

## 高性能スーパー・スカラプロセッサ用浮動小数点演算器

福久 浩人, 井出 進博, 近藤 勝久, 吉田 尊, 高野 裕之, 佐藤 寿倫,  
永松 正人\*, 森 順治\*, 山崎 到\*, 上野 喜代治\*

東芝 ULSI 研究所 半導体技術研究所\*

〒210 神奈川県川崎市幸区小向東芝町 1

あらまし 高性能スーパー・スカラプロセッサ用浮動小数点演算器を開発した。本演算器は、 $0.5\mu\text{m}$  CMOS、3層アルミのプロセスを用い、IEEE754規格に準拠している。ALU、乗除算器、レジスタファイルから構成されており、ALUと乗除算器は独立に動作可能である。  
各モジュール間にバイパスを設ける事によりデータ依存のある命令のレイテンシを削減している。また、2組の単精度データを64ビットアーキテクチャの演算器で並行に実行する、単精度倍速演算モードを用いる事により、演算能力を高めている。さらに、アンダフロー、オーバフローの例外予測を行う事により例外処理を円滑に行う事ができる。

和文キーワード 浮動小数点演算器 ALU 乗除算器 バイパス 例外予測

## Floating-point Processing Unit for High Performance Superscalar Processor

Hiroto Fukuhisa, Nobuhiro Ide, Yoshihisa Kondo, Takeshi Yoshida, Hiroyuki Takano, Toshinori Sato, Masato Nagamat<sup>†</sup>, Junji Mori<sup>‡</sup>, Itaru Yamazaki<sup>‡</sup>, Kiyoji Ueno<sup>\*</sup>

ULSI Reseach Center, Semiconductor Device Engineering Laboratory<sup>†</sup>, Toshiba corp.

1.Komukai Toshiba-cho, Saiwai-ku Kawasaki, 210. JAPAN

### Abstract

A CMOS pipelined floating-point processing unit(FPU) for superscalar processors is described. It uses triple-metal 0.5micron CMOS technology. It consists of an ALU, a multiply/divide unit(MDU), and a register file. They can function independently. The register file can load and store two 64-bit data in each cycle.

The ALU, the MDU, and the register file are closely connected with feed-forward loading paths to reduce latency. The FPU has a new execution mode, called "twin-single". It sets two single-precisions in 64-bit data to process two sets of data in each cycle.

英文 key words FPU ALU Multiply/Divide Unit Feed-forward Loading Exception Prediction

## 1. 概要

高性能スーパー・スカラ・プロセッサ用浮動小数点演算器を開発した。この演算器は、64ビットアーキテクチャのALU、乗除算器（以下、MDU）およびレジスタファイルを持ち、IEEE754規格<sup>[1]</sup>に準拠した演算を実行する。

本演算器は高速化のための3つの特徴を持つ。第1にALU、MDU、レジスタファイルの各モジュールおよびFPU外部バスとの間にバイパスを設け、ロード・ストア命令および演算器間のデータ転送の円滑化を考えている。第2に単精度倍速演算モードを備え、64ビット演算器で同時に2組の単精度数の演算を実行する。第3に例外予測を行うことにより、円滑に例外処理を行っている。

本演算器は表1に示すような性能を持つ。プロセスは0.5 $\mu$ m CMOSで3層アルミを用いている。トランジスタ数は約290K、回路面積はデータパス部で61mm<sup>2</sup>である。

本稿では、まず本演算器のアーキテクチャを示し、バイパス、単精度倍速演算および例外予測について述べる。

## 2. 演算器のアーキテクチャ

本演算器の構成を図1に示す。64ビットアーキテクチャのALU、MDUおよびレジスタファイルの3つのブロックにより構成される。それぞれの演算器は独立に動作することができる。動作周波数は80MHzであり、ピーク性能は160MFLOPSである。レジスタファイルは2つの64ビットデータを1クロックで入出力可能である。

表2に本演算器で実行可能な命令を示す。比較命令はレイテンシが1クロック、絶対値化、符号反転、スワップ、move命令は2クロック、除算は単精度で10クロック、倍精度で17クロック、その他の演算が3クロックで動作する。なお、スワップ命令は本演算器特有のもので、後述する単精度倍速演算モードにおいて、上位32ビットデータと下位32ビットデータの入れ替えを行う。以下、各演算器について述べる。

