

命令発行キューの遅延時間評価

甲 良 祐 也^{†1} 安 藤 秀 樹^{†1}

アウト・オブ・オーダー実行により、メモリ・ウォール問題を解決するためには、in-flight 命令を大幅に増加させる必要がある。そのためには種々の資源サイズを増大させる必要があり、その一つに発行キューがある。しかし、発行キューのサイズを増加させるとクロック周波数を低下させてしまう。これに対して我々は、IPC の低下を抑制しつつ、発行キューのパイプライン化によるクリティカル・パスの分割を行いクロック周波数の低下を防ぐ方式を検討している。この手法を定量的に評価するためには、種々のエン트리数を持つ発行キューの遅延時間を知り、与えられたクロック周波数で、何段にパイプライン化しなければならないかを知らなければならない。そこで、本論文では、発行キューの遅延時間を回路シミュレータ HSPICE を用いて評価した。評価においては、これまで行われた遅延評価と異なり、遅延に大きく影響を与える長いグローバル配線に、適切にリピータを挿入し、配線遅延の削減を行っている。評価の結果、大きなサイズの発行キューでは、配線遅延が支配的であるが、例えば、発行幅 8 の発行キューのタグ・ドライブ遅延の増加率は、リピータがない場合、ウィンドウ・サイズ $WS(\geq 64)$ に対し、 $O(WS^{1.8})$ のところ、挿入すれば、 $O(WS^{1.2})$ に抑えられることがわかった。また、発行キューのクリティカル・パスをなすウェイクアップ論理、セレクト論理、タグ RAM の遅延はほぼ等しく、全遅延は、ウィンドウ・サイズに対し、 $O(WS^{0.9})$ 、つまり、ほぼ比例して増加することがわかった。

Evaluation of Issue Queue Delay

YUYA KORA^{†1} and HIDEKI ANDO^{†1}

The number of in-flight instructions is required to be increased significantly to solve the memory-wall problem by using out-of-order execution. To implement this, it is necessary for the size of various resources to be increased. One of such resources is the issue queue. However, if increasing the issue queue size degrades the clock rate. To solve this problem, we are studying a scheme that divides the critical path of the issue queue by pipelining, while suppressing the deterioration of IPC. To evaluate this scheme quantitatively, we must know the delay of the issue queue with the various number of entries, and determine that we should find what number of the stages we must pipeline the issue queue into for a given clock rate. This paper evaluates the delay of the issue queue using

the circuit simulator, HSPICE. Unlike previous studies, to reduce the delay of the long global wires, which affects the delay significantly, we appropriately insert repeaters into them. Our evaluation results show that the wire delay is dominant in a large issue queue. However, for an example, the rate of the increase of the tag drive delay in the 4-issue queue is suppressed with $O(WS^{1.2})$ by inserting the repeaters, while $O(WS^{1.8})$ without the repeaters, where the window size, WS , is more than or equal to 64. We also found that the delays of the wakeup logic, selection logic, and tag RAM, which compose the critical path of the issue queue, are approximately identical, and the total delay grows with $O(WS^{0.9})$, which means that the delay grows approximately lineally.

1. はじめに

プロセッサとメモリ間の速度差は非常に大きく、一般に、メモリ・ウォールと呼ばれている。メモリ・ウォールは、メモリ・インテンシブなプログラムの性能を著しく制限している。積極的なアウト・オブ・オーダー実行は、この問題を解決する 1 手法である¹⁾。これは、メモリ・レベル並列性 (MLP: memory-level parallelism) を利用し、1 メモリ・アクセス当たりの実効的なペナルティを減少させようとするものである。

アウト・オブ・オーダー実行でメモリ・ウォール問題を解決するには、プロセッサがサポートする in-flight 命令を、現在の商用プロセッサで実現されているその数より大幅に増加させる必要がある。このために重要なハードウェアとしては、リオーダー・バッファ、発行キュー、レジスタ・ファイルがある。しかし、これらのサイズを増加させると、一般には、クロック・サイクル時間を悪化させるという問題がある。本研究では、発行キューに焦点を当てる。

動作時間が長くクロック速度に悪影響を与える論理において、その悪影響を除く一般的な手法としてパイプライン化がある。発行キューをパイプライン化することは回路的には可能であるが、一方で、最適な命令スケジューリングが行えず、IPC が低下し、必ずしも得策ではない。IPC が低下する理由は、発行された命令の結果タグは直後のサイクルに発行キューに放送され、依存する命令を発行させなければ、互いに依存のある命令を連続したサイクルで発行できないためである。

これに対して、我々はこれまで、命令の発行タイミングを予測することにより、パイプライン化された発行動作を早期に開始させる手法を提案した¹⁰⁾。この手法を定量的に評価する

^{†1} 名古屋大学大学院工学研究科
Graduate School of Engineering, Nagoya University

には、種々のエンタリ数を持つ発行キューの遅延時間を知り、何段にパイプライン化しなければならぬかを知らなければならぬ。そこで、本論文では、発行キューの遅延時間を回路シミュレータ HSPICE を用いて評価することを目的とする。

本論文では、まず、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構成について説明する。4 節では長い配線の遅延縮小方法について述べる。5 節で評価結果を示し、6 節でまとめる。

