

音響 FDTD 法演算支援 LSI の設計

伊藤 大地[†] 小林良太郎[†] 島田俊夫[†]

[†] 名古屋大学大学院工学研究科
〒464-8603 名古屋市千種区不老町

E-mail: †{ito,kobayasi}@shimada.nuee.nagoya-u.ac.jp, ††shimada@nuee.nagoya-u.ac.jp

あらまし FDTD(Finite Difference Time Domain) 法は、様々な構造物を扱えること、さらに他の数値解析法に比べて簡便な手続きで解が得られることから、最近音場解析にも利用されつつある。しかし、FDTD 法は解析領域が大きくなると計算時間が膨大になる問題がある。本論文では、音場解析に用いる FDTD 法の演算時間の短縮を目指して専用 LSI を設計し、FPGA を用いてその有効性を調べた。

キーワード 音響 FDTD 法, 専用 LSI, 高速化

An LSI design to support Sound Finite Difference Time Domain Method

Daichi ITO[†], Ryotaro KOBAYASHI[†], and Toshio SHIMADA[†]

[†] Graduate School of Engineering, Nagoya University
Furo-Cho, Chikusa-Ku, Nagoya-City, Aichi, 464-8603 Japan

E-mail: †{ito,kobayasi}@shimada.nuee.nagoya-u.ac.jp, ††shimada@nuee.nagoya-u.ac.jp

Abstract Recently the FDTD method is used for sound field analysis because it can treat various structures and the procedure to obtain solutions is. However, the FDTD method has a problem that it takes vary long time for execution as the scale of acoustic field becomes larger. This paper proposes a design of a special purpose processor to reduce the execution time for the FDTD method, and show some evaluation results by using FPGA.

Key words sound-FDTD method

1. はじめに

FDTD(Finite Difference Time Domain) 法は、K.S.Yee により電磁波解析を行う手法として提案されたものである。マクスウェルの方程式を時間、空間で差分化(電界節点と磁界節点が時間的にも空間的にも半ステップ分ずれた差分化となっている)し、解析空間の電磁界を時間的に交互に更新し、出力点の時間応答を得る方法である。

FDTD 法はアルゴリズムが簡単であること、優れた精度を持つこと、複雑な物質の解析や材料定数の異なる物質の解析にも適していることなどが知られているが、その計算量の多さから当初はあまり用いられていなかった。しかし、近年のコンピュータの高速化・メモリの大容量化により非

常にポピュラーな方法になっている。

多くの利点がある FDTD 法だが、コンピュータの発達した現在でも、解析領域の大きい 3 次元解析を行う場合は長時間の計算時間が必要である。そこで、過去様々な方法でその計算時間の短縮化が図られてきた。スーパーコンピュータを利用し並列計算すれば高速に計算できるが、コストがかかるため、最近ではパーソナルコンピュータを複数利用し、並列計算する方法も行われている。しかし、この方法は FDTD 法の並列化のために本来簡単であるアルゴリズムが複雑になるという問題があり、誰もが簡単に導入できるものではない。

また FDTD 法は、様々な構造物を扱えること、さらに他の数値解析法に比べて簡便な手続きで解が得られることか

ら、最近音場解析にも利用されつつある。しかし、この音響 FDTD 法を音場解析に適用している事例はまだ少なく、並列化のノウハウもまだほとんど無い。

本研究では、プログラムの並列化よりは少ない手間で、かつ低コストに音響 FDTD 法の演算時間を短縮することを旨とし、専用 LSI を設計し FPGA でその有効性を調べた。

本稿では、2 章で音響 FDTD 法の概要を述べる。3 章では設計した演算器の概要を述べ、4 章で評価する。最後に 5 章でまとめ、6 章で今後の課題を述べる。

2. FDTD 法

FDTD 法は電磁波解析を行う手法として 1966 年に K.S.Yee により提案されたものである [7]。この手法では空間領域および時間領域に差分表現を用いて、波動関数の時間応答を計算する。特に電磁界では電界と磁界の対称性を用いることにより、両者を交互に更新することで計算の精度を上げる工夫がなされている。この手法の特徴としては、