ALUの構成を図2に示す。演算実行は3ステージに分け、1クロック毎に演算結果を出力する。加減算、数値変換命令は3クロック、比較命令は1クロック、絶対値化、符号反転等は2クロックで動作する。また、数値変換命令は単精度、倍精度、整数間のすべての変換命令が実行可能である。

各ステージ内での処理内容は次の通りである。まず、第1ステージで加減算命令の桁合わせ、比較命令の実行および非正規化数等の検出を行う。第2ステージでは加減算命令の仮数部の加減算、IEEE754規格に基づく丸め判定および先行0チェックを行う。第3ステージでは、丸めと正規化を行う。演算の性質上、丸めと正規化は依存性がないため、並行して行うことができる。すなわち、この演算器は丸めと正規化を並行して実行している。これにより、加減算の結果を高速に得ることができる。

ALUのクリティカルパスは第1ステージの（指数部比較）→（桁合わせシフト）の箇所である。指数部比較回路に11ビット絶対値加算器を用いることにより、クリティカルパスおよび回路面積を小さくすることができた。

MDUの乗算器および除算器は、それぞれ仮数部のデータパスを第1、第2ステージで独立に持つ。指数部および仮数部第3ステージは共用している。オペランドの入力をクロック毎に乗算器と除算器のいずれかに切り換えることにより、それぞれを独立に動作させることが可能である。このため第3ステージにおいて乗算と除算の結果がコンフリクトを起こさないように制御する必要がある。本演算器では除算結果が優先的に第3ステージで処理される。これは除算は乗算よりレイテンシが多く、演算結果が同時に第2ステージから出力された場合は除算の方が乗算より前に命令が発行されているためである。

乗算器は2次のBoothのアルゴリズム<sup>[2]</sup>、Wallaceツリー<sup>[3]</sup>を用いている。Wallaceツリーは4-2コンパクトで構成した。Wallaceツリーの乗算器アレイは55 $\times$ 55ビット構成で倍精度演算対応<sup>[4]</sup>となっている。55ビットの内訳は符号1、仮数部52、Boothの符号ビット1、シフトビット1であ

る。演算は3クロックで行う。第1ステージでは指数部の加算による中間値の導出を行う。また、仮数部は部分積の導出、Wallaceツリーで部分的な加算を行う。第2ステージで仮数部の最終加算を行い、仮数部の中間結果を得る。第3ステージで仮数部の正規化およびIEEE754規格に基づく丸め、正規化を実行、さらに指数部の調節を行う。Wallaceツリーの加算器アレイのレイアウト構成は4-2コンパクト間を3層アルミで結合することにより、 $3.2 \times 3.7 \text{mm}^2$ の大きさで実現できた。

一方、除算器は4を基数とする収束型高基数除算アルゴリズム(非回復型除算) [5],[6]を用いている。演算は3つのステージに分けられるが、第1および第2ステージを繰り返すため、単精度で10クロック、倍精度で17クロックのレイテンシとなる。第1ステージでは、仮数部は2段重ねのキャリーセーブ加算器を用いて1クロックで4ビットの商を求めると同時に部分剰余を演算する。これにより、演算速度が向上する。しかし、商および部分剰余を冗長2進フォーマットのままにして、次のディジットを求めることは困難である。本演算器では冗長性を用いて部分剰余と除数の上位9ビットを参照し、次の商のディジットを決定している。9ビットのキャリープロパゲート加算器で部分剰余の9ビットと定数の減算を行う。これと、除数の1/2もしくは3/2の値と比較して、次のディジットを決定する。本演算器では、部分剰余の導出と商のディジットの決定を同一ステージ内で並行して行う。第2ステージで部分剰余と商をキャリープロパゲート加算器で基数4から2進数に変換する。第3ステージでは丸めを行う。

レジスタファイルは64ビット×32ワードで構成した。1セルあたりリードポートを6、ライトポートを4つ持つ。リードポートは、後述するバイパスの各系統に該当し、ライトポートは2つが外部バスからのデータ入力用であり、残りの2つがALU、MDUからの演算結果を書き込むものである。リードは1サイクルで同一または異なるワードにアクセス可能である。ライトは同一サイクルに異なるワードに書き込みが可能である。この構成により、ALUとMDUが独立に動作することが可