2. 関連研究

発行キューの構造に関する研究は多く存在するが(例えば、文献 5)), それが回路にまで及ぶ研究はそれほど多くない。以下、回路に着目した研究について述べる。

2.1 ウェイクアップ論理

Palacharla らは、プロセッサを構成する種々の部品の遅延測定を行った⁶⁾。その結果、将来に向けて拡大が望まれる発行キューの遅延が、プロセッサの部品の中で最もクリティカルなものとなることを示した。彼らは、ウェイクアップ論理を構成する回路として、最も基本的な CAM で構成した場合について測定を行い、大きなサイズの発行キューにおいては、タグをドライブする遅延が最も問題となることを示した。タグ・ドライブ遅延は配線遅延によって支配されるから、それはウィンドウ・サイズの 2 乗で増加するとした。しかし、彼らは配線遅延を短縮するために通常行われるリピータの挿入を行っておらず、配線遅延は過大評価されている。

五島らは、CAM ではなく RAM によりウェイクアップ論理を構成する手法を提案した³⁾。彼らは、この RAM のことを依存行列と呼んでいる。依存行列の各行、各列は、発行キュー内の命令に対応し、各セルは、行に対応する命令が列に対応する命令に依存している場合、1 がセットされる。命令が発行されると、対応する列のワード線がアサートされ、その結果、接続されているセルの内容が行ビット線に読み出される。セルに 1 が格納されていた行では、レディ・ビットがセットされる。ウェイクアップ論理を RAM で構成することにより、従来の CAM の場合と異なり、比較器が不要になる。また、1 エンタリ当たりのサイズが小さくなり、信号を放送するグローバル配線長を短くすることができる。このため、配線遅延が大きな問題となる大きな発行キューにおいて、遅延を大幅に短縮できる。

Sassone らは、五島らの提案した依存行列のサイズを縮小する方法を提案した⁷⁾。この手法では、一般に依存行列が疎であることに着目し、列を命令に動的に割り当て、列数を減少させた。

2.2 セレクト論理

Palacharla らは調停回路を用いて、発行要求を出力している命令の中から、最も優先度の高い命令を選択する回路を構成し、遅延評価を行った⁶⁾。一般に、多くの要求を一度に調停するのは大きな遅延を要するため、Palacharla らは、4 命令の中から最も優先度の高い命令を選択する小さな調停器を直列に接続することにより、多くの要求を調停する回路を示した。この調停回路は、全要求の中から 1 つだけを許可するものであるが、優先度を考慮し複数の許可を出すよう拡張するには、当該調停回路を直列接続する必要があり、非常に遅延が大きくなるという欠点がある。

五島は、プレフィクス・サムによるセレクト論理を提案した¹¹⁾。この手法は、各命令より優先度の高い命令で発行要求を行っているものの数を加算器で数えるものである。その数が命令発行幅より少なければ、当該命令の発行要求は許可される。この方式は、上述した調停回路による方式と異なり、複数の要求に許可を出す場合においても、遅延増加が少ない。欠点は加算器の遅延が大きいことにあるが、五島は、加算器の入出力のエンコードを工夫することによりこの問題を緩和した。

Sassone らは、RAM によるセレクト論理を提案した⁷⁾。各行、各列は、発行キュー内の命令に対応し、各セルは、行に対応する命令より列に対応する命令の方が優先度が高い場合、1 がセットされる。発行要求が出されると、対応する列のワード線をアサートされる。接続されているセルが 1 を保持しているなら、その行の発行要求が(出されているなら)ネゲートされる。彼らは、さらに命令をグルーピングし、セルを共有することにより RAM を小さくすることも提案した。この方式は、基本的に、ウィンドウ・サイズの 2 乗で RAM が大きくなるという欠点があるものの、RAM で構成されるため、遅延は小さく抑えられる。また、要求許可の数によって遅延が増加することもない。

3. 発行キューの構成

発行キューはフェッチした命令を保持し、その中から発行する命令を選択する論理である。発行キューは図 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、パイロード RAM からなる。ウェイクアップ論理の各エンタリは、対応する命令の 2 つのソース・オペランドのタグとデータ依存の状態(解決か未解決か)を示すレディ・フラグを保持する。両方のデータ依存が解決した場合、セレクト論理に発行要求を出す。セレクト論理では、資源競合を考慮し、発行許可を出す。発行許可を受け、パイロード RAM から対応する命令の情報が出力され、発行が完了する。発行許可信号は同時に、タグ RAM にも出力され、発行さ

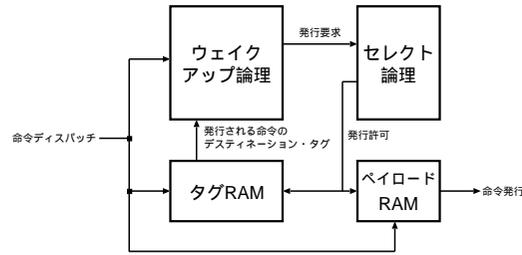


図 1 発行キューの構成

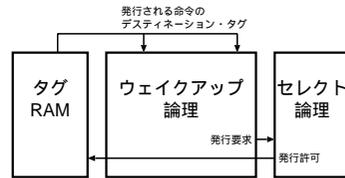


図 2 発行キューの配置

れる命令のデスティネーション・タグを得る。これをウェイクアップ論理の全エンタリに放送し、レディ・フラグを更新する。

発行キューのクリティカル・ループは、ウェイクアップ論理 セレクト論理 タグRAM を経てウェイクアップ論理に戻るパスである。本論文では、このクリティカル・ループの遅延を測定する。