- 複雑な微分・積分などの計算を行う必要がない
- 時間領域で計算するため非線形問題にも適用可能
- 差分に起因する格子形状の制約がある
- 格子間隔を無限に小さくすれば厳密解に近づく
- CPU 速度と多くのメモリを要する

などが挙げられる。この FDTD 法は近年、コンピュータの進化に伴い電磁波解析のみならず音波伝搬解析の手法としても注目されている。

FDTD 法を用いて音場解析をする方法は解析対象により 2 通りに分類することができる。流体内の音場解析に用いる方法と弾性体中の音場解析に用いる方法である。前者を「音響 FDTD 法」、後者を「弾性 FDTD 法」と呼ぶが、後者は横波を考慮しているという点で前者と異なる。

本研究では、音響 FDTD 法を用いた。その理由は、取り扱う解析モデル中に吸収境界面が存在するとき、弾性 FDTD 法による計算では安定化が困難であるからである。特に吸収境界層で媒質の変化が著しいモデルは、計算の安定化がさらに難しくなる。

以下、本研究で用いた音響 FDTD 法の計算方法についての概略を述べる。

2.1 音響 FDTD 法

音響 FDTD 法の支配方程式は連続の式 (equation of continuity) と運動方程式 (equation of motion) を音圧と粒子速度で表した式であり、3 次元直交座標系において以下の 4 式である。

$$-\frac{1}{\kappa} \frac{\partial p}{\partial t} = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \quad (1)$$

$$-\rho \frac{\partial v_x}{\partial t} = \frac{\partial p}{\partial x} \quad (2)$$

$$-\rho \frac{\partial v_y}{\partial t} = \frac{\partial p}{\partial y} \quad (3)$$

$$-\rho \frac{\partial v_z}{\partial t} = \frac{\partial p}{\partial z} \quad (4)$$

ここで p は音圧 (sound pressure)、 v_i は粒子速度 (particle velocity)、 κ は体積弾性率 (volume elasticity)、 ρ は密度 (density) である。

(1)(2)(3)(4) 式は無減衰の方程式であるが、媒質に減衰を考慮し (2)(3)(4) 式に吸収減衰項を追加すると、

$$-\rho \frac{\partial v_x}{\partial t} = \frac{\partial p}{\partial x} + \eta v_x \quad (5)$$

$$-\rho \frac{\partial v_y}{\partial t} = \frac{\partial p}{\partial y} + \eta v_y \quad (6)$$

$$-\rho \frac{\partial v_z}{\partial t} = \frac{\partial p}{\partial z} + \eta v_z \quad (7)$$

となる。ここで η は吸収減衰係数 (attenuation coefficient) である。

2.2 支配方程式の差分化

FDTD 法における支配方程式の差分化は空間領域、時間領域のそれぞれに対して中心差分法により差分化される。座標位置が (i, j, k) 、時間ステップ n における関数 f を $f^n(i, j, k)$ と定義すると、空間領域・時間領域それぞれにおける差分化は以下ようになる。

$$\frac{\partial f^n(i, j, k)}{\partial x} = \frac{f^n(i + \frac{1}{2}, j, k) - f^n(i - \frac{1}{2}, j, k)}{\Delta x} \quad (8)$$

$$\frac{\partial f^n(i, j, k)}{\partial t} = \frac{f^{n+\frac{1}{2}}(i, j, k) - f^{n-\frac{1}{2}}(i, j, k)}{\Delta t} \quad (9)$$

図 1 に音響 FDTD の音圧及び粒子速度の空間配置図を示す。

これらは FDTD 法の基本的配置であり、このように解析領域は直方体格子に分割していることを示している。また、時間領域においては図 2 のように離散化し、音圧と粒子速度を交互に更新することにより場の計算を行っている。