能となっている。ライト動作はクロックと同期しているが、リード動作は非同期である。

### 3. バイパス

従来の演算器では、ALU、MDU間のデータ転送およびキャッシュメモリと浮動小数点演算器内部とのデータ転送はレジスタファイルを介して行う必要がある。その場合、演算速度がデータ転送に律束される問題がある。この問題を解決する方法として、既存の浮動小数点演算器ではデータ転送がボトルネックとなる演算器の入出力にバイパスを設けている。

さらに演算処理中には、前の命令で得た演算結果を次の命令で利用する場合が生じる。本演算器のように2つの演算器を並列に動作させて処理を行う場合、問題となるのが2命令間のデータの依存関係である。例えばMDUがALUの実行結果を用いる場合、ALUの演算結果が出力されるまで、その命令を実行できない。このような演算性能の低下は実行時のパイプラインの段数が多く、また実行時間が長いほど顕著である。

2つの演算器の動作における依存関係から生じる演算性能の低下は、演算器間をバイパスで結合することにより緩和できる。本演算器は既存の演算器が備えている演算器の入出力に関する2本のバイパスに加え、ALU、MDU、レジスタファイルといった演算器内部間のデータ転送を高速に行うための4本のバイパスを備えている。

図3にバイパスの構成を示す。バイパスは系統中にあるトライステート・バッファの開閉により行う。トライステート・バッファは各演算器の第3およびライトステージに備えている。特にALUはレイテンシが2で処理可能な命令があるため、第2ステージにも備えている。それぞれの系統は、外部バスへの出力を行う系統、ALUへの入力を行う系統、MDUへの入力を行う系統を、それぞれソースオペランド、ターゲットオペランドの2系統づつ備えている。これらの系統はレジスタファイルの6つのリードポートに対応している。

各バイパスは独立に動作が可能である。また、後述する単精度倍速演算モードに対応して、上位、

下位32ビットは独立に動作可能である。バイパスは演算結果を出力する各ステージの後半で行う。そのため演算結果を出力するステージの実質的な演算実行時間は、バイパスの遅延を考慮して短くしている。

バイパスされる演算結果は常に新しくなくてはならない。本演算器はALUとMDUから同時に演算結果を出力できる。さらに、同一演算器内でも各ステージから同時に演算結果を出力できる。このように、同時に出力されたデータの内、どのデータを最新のものとしてバイパスするかが問題となる。そこで、本演算器ではデータ供給の優先順位を定めている。

実際のレイアウト上では、トライステートバッファ間をつなぐバイパス配線は第3アルミを用いて、配線によるディレイと回路面積を最小限に抑えている。また、モジュール間の配線が長くなる系統には、遅延による影響を緩和するために駆動力を高めたトライステートバッファを設けた。バイパスを用いることにより、データ依存による命令開始のレイテンシを減らすことが可能になった。

#### 4. 単精度倍速演算

浮動小数点演算器に要求される能力は速度と演算結果の精度である。実際のアプリケーションでは、科学技術計算のように演算結果の精度を重視したものと、リアルタイム制御計算のように演算速度を重視したものがある。

本演算器で用いている単精度倍速演算モードは演算速度を向上させることを目的としている。単精度データを64ビットアーキテクチャで表現した場合、上位32ビットは未使用の状態となる。そこで、図4-aに示すように、未使用の上位ビットを用いて、64ビット中に2つの単精度データを表現することができる。このデータを64ビットアーキテクチャの演算器で処理すれば、1サイクルに2組の単精度データを演算することができる。これが単精度倍速演算モードである。

本演算器ではALUおよびMDUに単精度倍速モードを備えている。図4-bに、その構成を示す。演算は加減算、乗除算、絶対値化および符号反転

を行うことができる。また、演算は通常の場合と同じくALUとMDUは独立に実行することができるが、同一演算器内では、2つの32ビットデータに対して1つの演算命令を実行する。例えばALU内で上位32ビットを加算、下位32ビットを減算といった演算は実行しない。1つの演算器内で独立した2つの命令を実行できるようにするには各演算命令の制御回路を2つ設けなければならない。さらにレイテンシの異なる演算命令では、制御が著しく複雑になるためである。また、丸めモードはIEEE754規格のゼロ方向丸めのみをサポートしている。