本論文で仮定した、ウェイクアップ論理、セレクト論理、タグRAMの全体配置を図2に示す。

2節で述べたように、発行キューを構成するウェイクアップ論理とセレクト論理を実現する回路には種々のものがある。1節で述べたように、本研究は、発行キューのパイプライン化の研究のために発行キューの遅延を測定するものである。この目的の下では、発行キューを実現する回路は基本的なものが適当である。そこで、ウェイクアップ論理にはCAMを用いた回路を採用する。セレクト論理には、基本的な回路であることに加え、調停する機能ユニットの数によって遅延が大きく延びることがないプレフィクス・サム回路を採用する。

3.1 ウェイクアップ論理

図3に、CAMによるウェイクアップ論理の回路構成を示す。タグRAMから読み出され

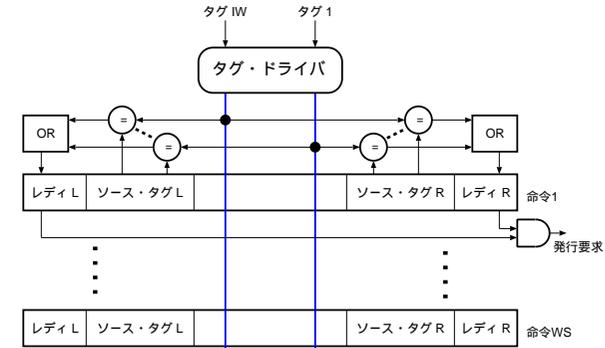


図 3 ウェイクアップ論理

た IW 個のデスティネーション・タグは、タグ・ドライバにより、ウェイクアップ論理の WS 個の全エンタリに放送される。ここで、 IW は命令発行幅、 WS はウィンドウ・サイズである。各エンタリは、2つのソース・オペランド・タグ(図のソース・タグLとソース・タグR)を保持しており、放送された各デスティネーション・タグとの比較が行われる。一致するものがあれば、対応するレディ・フラグ(レディLまたはレディR)がセットされる。2つのレディ・フラグがともにセットされれば、発行要求が出される。

タグ比較を行うCAMの1セル(1ビットの比較を行う)の回路を、図4に示す。ウェイクアップ論理1エンタリのCAMは、同図の回路をタグ・ビット幅ほど横に接続したものである。図4の中央上部のSRAMに、レディ・フラグを保持する。横に貫かれた赤い線は、一致の論理値を出力するマッチ線であり、左右にある直列接続のトランジスタは、レディ・ビットとタグの比較結果に応じてマッチ線をディスチャージするプルダウン・トランジスタである。

次のように動作する。まず、マッチ線をプリチャージした後、タグをドライブする。タグとレディ・ビットが1ビットでも不一致なら、直列接続されたプルダウン・トランジスタによって、マッチ線がディスチャージされ、L(つまり不一致)を出力する。もし全てのビットが一致するなら、マッチ線はチャージ・アップされたままであり、H(つまり一致)を出力する。

CAM1セルのレイアウトを図5に示す。トランジスタの配置は、図4に示した回路図と同じである。タグ線は、トランジスタ領域の外側に配置する。RAMへの書き込みビット線

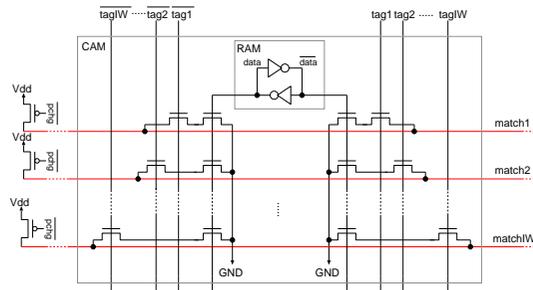


図4 タグの値比較を行う回路図

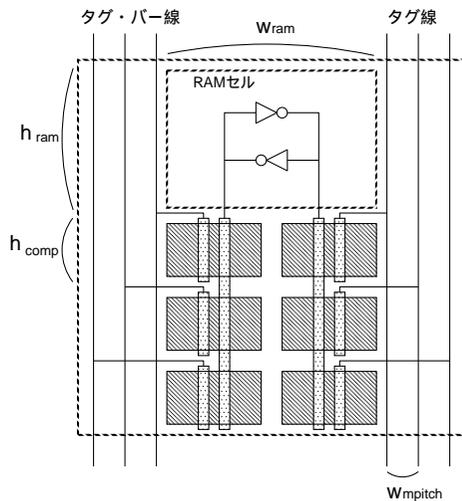


図5 CAMセルのレイアウト

も外側に配置する(図では省略している)。これらは、中間層メタルを仮定した。幾何的パラメータを表1に示す。 w_{ram} , h_{ram} の値は、文献9)から得た。 w_{mpitch} の値は、LSIプロセスにおける典型的な値であり、文献8)に記載されている。 h_{comp} の値は、実際にレイアウトして求めた。なお、 λ は、LSIプロセスにおける最小加工寸法の1/2の値である。

