が行われていることに注意されたい．図からわかるように，小さなキューではビット線の遅延が支配的である (ワード線の遅延は図では見えないくらい小さい)．対して大きなキューでは，サイズが大きくなるほどバンクの遅延が支配的であるが，グローバル・ビット線の遅延も大きい．

5.4　クリティカルパス

クリティカル・パスを見つけるために，4 節で述べたように，2 つのパスについて n と m を変化させて遅延を測定した．**図 15** に，例として，128 エントリの発行キューにおいて n と m を変化させた時の部品の遅延の測定結果を示す．構成 A のグラフでは，セレクト論理とタグ RAM の遅延，及びそれらの合計を示している．ここで，ウェイクアップ論理の遅延は n に関わらず一定なので示していない．構成 B のグラフでは，ウェイクアップ論理とセレクト論理の遅延，及びそれらの合計を示している．タグ RAM の遅延は m に関わらず一定なので示していない．

構成 A では，セレクト論理の遅延が $O(\log_2 n)$ で増加している．これは，信号が通るパス上の加算器の数がこの割合で増加しているからである．他方で，タグ RAM の遅延は比較的一定である (タグ RAM の遅延における不連続な降下は，信号がどのバンクを通過するかによって生じていている．小さなエントリ番号に対応するバンクを通る信号は，グローバル・ビット線を完全に通過しなければならないが，大きなエントリに対応するバンクを通る信号の方は，一部を通過するのみである)．この構成で最も遅いパスは，$n = 43$ で折り返す



(a) 構成 A　　(b) 構成 B

図 15　発行キューサイズが 128 において，n と m を変化させた際の遅延
Fig. 15　Delay of components for various turning points when issue queue size is 128.

パスであった．

構成 B では，ウェイクアップ論理の遅延は m の増加によってゆるやかに増加する (遅延の不連続な上昇は，タグ線に挿入されているリピータによって引き起こされているものである．評価したパスは，$n \leq 63$ では，リピータを通らない)．これに対して，セレクト論理の遅延はほとんど一定である．これは，最後のエントリからの出力への加算器の数は m を変化させても一定であるからである (図 4 参照)．グラフの小さな傾きは，配線遅延によるものである．この構成で最も遅いパスは，$m = 127$ で折り返すパスであった．

構成 A と B の最も遅いパスを比較すると，遅延が小さいのは構成 A のパスである．よって，構成 A がより良い構成であり，求められたパスがクリティカル・パスである．

表 3 に，構成 A と B において最も遅いパスの遅延を，n と m の値とともに示す．構成 A と B における最も遅いパスの遅延は，(特に小さなキューで) ほぼ等しいが，構成 A の方がわずかに短い．

図 16 に，様々なサイズの発行キューに対して異なる 3 つの最適化レベルにおける発行キューの遅延をまとめる．最適化レベルのうち，simple レベルでは，タグ RAM のバンク化を行わず，発行キュー全体の遅延を，各部品のクリティカル・パスの遅延を加えて得る．not-banked レベルでは，タグ RAM を必要に応じてバンク化するが，正しいクリティカル・パスを測定する．optimized レベルは，タグ RAM を必要に応じてバンク化し，正しいクリティカル・パスを測定する．各棒グラフは，ウェイクアップ論理，セレクト論理，タグ RAM，

表 3　構成 A と B において最も遅いパスの遅延
Table 3　Delay of slowest paths in comfigurations A and B.