数値の型はIEEE754規格の単精度と同じ、符号部1ビット、指数部8ビット、仮数部23ビットである。本演算器は倍精度対応のため、仮数部ではデータパスのビット幅は充分である。ただし、上位仮数部と下位仮数部の分割を行うスイッチが必要である。例えば加算器について考えると、下位32ビットの加算結果で生じるキャリーが上位32ビットに伝搬しないようにする必要がある。また指数部では通常1組のデータを処理する指数部比較、指数部正規化回路しか備えていない。そのため、もう1組のデータを処理する単精度データ専用の回路が必要である。実際のレイアウトパターンでは全体の4%以下の回路面積の増加で実現できた。単精度倍速演算モードでは最大4つの演算器が並列に動作できるため、動作速度は最大320MFLOPSになる。

#### 5. 例外予測

本演算器では、2つの演算器が同時にパイプライン処理を行うため、常に複数の命令が実行されている。ある命令で例外が発生した場合、演算器で実行中の命令のうち、例外が発生した命令より前に処理を始めた命令は最後まで処理を行い、後で処理を始めた命令は無効として例外を処理する。そして、例外が発生した命令の処理が終了すると、その命令と同時に発行された命令から演算を再開する。

さらに、本演算器では命令の種類によりレイテンシが異なる。そのため、演算処理の追い越しを

生じる場合がある。本演算器ではこの追い越しを許している。ここで、下記の命令列を考える。

# 1 FDIV fr0, fr1, fr2 (fr2←fr1÷fr2)

# 2 FADD fr3, fr4, fr0 (fr0←fr3+fr4)

この場合、除算命令の入力と加算命令の書き込みアドレスが共に  $f r 0$  と一致している。図 5 - a に示すように除算は加算より 1 クロック先に実行するが、レイテンシの違いから加算命令の方が先に演算を終了する。ここで除算に例外が発生しなければ、加算結果を除算命令の入力値のあったアドレス  $f r 0$  に直接書き込む。しかし、除算の最終ステージで例外が発生した場合、本来除算の入力となる  $f r 0$  の値が、すでに加算結果に書き変わっているため再実行不可能となる。これを避けるためには、図 5 - b に示すように除算命令で例外が発生する可能性がある場合に、加算命令を待ち合わせておき、除算命令が例外が発生しないと判定できたときに加算結果を書き込む必要がある [7]。問題となるのは、この場合の加算命令のように待ち合わせ制御を行うことによって除算命令に例外が発生しない場合、演算器の性能を低下させることである。したがって本演算器では後続命令の待ち合わせによる性能低下を抑えるために、例外の発生を実行ステージの早い段階で予測し、例外発生が起きるがどうか確実に判定できないときのみ、例外予測信号を発生させるメカニズムを備えている。

IEEE754規格では、無効演算例外、ゼロ除算例外、アンダフロー例外、オーバフロー例外および精度落ち例外を規定している。このうち、無効演算例外、ゼロ除算例外は入力されるオペランドから容易に判断できるため、例外予測は必要ない。その他の例外は演算終了間近でないと検出できない。本演算器はALUでは、加減算のアンダフローとオーバフローだけでなく、単精度、倍精度から32ビット整数への変換に対するオーバフローの予測も行う。MDUでは、乗除算のアンダフローとオーバフローの予測を行う。精度落ち例外についてはオーバフローによるものと、丸め処理によるものがある。前者は、オーバフローの例外予測で可能である。後者は丸めが行われる場合を検出す

ればよい。しかし丸めは頻繁に行われるため、ほとんどの場合が例外予測の対象となり、演算器の性能をかえって低下させることになる。そのため、本演算器では精度落ち例外の予測は行っていない。

IEEE754規格の場合、指数部の表現可能範囲がバイアス値を与えた値で倍精度の場合1から2046、単精度の場合1から254である。最終的な指数部の演算結果は、指数部比較を行った中間値と正規化に要する仮数部のシフト数で決まる。従って、正規化に要する最大シフト数と表現可能な範囲、および指数部の中間値との関係から例外発生を予測することが可能である。

ここでは加減算に関するアンダフロー例外予測について述べる。加減算で仮数部の正規化に要する最大シフト数は左シフトの場合、倍精度で53ビット、単精度で24ビットである。アンダフローは左シフトが生じた場合に、指数部を正規化する際に検出される。指数部の正規化は中間値から左シフト数分引くことで行う。したがって、指数部の中間値が倍精度の場合54以上、単精度の場合25以上であるなら、アンダフローが生じないと判断する。さらに、以下に示した桁合わせ後の演算を考える。