3.1.1 タグ・ドライブ回路

図6に、シミュレーションしたタグ・ドライブの等価回路を示す。ウェイクアップ論理の

表1 CAM1セルの幾何的パラメータ

RAMセルの横幅	w_{ram}	20λ
タグ線、書き込みビット線のピッチ	w_{mpitch}	7λ
RAMセルの高さ	h_{ram}	40λ
プルダウン・トランジスタの高さ	h_{comp}	16λ

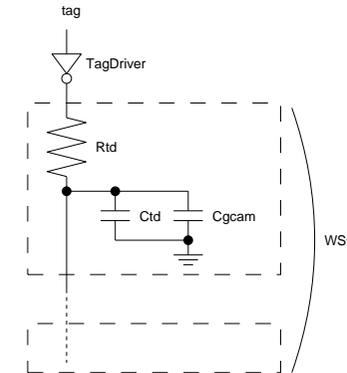


図6 タグ・ドライブの等価回路

上端にドライバ(1つのインバータ)を配置し、縦断するタグ線をドライブする。タグ線の等価回路は、ウェイクアップ論理1エントリにつき、配線抵抗 R_{td} 、配線容量 C_{td} 、CAMのプルダウン・トランジスタのゲート容量 C_{gcam} で構成されるものであり、それを直列に接続した回路とした。

発行キュー1エントリ当たりの配線長は、図5に示すCAMセルの高さである。すなわち、

$$h_{ram} + h_{comp} \times IW$$

である。

3.1.2 タグ・マッチ回路

図7に、シミュレーションしたタグ・マッチの等価回路を示す。マッチ線の等価回路は、CAM1セルにつき、配線抵抗 R_{tm} 、配線容量 C_{tm} 、CAMの2つのプルダウン・トランジスタのドレイン容量 C_{dcam} , C_{dcamb} で構成されるものであり、それをタグのビット数 $PREG_{width}$ だけ直列に接続した回路とした。

CAM1セル当たりの配線長は、図5に示すCAMセルの幅である。すなわち、

$$w_{ram} + 4 \times w_{mpitch} \times IW$$

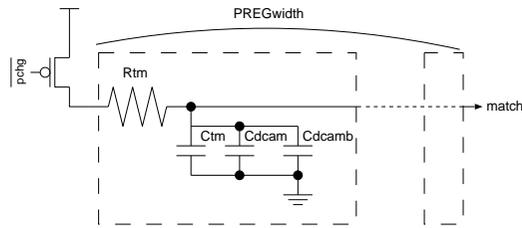


図 7 タグ・マッチの等価回路

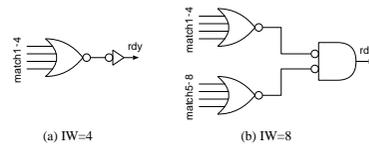


図 8 マッチ OR

である。

3.1.3 マッチ OR 回路

図 8 に、 $IW = 4, 8$ の場合のシミュレーションしたマッチ OR 回路を示す。高速化のためにファン・インは 4 つまでに制限し、 $IW \leq 4$ では同図 (a) に示すように、1 つの NOR ゲートとインバータで構成し、 $5 \leq IW \leq 8$ では (b) に示すように、NOR ゲートを 2 段に接続して構成した。

3.1.4 レディ AND 回路

2 つのレディ・フラグの AND をとるレディ AND 回路は、2 入力 NAND の出力にインバータを接続した回路とした。

3.2 セレクト論理

3 節の最初で述べたように、セレクト論理はプレフィクス・サム回路で構成した。一般に、プレフィクス・サムとは、長さ N の配列があり、1 番目の要素の値から i ($1 \leq i \leq N$) 番目の要素の値の和を i 番目の出力とするものである。これをセレクト論理に用いる場合、配列の各要素の値を発行要求のプール値とし、プレフィクス・サムの値が発行幅以下である位置に対応する発行要求を許可とすればよい。

図 9 に、配列の要素数が 16 の場合のプレフィクス・サムの回路を示す¹¹⁾。長方形は加算器、三角はドライバである。この回路では、プレフィクス・サムの論理を直接的に実現する