媒質での減衰を考慮にいたれた音響 FDTD 法の基本式 (1)(5) (6)(7) を、先ほど定義した方法で空間領域および時間領域で中心差分により差分化し、まとめると、

$$p^{n+\frac{1}{2}}(i, j, k) = p^n(i, j, k) - \frac{\kappa \cdot \Delta \delta t}{\Delta x} \left\{ \begin{array}{l} v_x^{n+\frac{1}{2}}(i + \frac{1}{2}, j, k) - v_x^{n+\frac{1}{2}}(i - \frac{1}{2}, j, k) \\ v_y^{n+\frac{1}{2}}(i, j + \frac{1}{2}, k) - v_y^{n+\frac{1}{2}}(i, j - \frac{1}{2}, k) \\ v_z^{n+\frac{1}{2}}(i, j, k + \frac{1}{2}) - v_z^{n+\frac{1}{2}}(i, j, k - \frac{1}{2}) \end{array} \right\} \quad (10)$$

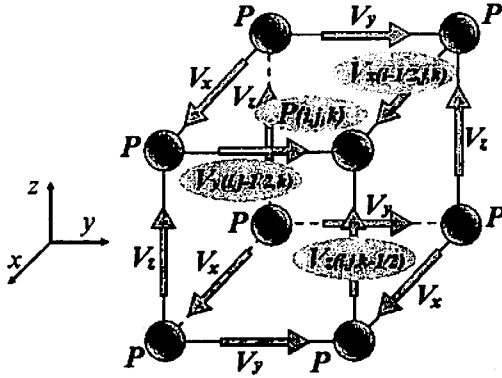


図1 音響 FDTD の音圧及び粒子速度の空間配置図

Fig.1 Positions of the sound pressure and the particle velocity components for the sound-FDTD.

● p : sound pressure
▲ v : particle velocity

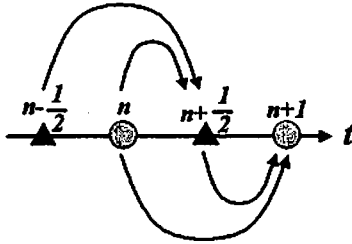


図2 音響 FDTD の時間領域の離散化と更新の様子

Fig.2 Leapfrog for the time derivatives for sound-FDTD.

$$V_x^{n+\frac{1}{2}}(i-\frac{1}{2}, j, k) = \frac{(1 - \frac{\eta \Delta t}{2\rho})}{(1 + \frac{\eta \Delta t}{2\rho})} \cdot V_x^{n-\frac{1}{2}}(i-\frac{1}{2}, j, k) - \frac{\Delta t}{\rho \cdot \Delta x (1 + \frac{\eta \Delta t}{2\rho})} [p^n(i, j, k) - p^n(i-1, j, k)] \quad (11)$$

$$V_y^{n+\frac{1}{2}}(i, j-\frac{1}{2}, k) = \frac{(1 - \frac{\eta \Delta t}{2\rho})}{(1 + \frac{\eta \Delta t}{2\rho})} \cdot V_y^{n-\frac{1}{2}}(i, j-\frac{1}{2}, k) - \frac{\Delta t}{\rho \cdot \Delta x (1 + \frac{\eta \Delta t}{2\rho})} [p^n(i, j, k) - p^n(i, j-1, k)] \quad (12)$$

$$V_z^{n+\frac{1}{2}}(i, j, k-\frac{1}{2}) = \frac{(1 - \frac{\eta \Delta t}{2\rho})}{(1 + \frac{\eta \Delta t}{2\rho})} \cdot V_z^{n-\frac{1}{2}}(i, j, k-\frac{1}{2}) - \frac{\Delta t}{\rho \cdot \Delta x (1 + \frac{\eta \Delta t}{2\rho})} [p^n(i, j, k) - p^n(i, j, k-1)] \quad (13)$$