発行キューサイズ	構成 A		構成 B	
	n	delay [ps]	m	delay [ps]
8	7	247	7	247
16	15	276	15	277
32	31	309	28	319
64	63	349	61	369
128	43	430	127	464

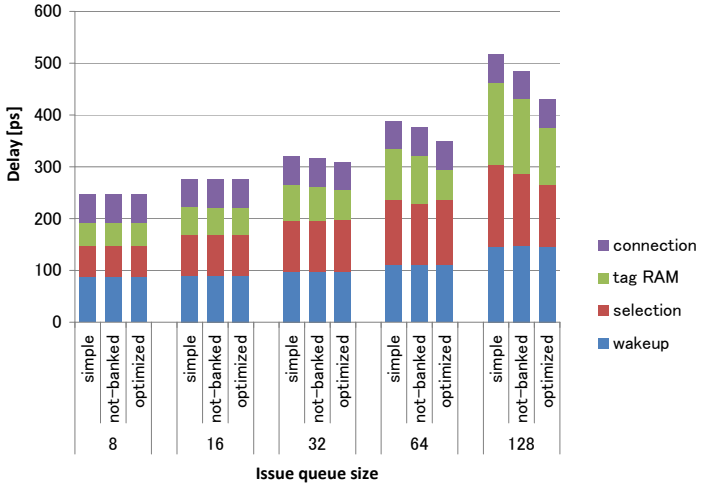


図 16　設計の最適化レベルに対する発行キューの全体の遅延
Fig. 16　Total delay of issue queue with design optimizations.

connection の遅延に分けられている．connection はウェイクアップ論理とセレクト論理とタグ RAM をつなぐ配線遅延の合計である．図に示すように，最適化レベルを optimized にすることにより，発行キュー・サイズが大きくなるほど，発行キューの遅延をより多く減少させることができることがわかる．not-banked レベル，simple レベルと比較すると，128 エントリのキューの場合，それぞれ，13%と 20%と大きく減少させることができた．

Vol.2011-ARC-196 No.17
2011/7/28

6. ま と め

本論文では発行キューのタグ RAM をバンク化し，正しいクリティカル・パスを求めたうえで，様々なサイズの発行キューについての遅延時間を示した．これらの最適化を行ったことにより，我々の調査したサイズにおいては，最適化を行わない場合に比べて，最大 20%遅延を削減することができた．

謝辞

本研究の一部は，日本学術振興会 科学研究費補助金基盤研究 (C)（課題番号 22500045）による補助のもとで行われた．また，本研究は東京大学大規模集積システム設計教育研究センターを通し，シノプシス株式会社の協力で行われたものである．

参 考 文 献

1) Goshima, M., Nishino, K., Nakashima, Y., Mori, S., Kitamura, T. and Tomita, S.: A high-speed dynamic instruction scheduling scheme for superscalar processors, *Proceedings of the 34th Annual International Symposium on Microarchitecture*, pp. 225–236 (2001).
2) Gwennap, L.: AMD Bulldozer plows new ground, *Microprocessor Report* (2010).
3) Gwennap, L.: Sandy Bridge spans generations, *Microprocessor Report* (2010).
4) http://www.eas.asu.edu/˜ptm/: .
5) http://www.mosis.com/: .
6) International Technology Roadmap for Semiconductors: (2010 update (http://www.itrs.net/)).
7) Palacharla, S., Jouppi, N.P. and Smith, J.E.: Quantifying the complexity of superscalar processors, Technical report, CS-TR-1996-1328, University Wisconsin (1996).
8) Palacharla, S., Jouppi, N. P. and Smith, J. E.: Complexity-effective superscalar processors, *Proceedings of the 24th Annual International Symposium on Computer Architecture*, pp.206–218 (1997).
9) Sassone, P. G., J. Rupley, I., Brekelbaum, E., Loh, G. H. and Black, B.: Matrix scheduler reloaded, *Proceedings of the 34th Annual International Symposium on Computer Architecture*, pp.335–346 (2007).
10) Zhao, W. and Cao, Y.: New generation of predictive technology model for sub-45nm design exploration, *Proceedings of the 7th International Symposium on Quality Electronic Design*, pp.585–590 (2006).
11) 五島正裕：Out-of-Order ILP プロセッサにおける命令スケジューリングの高速化の研究，京都大学，博士論文 (2004).
12) 甲良祐也，安藤秀樹：命令発行キューの遅延時間評価，2010 年先進的計算基盤システムシンポジウム SACSIS 2010，pp.45–52 (2010).