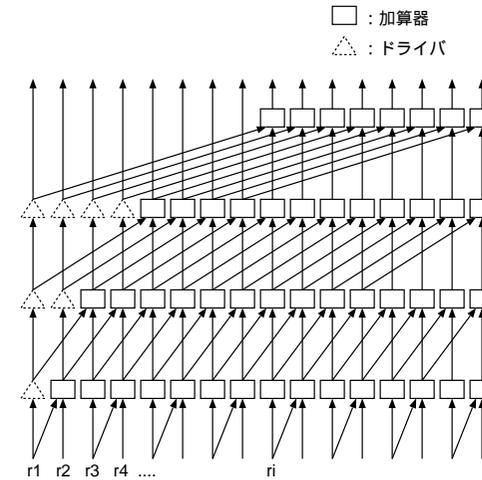


図 9 プレフィクス・サム

表 2 疑似ワンホット・エンコーディング

	a0	a1	a2	a3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4 以上	0	0	0	0

回路を基に、各加算器のファンアウトが 2 以下になるように、加算器が追加配置されている。

この回路をセレクト回路に用いる場合、使用する加算器の入出力値は、たとえば、 $IW = 4$ の場合、「0」、「1」、「2」、「3」、「4 以上」の 5 値で扱えば良い。五島は、この加算器を通常の加算器ではなく、入出力値を 4 ビットの疑似ワンホット・エンコーディングし (表 2 参照)、回路をドミノ論理で構成した加算器を提案した¹¹⁾。図 10 に回路図を示す。4 ビットの a と b を加え、4 ビットの和 c を出力する回路である。

同図から容易にわかるように、この回路は通常の加算器に比べ、高速である。本測定においても、この回路を用いる。

図 11 に、 $WS = 8$ の場合のセレクト論理のレイアウトを示す。入力から出力の経路を途中で 2 つに折り曲げてレイアウトしている。ウェイクアップ論理と高さを合わせて配置する

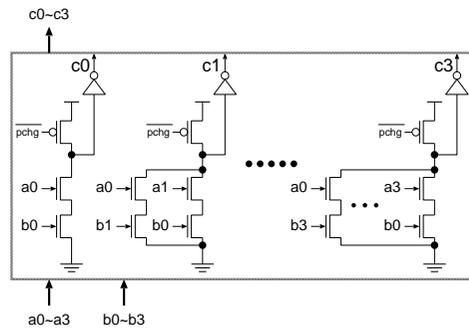


図 10 プレフィクス・サムで用いる加算器

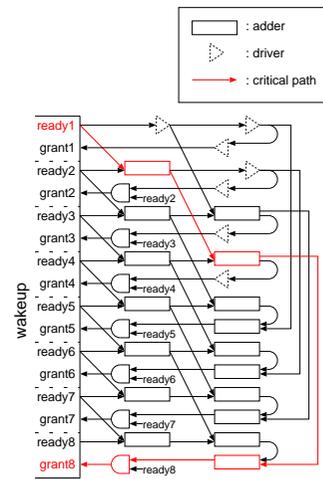


図 11 セレクト論理のレイアウト

ものとする。クリティカル・パスは、図中赤で示した経路であり、最も若い命令の発行要求から最も古い命令の発行許可を出力するまでである。配線は、横方向には短く、配線容量/抵抗を無視するとしたが、縦方向には長く、これらを考慮した。HSPICE のネットリストとしては、ウェイクアップ論理 1 エントリ分を縦断する毎に配線容量/抵抗を付加した。

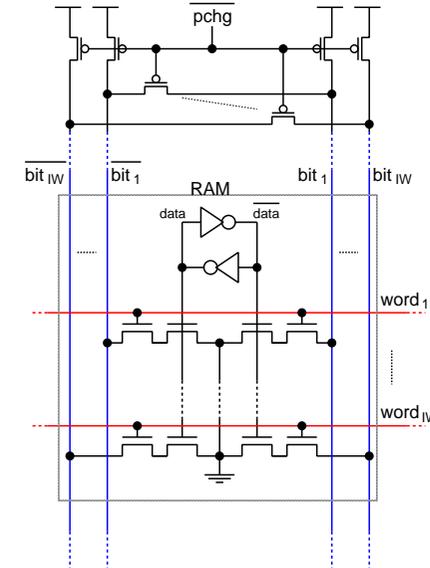


図 12 タグ RAM の構造

3.3 タグ RAM

タグ RAM はアドレス・デコーダのない SRAM で構成される。セレクト論理からの発行許可信号がワード線に結合され、セルに格納されたタグがビット線に出力され、センスアンプで増幅され読み出される。

タグ RAM の回路を図 12 に示す (セルは 1 ビットのみ示している)。書き込みに関する回路は、図が繁雑になるため省略している。

同図に示したセルと図 4 示した CAM セルを比較すればわかるように、タグ線をビット線に置き換えれば、これらの回路は等価である。よって、タグ RAM のセルの幾何的サイズは CAM セルと同一とした。

3.3.1 ワード線

図 13 に、シミュレーションしたワード線とそのドライバの等価回路を示す。タグ RAM の右端にドライバを配置し、横断するワード線をドライブする。ワード線の等価回路は、タグ RAM1 セルにつき、配線抵抗 R_{wd} 、配線容量 C_{wd} 、読み出しのアクセス・トランジスタのゲート容量 C_{gacc} および C_{gaccb} で構成されるものであり、それを直列に接続した回路と