となる。ここで、 x, y, z 方向の空間の刻み幅は $\Delta x, \Delta y, \Delta z$ 、時間刻み幅は Δt であり、 $\Delta x = \Delta y = \Delta z$ とした。

これらの (10)(11)(12)(13) 式をプログラム上で扱えるようにまとめると、

$$P[i, j, k] = P[i, j, k]$$

$$- C_p \begin{Bmatrix} V_x[i+1, j, k] - V_x[i, j, k] \\ V_y[i, j+1, k] - V_y[i, j, k] \\ V_z[i, j, k+1] - V_z[i, j, k] \end{Bmatrix} \quad (14)$$

$$V_x[i, j, k] = C_{v0}[i, j, k] \times V_x[i, j, k] - C_{v1}[i, j, k] \times (P[i, j, k] - P[i-1, j, k]) \quad (15)$$

$$V_y[i, j, k] = C_{v0}[i, j, k] \times V_y[i, j, k] - C_{v1}[i, j, k] \times (P[i, j, k] - P[i, j-1, k]) \quad (16)$$

$$V_z[i, j, k] = C_{v0}[i, j, k] \times V_z[i, j, k] - C_{v1}[i, j, k] \times (P[i, j, k] - P[i, j, k-1]) \quad (17)$$

となる。ここで、 P は音圧、 V は粒子速度、 C は定数である。今後、本稿ではこの式を使って説明する。

2.3 解の正確性と安定性 [?]

FDTD 法は先ほど述べたように空間領域および時間領域での差分化により計算するため、差分化するセルの大きさや時間によって不安定であったり正確さに欠けたりする。以下に今回用いた安定条件について述べる。

2.3.1 時間刻みに関する安定条件

時間に関して安定となるための条件は 1 ステップ Δt 、1 セルサイズ: $\Delta x, \Delta y, \Delta z$ 、伝搬速度: v とすると、

$$v \cdot \Delta t \leq \frac{1}{\sqrt{(\frac{1}{\Delta x})^2 + (\frac{1}{\Delta y})^2 + (\frac{1}{\Delta z})^2}} \quad (18)$$

である。この条件は Courant の安定条件とも呼ばれ、きわめて厳しく、わずかでも満足しなければ時間とともに発散する。

2.3.2 セルサイズに関する安定条件

グリッド分散誤差と呼ばれる差分近似のために生じる数値誤差を小さくする為には波長を λ とすると、

$$\left(\frac{\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}}{\lambda} \right)^2 \ll 1 \quad (19)$$

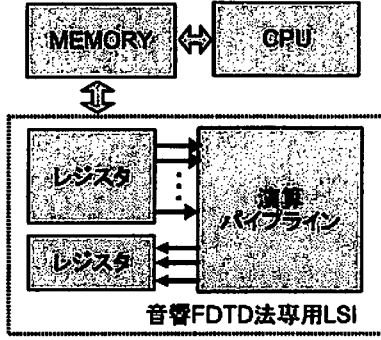


図3 システムの構成図
Fig. 3 Constitution of a system.

とする必要がある。セルサイズは波長に対して1/10あるいは1/20以下にする必要があるといわれる。

2.4 吸収境界条件

無限空間における音場など、いわゆる開放領域の問題を扱う場合には、解析領域を仮想的な境界で閉じておく必要がある。この境界を吸収境界といい、その条件を吸収境界条件という。これは限られた解析領域内での数値計算において、計算機メモリと計算時間の縮小につながる。吸収境界条件が完全でないと、それからの反射波が解析領域に戻り、誤差の原因となる。

吸収境界では、先に述べた FDTD 法のアルゴリズムが使えないため、特別な扱いが必要となる。これまでに多くの吸収境界条件が提案されてきたが、大別すると「吸収境界で反射がないという近似的な微分方程式から導かれたもの」、「境界に仮想的な媒質を置いて入射波を減衰させようという発想から生まれたもの」に分けられる。簡単で、しかもオールマイティに使えるという方法は存在しないが、PML, Mur, Higdon といった方法が現在よく使われている。