H	LSB G R St
1 . 0 0 0 0 0 0 0 0 0 0 . . . . . 0 0 0 0	
- 0 . 1 1 1 1 1 1 1 1 1 1 . . . . . 1 1 1 1	
-----	
0 . 0 0 0 0 0 0 0 0 0 0 . . . . . 0 0 0 1	

この場合に正規化シフトが最大となることがわかる。減数について注目すると桁合わせに1ビットシフトを行っており、それ以上の桁合わせシフトでは正規化シフトは高々1ビットである。すなわち、桁合わせに必要なシフト数が1より大きければアンダフローが生じないと判断できる。

オーバフローはアンダフローの例外予測と同様に指数部の中間値と右シフト数の最大値の和が、指数部の表現可能範囲であればよい。したがって指数部の中間値が倍精度で2045以下、単精度で

253以下の場合、オーバフローが生じないと判断する。

例外予測のハードウェアは第1ステージに設けた。これは第1ステージでは加減算に対する指数部比較だけではなく、各種数値フォーマット変換に必要となる指数部の中間値も求めるためである。得られた中間値と先の定数を比較することにより例外予測信号を制御部へ出力する。ハードウェアとしては、数値比較回路と定数セレクタで実現できる。

## 6. まとめ

高性能浮動小数点演算器を開発した。演算器は64ビットアーキテクチャで、IEEE754規格に準拠している。また、内部はALU、MDU、レジスタファイルを備え、各演算器は80MHzで独立に動作可能である。演算速度を高速化するため、演算器間にフルバイパスを装備、単精度浮動小数点数に対する演算を64ビット幅の演算器で2つ並行に処理する単精度倍速演算モードを実装、演算実行結果から生じる例外を指数部の中間値から予測する例外予測を実装した。

## 7. 参考文献

- [1] ANSI/IEEE Standard 754-1985 for Binary Floating-Point Arithmetic, IEEE Computer Society Press, Los Alamitos, California, 1985.
- [2] A.D.Booth: "A Signed Binary Multiplication Technique," Quarterly J.Mechanics and Applied Mathematics, Vol. 4, Part 2, 1951
- [3] C.S.Wallace: "A suggestion for fast multipliers," IEEE Trans. Electron. Computer, vol. EC-13, pp. 14-17, Feb. 1964.
- [4] J.Mori et al.: "A 10-ns 54 × 54-b Parallel Structured Full Array Multiplier with 0.5- $\mu$  m CMOS Technology," IEEE Journal of Solid-State Circuits, Vol. 26, No. 4, April 1991.
- [5] J.E.Robertson: "A New Class of Digital Division Methods," IEEE Trans. Computer, Vol. C-7, pp. 218-222, Sep. 1958.

[6] D.E.Atkins: "Higher-Radix Division Using Estimates of the Divisor and Partial Remainders," IEEE Trans. Computer, vol. C-17, No. 10, pp. 925-934, Oct. 1968.

[7] 吉田 尊 他: "スーパースカラ向け高性能FPUの制御方式", 信学報VLD 91-142, ICD 91-228.