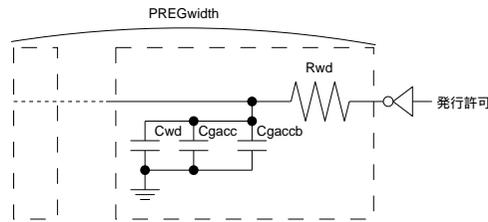


図 13 ワード線モデル

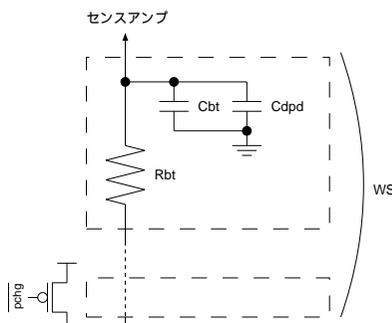


図 14 ビット線モデル

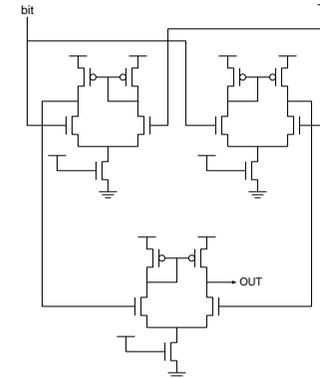


図 15 センスアンプ

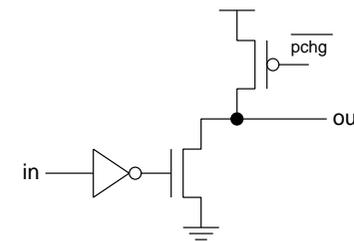


図 16 動的配線用リピータ

した。

3.3.2 ビット線

図 14 に、シミュレーションしたビット線とセルのプルダウン・トランジスタからなる等価回路を示す。ビット線の等価回路は、タグ RAM1 セルにつき、配線抵抗 R_{bt} 、配線容量 C_{bt} 、読み出しのアクセス・トランジスタのドレイン容量 C_{dpd} で構成されるものであり、それを直列に接続した回路とした。

3.3.3 センスアンプ

センスアンプの回路図を図 15 に示す。作動アンプを直列につないだ構成とした。

4. 長い配線の遅延縮小

配線は、配線抵抗と配線容量の積からなる時定数で遅延の下限が与えられる。配線抵抗と配線容量は、配線長に比例するので、時定数は配線の長さの 2 乗に比例して長くなる。この

ため、配線が長くなると、遅延は非常に長くなる。この遅延を短くするには、配線の途中に適宜リピータを挿入することが有効である。

配線には、静的回路によって駆動される静的配線と、プリチャージとプルダウンによって値が定まる動的配線がある。リピータ回路は、それぞれにおいて適切な回路が必要である。本測定においては、リピータとして、静的配線には静的なインバータを、動的配線には図 16 に示す動的なバッファを使用した。

リピータの挿入位置は、次のようにして求めた。配線を n ($= 1, 2, 4, \dots$) 等分し、リピータを挿入し、部品全体の遅延時間を測定する。配線遅延は、 n に対してただ 1 つの極小値を持つので (A.1 節参照)、 n を 1 より順に大きくして測定し、極小となる n を求めた。

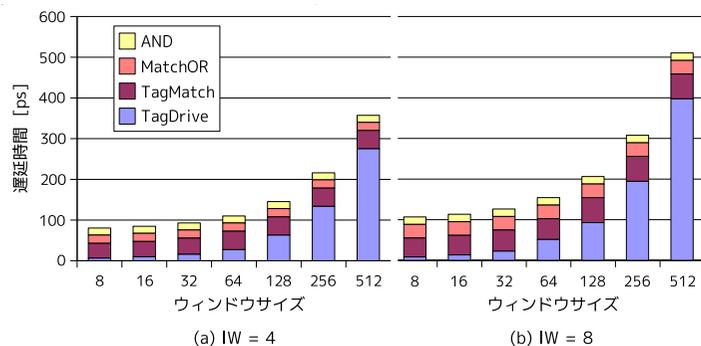


図 17 ウェイクアップ論理の遅延時間

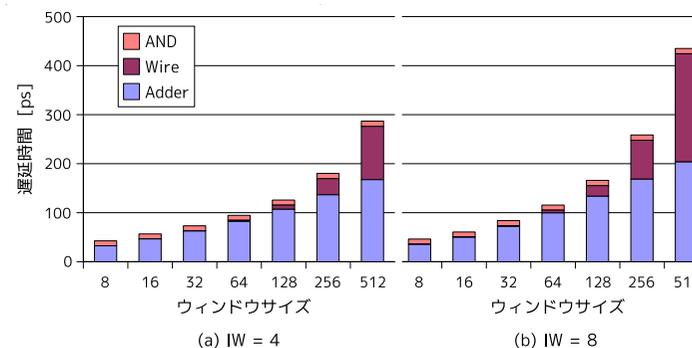


図 18 セレクト論理の遅延時間

5. 評価

種々のウィンドウ・サイズに対し、 $IW = 4, 8$ の場合の発行キューの遅延を、HSPICE を用いたシミュレーションにより評価した。タグのビット幅は、8 に固定した。通常、タグはレジスタ・ファイルのアドレスで与えられるので、256 エントリのレジスタ・ファイルを仮定していることになる。