本研究では、Mur より有効であり、しかもプログラムが容易である Higdon の 2 次の吸収境界条件 [?] を適応させ、境界からの反射などを抑えた。

3. 音響 FDTD 法演算支援 LSI

3.1 システムの概要

専用 LSI は、アルゴリズム全体 (図 4) をハードウェア化するのではなく、全体のアルゴリズムの一部、処理に時間のかかる部分をハードウェアに置き換えて全体の処理時間の向上させる。

システムの構成図を図 3 に示す。処理に時間のかかる部分を専用 LSI で演算し、その他の処理を汎用 CPU で行う。

3.2 音響 FDTD の計算時間

LSI を設計するに当たり、実際に音響 FDTD 法で音場解

表 1 音響 FDTD プログラムの条件

Table 1 A condition of a sound-FDTD program.

セルサイズ	115 × 115 × 134 [セル]
格子幅	2.0 [mm]
時間刻み幅	0.25 [μs]
入力信号周波数	21 [kHz]
計算ステップ回数	6000 [回]

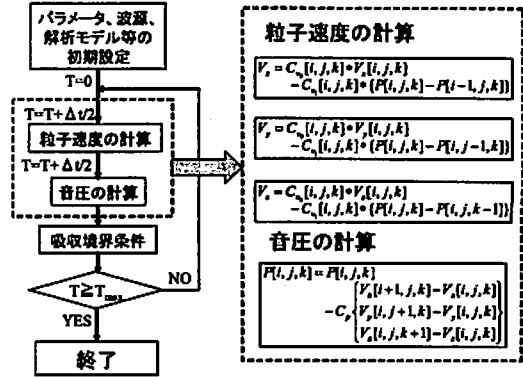


図 4 計算の流れ
Fig. 4 A flow of a calculation.

表 2 音響 FDTD の処理時間

Table 2 Process time of a sound-FDTD program.

入力信号の更新	0.25 [%]
粒子速度の更新	62.43 [%]
音圧の更新	35.25 [%]
吸収境界条件の演算	2.04 [%]
その他	0.03 [%]

析を行うプログラムを作成し、どの処理に時間がかかるのかを調査した。作成したプログラムの各条件をまとめたものを表 1 に、計算の流れを図 4 に示す。このプログラムでは、まず各パラメータの初期設定を行い、(1) 粒子速度の更新、(1) 音圧の更新、(2) 吸収境界条件の演算、の (1)~(3) を計算ステップ回数分だけ繰り返す。このほかに、入力信号の更新、ファイルへのデータの書き出しなど細かな処理を行っている。

このプログラムにおける各処理の、全処理時間に占める割合を表 2 に示す。ここで、粒子速度の更新の時間は x,y,z 成分それぞれの更新にかかる時間の合計値である。

この表から分かるとおり、粒子速度の更新と、音圧の更新で全時間の約 97.5% を占める。これ以外の処理は全体の約 2% ほどしかなく、専用ハードウェアに実装しても効果は小さい。

また、粒子速度の更新、音圧の更新部分以外の処理は、実際に解析を行う人によって異なる場合が多い。例えば、入

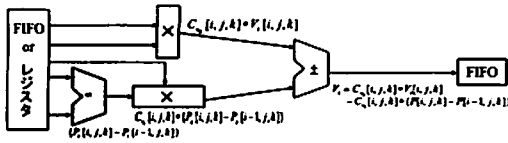


図5 粒子速度更新用演算パイプラインの概要

Fig.5 An operation pipeline for sound pressure update.

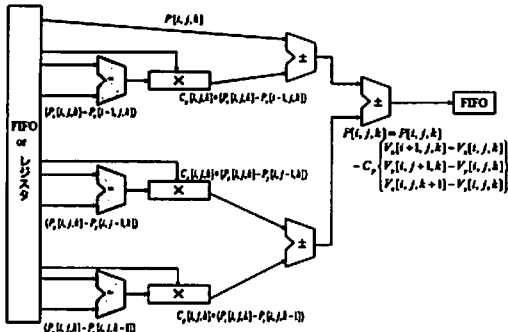


図6 音圧更新用演算パイプラインの概要

Fig.6 An operation pipeline for sound pressure update.