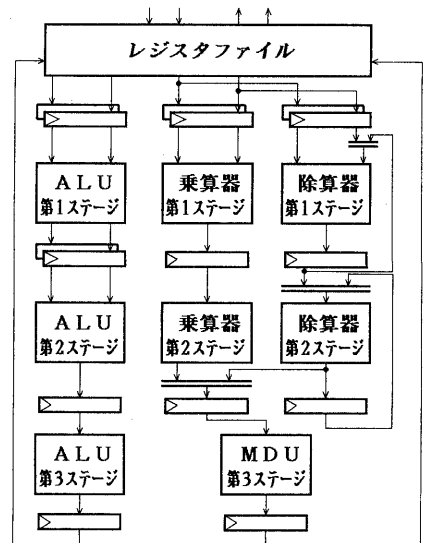


図1 FPUの構成

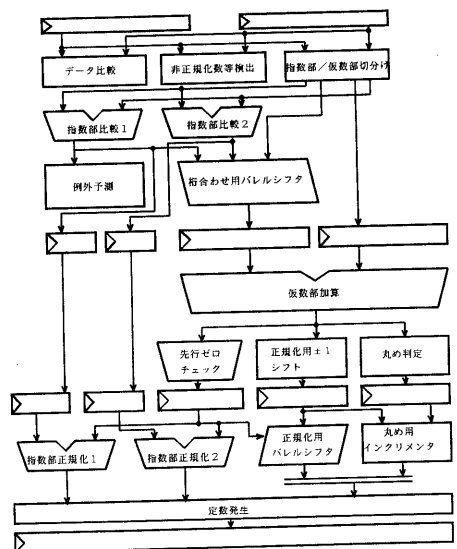


図2 ALUの構成

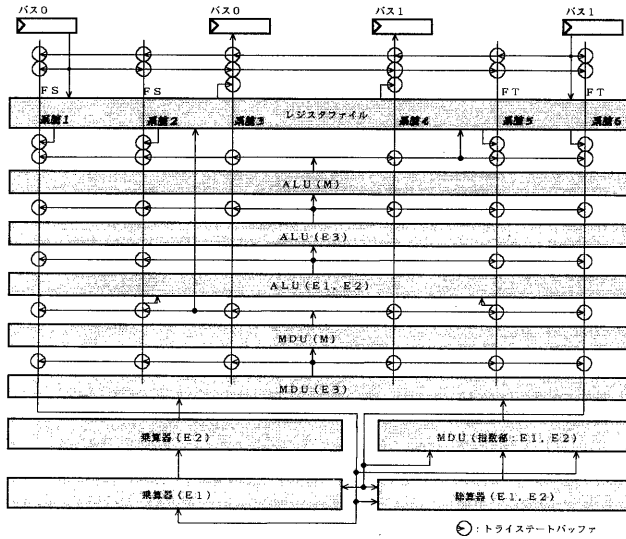


図3 バイパスの構成

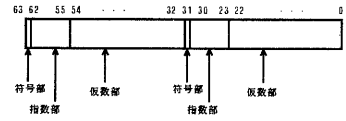


図4-a 単精度倍速演算数値形式

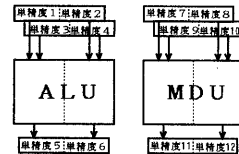


図4-b 単精度倍速演算の演算器構成

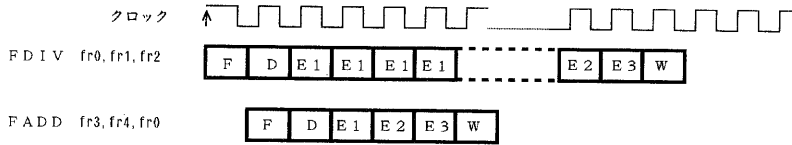


図5-a 例外予測を用いていない演算制御

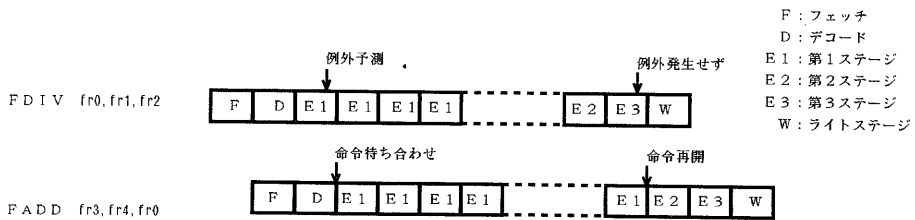


図5-b 例外予測を用いた演算制御 (例外発生しない場合)

面積	61mm <sup>2</sup>
プロセス	0.5μm CMOS 3層アルミ
電源電圧	3.3V
動作周波数	80MHz
ピーク性能	320MFLOPS (単精度倍速モード) 160MFLOPS (倍精度, 単精度)

表1 本演算器の性能

演算器	命令	レイテンシ	スループット
ALU	加減算	3	1
	倍精度変換	3	1
	単精度変換	3	1
	整数変換	3	1
	比較	1	1
	絶対値化	2	1
	符号反転	2	1
	スワップ	2	1
	Move	2	1
MDU	乗算	3	1
	除算	17 (倍精度)	14
		10 (単精度)	7

表2 本演算器の命令レイテンシおよびスループット