LSI テクノロジは 32nm を仮定した。HSPICE のトランジスタ・モデルとしては、Arizona State University の Nanoscale Integration and Modeling (NIMO) グループによる予測モデル (PTM: predictive transistor model)⁴⁾ を用いた。長い配線には、中間層メタルが使われると仮定した。その単位長さ当たりの配線抵抗と配線容量の値は、ITRS により予測されたものを用い、それぞれ、 $11.2\Omega/\mu\text{m}$ 、 $0.165\text{fF}/\mu\text{m}$ である²⁾。

5.1 ウェイクアップ論理の遅延時間

ウェイクアップ論理の遅延時間の測定結果を図 17 に示す。ウィンドウ・サイズが大きくなって、タグ・マッチ、マッチ OR、レディ AND はほとんど変化がないが、タグ・ドライブ遅延は大幅に増加している。これは、リピータを挿入し遅延を短縮しているものの、長さに対し 1.2 乗程度にしか短縮できなかったためである。

5.2 セレクト論理の遅延時間

セレクト論理の遅延時間の測定結果を図 18 に示す。ウィンドウ・サイズが小さい間はゲート遅延が支配的であるため、遅延時間は $\log_2 WS$ に比例して伸びている。しかし、ウィンドウ・サイズが非常に大きいところでは、配線遅延が大きな割合を占めている。

5.3 タグ RAM の遅延時間

タグ RAM の遅延時間の測定結果を図 19 に示す。ビット線とセンス・アンプの遅延時間はそれらの合計で示している。一般にある回路の遅延時間は、入力 $V_{DD}/2$ を横切った時刻から出力が $V_{DD}/2$ を横切った時刻の差とするが、本測定でもそのように定義している。しかし、センス・アンプはビット線のわずかな変動で出力が得られるため、 $V_{DD}/2$ を閾値とする遅延測定方法は適当でない。このため、センス・アンプ単独の遅延は示さず、ビット線遅延とセンス・アンプ遅延の和を示す。

図 19 よりわかるように、ワード線遅延は非常に小さい (グラフ上では見えない)。これは、短い配線遅延だけだからである。これに対して、ビット線遅延はウィンドウ・サイズの増加に伴って、大きく増加している。これは、配線遅延によるものである。

5.4 リピータの効果

長い配線に挿入したリピータの効果を示す。図 20 に、リピータを挿入しなかった場合と挿入した場合のタグ・ドライブ遅延を示す。同図からわかるように、ウィンドウ・サイズが大きい時、リピータ挿入により遅延は大きく減少している。配線遅延が大きい $WS \geq 64$ の場合において、 $IW = 4, 8$ でタグ・ドライブ遅延は、リピータを挿入しなければ、それぞれ、 $O(WS^{1.85})$ 、 $O(WS^{1.83})$ で増加するが、リピータを挿入すれば、その増加はそれぞれ、 $O(WS^{1.52})$ 、 $O(WS^{1.21})$ に抑えられる。

なお、 $IW = 4$ 、 $WS = 128$ のとき、リピータ挿入時の方が遅延が長くなっているが、これは次の理由による。4 節で述べたように、リピータ挿入の最適化においては、部品 (この