力信号、ファイル出力、吸収境界条件などが挙げられる。したがって粒子速度の更新、音圧の更新部分以外の演算は専用ハードウェアに実装することは難しく、この部分の処理は汎用CPUで処理することとする。

3.3 LSIの概要

2章で示した結果を元に、粒子速度の更新、音圧の更新部分を専用ハードウェアに実装する。粒子速度の更新用演算パイプラインと音圧更新用演算パイプラインの概要を図5、図6に示す。ここで、 V_x は粒子速度Vのx,y,z成分を表す。

ここで問題になるのは、粒子速度の更新用演算パイプラインと、音圧更新用演算パイプラインを別々に用意しても、並列に動作できないことである。その理由は、音響FDTD法のアルゴリズムにより、粒子速度と音圧のデータに依存関係があるからである。

並列に動作できないのなら、できるだけ資源を共有したほうが良い。粒子速度1成分の更新は、音圧の更新用演算パイプラインを用いれば、計算できることが図5、図6から分かる。しかし、この場合音圧更新用回路の大部分が、粒子速度の更新時には使用されず無駄になる。そこで、粒子速度3成分の更新が可能な回路を用意し、その回路で音圧の更新ができないか検討した(図7)。

図7中の楕円部分は、粒子速度更新用、音圧更新用回路の共通部分である。また、3つの加減算器は、配置こそ違うが共通の資源として使える。乗算器が3つ余りのみなのでこちらの方が無駄が少ない。

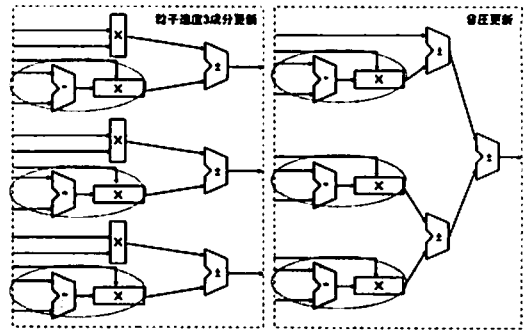


図7 粒子速度と音圧更新用演算パイプラインの比較

Fig.7 Comparison operation pipelines of particle velocity and sound pressure update.

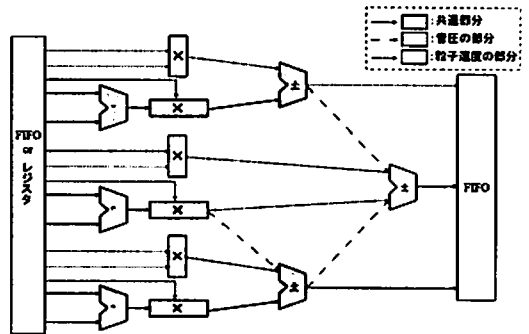


図8 演算パイプラインの概要

Fig.8 A summary of an operation pipeline.

これを元に作成した演算パイプラインの概要を図8に示す。粒子速度と音圧の更新処理を切り替えは、回路中の一部の経路を選択することで行う。演算パイプラインは、入力側のレジスタ及びFIFOにデータがそそい次第演算を行う。データの経路や加減算器の正負などの各種パラメータは、演算を開始する前に専用のレジスタに保存し、演算中はその保存された値が直接セクタなどの制御信号として用いられる。実際に作成した演算器に用いた主なモジュールを表??に示す。表中の()内の数字はそのモジュールのパイプライン段数である。

各入力データ用にそれぞれレジスタとFIFOが用意してある。配列データである場合はFIFOを、定数データである場合はレジスタを用いる。また、データのFIFOはそれぞれ2つ用意してあり、1つのFIFO中のデータを使って処理している間に、もう一方のFIFOにデータをロードするようになっている。使用中のFIFOの中身が空になったが、もう一方のFIFOにデータがたまりきっていない時は、処理はその間停止するようになっている。

表3 演算パイプラインに用いたモジュール

Table 3 The module which I used for an operation pipeline.