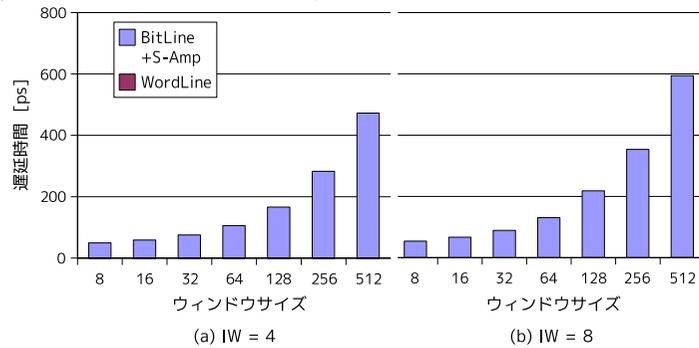


図 19 タグ RAM の遅延時間

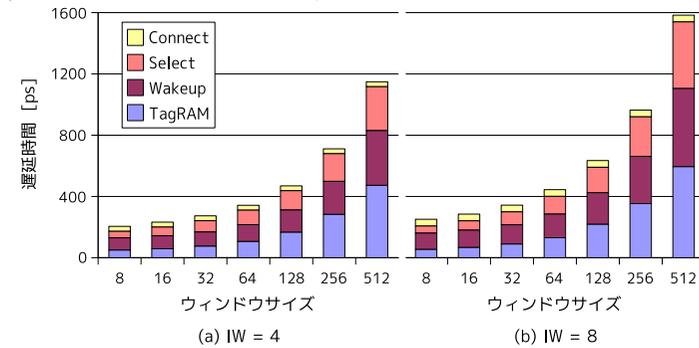


図 21 発行キューの遅延時間

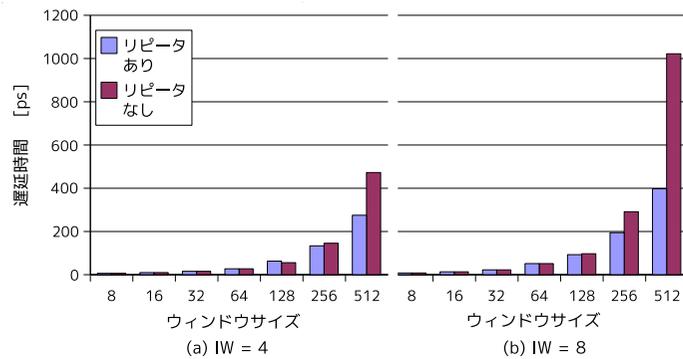


図 20 タグ・ドライブ遅延におけるリピータの効果

場合、ウェイクアップ論理)の遅延で最適化を行っており、配線遅延(この場合タグ・ドライブ遅延)で最適化を行っているわけではない。リピータを挿入するとタグ・ドライブの遅延自体は若干長くなるが、タグ線の立上り時間は短くなり、その結果、それを受けたCAMの遅延が短くなり、ウェイクアップ論理の遅延は短くなっている。

5.5 発行キューの総遅延時間

5.1~5.3節で示した、各部品の遅延時間を合計した発行キュー全体の遅延時間を図21に示す。なお、Connectは3つの部品をつなぐ配線の遅延の和を表している。

測定の範囲内では、どの構成でも、タグRAM、ウェイクアップ論理、セレクト論理の遅延は近い値であることがわかる。WSの増加に応じて遅延は増加するが、その速度は、IW=4のとき、 $O(WS^{0.84})$ 、IW=8のとき、 $O(WS^{0.86})$ ということがわかった。つまり、WSにほぼ比例して増加することがわかった。

6. まとめ

本論文では発行キューの遅延時間評価を行った。今回の測定により、最新のテクノロジーに対応した、発行キューの遅延時間を得ることができた。発行キューのクリティカル・パスをなすウェイクアップ論理、セレクト論理、タグRAMの遅延はほぼ等しく、全遅延は、ウィンドウ・サイズに対してほぼ比例して増加することがわかった。

謝辞 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。

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19500041)による補助のもとで行われたものである。

参考文献

- 1) A. Cristal, O. J. Santana, F. Cazorla, M. Galluzzi, T. Ramirez, M. Pericas, and M. Valero, Kilo-instruction processors: Overcoming the memory wall, *IEEE Micro*, Vol.25, No.3, pp. 48-57, May/June 2005.

- 2) International Technology Roadmap for Semiconductors, 2007 <http://www.itrs.net/>.
- 3) M. Goshima, K. Nishino, T. Kitamura, Y. Nakashima, S. Tomita, and S. Mori, A high-speed dynamic instruction scheduling scheme for superscalar processors, In *Proceedings of the 34th Annual International Symposium on Microarchitecture*, pp. 225–236, December 2001.
- 4) <http://www.eas.asu.edu/~ptm/>.
- 5) P. Michaud and A. Sez nec, Data-flow prescheduling for large instruction windows in out-of-order processors, In *Proceedings of the Seventh International Symposium on High-Performance Computer Architecture*, pp. 27–36, January 2001.
- 6) S. Palacharla, N.P. Jouppi, and J.E. Smith, Quantifying the complexity of superscalar processors, Technical Report CS-TR-1996-1328, University Wisconsin, November 1996.
- 7) P.G. Sassone, I.J. Rupley, E. Brekelbaum, G.H. Loh, and B. Black, Matrix scheduler reloaded, In *Proceedings of the 34th Annual International Symposium on Computer Architecture*, pp. 335–346, June 2007.
- 8) N.H.E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design: A Systems Perspective, second edition*, Addison-Wesley, 1993.
- 9) S.J.E. Willton and N.P. Jouppi, Cacti: an enhanced access and cycle time model for on-chip caches, WRL research report 93/5, Digital WRL Research Laboratory, July 1994.
- 10) 加藤伸幸, 安藤秀樹, 命令発行キューの深いパイプライン化のための投機発行, 先進的計算基盤システムシンポジウム (SAC SIS 2009), pp. 319–326, 2009年5月.
- 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年3月.

付 録

A.1 リピータ挿入による配線遅延の変化

長い配線を $n (= 1, 2, \dots)$ 等分し, リピータを $n - 1$ 個挿入することを考える. このとき, 配線遅延 τ_l は, n の変化に対して, ただ 1 つの極小値を持つことを示す.

単位長さ当たりの配線抵抗と配線容量を, それぞれ, r, c としたとき, 長さ l の配線のみの遅延は,

$$\tau_l = 0.5crl^2$$

で与えられる⁶⁾. ここで, 係数 0.5 は, RC 分布定数回路の遅延の 1 次近似における係数である.

今, 配線の始点にドライバを配置し, そのドライバとリピータの回路およびそのトランジ

スタ・サイズを同一とする. そして, それら自体の遅延を単純に定数 τ_r と仮定する. 配線にリピータを等間隔に $n - 1$ 個挿入し, ドライバとリピータの遅延を含めた配線遅延は,

$$\begin{aligned}\tau_l &= 0.5cr \left(\frac{l}{n}\right)^2 \cdot n + \tau_r n \\ &= \frac{0.5crl^2}{n} + \tau_r n\end{aligned}$$

となる. よって, 配線遅延 τ_l は, n の変化に対して, ただ 1 つの極小値を持つ.