レジスタ	32bit 幅	16 本
入力用 FIFO	32bit 幅×8	30 本
出力用 FIFO	32bit 幅×8	6 本
差分器	32bit 幅 (3 段)	3 本
乗算器	32bit 幅 (4 段)	6 本
加減算器	32bit 幅 (3 段)	3 本
パイプライン段数	13 段	

表4 専用演算器のスペック

Table 4 A spec of special circuit.

動作周波数	100[MHz]
セル数	4568
サイズ	27%

4. 評価

提案した演算器の有効性を確かめるために、ソフトウェア (CPU:Athlon64 3200+, OS:WindowsXP, メモリ:1GB) との比較を行った。

4.1 設計環境

設計に使用したツール、FPGA 等は以下の通り。

- FPGA ボード:Altera 社, EP20K400EBC652-1X
- 論理合成ツール:Altera 社, QuartusII
- 設計言語:Verilog

設計した専用演算器のスペックを表??に示す。

4.2 評価モデル

評価は、以下の環境で行った。

- 音響 FDTD 法プログラム
 - 測定モデル:115x115x134[セル]
 - 計算ステップ回数:6000[回]
- FPGA 動作クロック周波数
 - 100[MHz]
- メモリの動作
 - 理想モデル:FIFO 中に常にデータが入っている状態で処理が途切れずに行われるモデル
 - 遅延モデル:1つの FIFO のデータが空になっても、もう一方の FIFO にデータが埋まりきっておらずデータが埋まるまで6サイクル無駄な状態が生じるモデル

4.3 評価結果

結果を図??に示す。クロック動作周波数が Athlon64 の 20 分の 1 なのにもかかわらず、倍近い処理速度が得られ、この演算器の有効性が確認できた。

5. まとめ

音響 FDTD 法の高速化を目指し、専用 LSI を設計した。

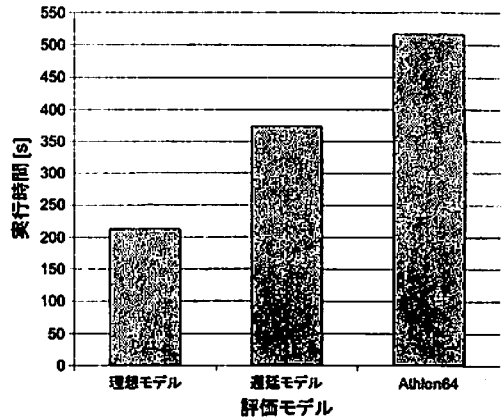


図9 評価結果

Fig.9 An evaluation result.

専用 LSI は音響 FDTD 法の粒子速度の更新、音圧の更新部分を専用演算器で演算する。この2つの処理は音響 FDTD 法を用いた音場解析プログラム中約 97%を占めている。専用演算器は、パイプライン段数 13 段、クロック周波数 100MHz で動作し、理想モデルでは Athlon64 の 20 分の 1 の動作周波数ながら倍近い処理速度を得ることができた。

6. 今後の課題

今後の課題は、動作クロック周波数を上げるためにパイプライン段数の細分化、この専用 LSI のさらなる有効活用できる方法を検討すること等が挙げられる。現在、この専用 LSI を複数用いてさらなる高速化が計れないか検討を行う予定である。

文 献

- [1] K. S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," IEEE Transactions on Antenna and Propagation, vol. AP-14, no. 3, pp.302-307, 1966.
- [2] 宇野 亮, "FDTD 法による電磁界およびアンテナ解析," コロナ社, pp. 262-269, pp.68-72, 1998.