

法政大学学術機関リポジトリ

HOSEI UNIVERSITY REPOSITORY

PDF issue: 2025-06-01

ハードウェア記述言語によるガウス消去法専用演算ボードの設計

檀, 良 / DANG, Ryo / 太田, 錦光 / OHTA, Kanemitsu

(出版者 / Publisher)

法政大学計算センター

(雑誌名 / Journal or Publication Title)

Bulletin of Computer Center, Hosei University / 法政大学計算センター研究報告

(巻 / Volume)

8

(開始ページ / Start Page)

65

(終了ページ / End Page)

72

(発行年 / Year)

1995-03-31

(URL)

<https://doi.org/10.15002/00024699>

ハードウェア記述言語による ガウス消去法専用演算ボードの設計

太田 錦光

法政大学工学部[†]

檀 良

法政大学工学部[†]

本論文では、ガウス消去法専用演算ボードの設計について述べる。設計した回路は、最も計算時間のかかる連立1次方程式を解くための行列演算だけを行い、その他の処理、例えば行列要素の生成などは接続されたホストコンピュータで行う。

また、回路のタイミング解析や動作の確認、性能の評価のためにハードウェア記述言語のVerilog-HDLを用いてシミュレーションしたので報告する。

1. はじめに

計算機処理の高速化、ターンアラウンドタイムの短縮をはかる方法として、処理問題向きの専用計算機を構築し、計算能率を最大限に引き上げる方法が考えられる。この専用計算機は汎用性を犠牲にすることによって、計算手続きをハードウェア化ひいては計算の高速化を可能にするわけである1)。

これまで開発されてきた専用計算機には、すべての計算手続きをハードウェア化する全ハードウェア化方式のものと、計算量が多く最も時間のかかる部分だけをハードウェア化する、いわば部分ハードウェア化方式2)のものがある。前者は、その性能に比較してハードウェアが大きく複雑であり、コストや製作日数もかなりかかる。それに対して、後者は、市販の汎用計算機と共同で計算を進めるため、製作するハードウェアは計算の一部分で良い。従って、コストや製作日数はかなり削減される。計算速度についても、計算のほとんどをハードウェアが担当する事になるので前者とほぼ同程度の性能が得られ、コストパフォーマンスの面から非常に効率がよい。

本稿では、後者のハードウェア化に着目して半導体デバイスシミュレーションや回路シミュレーション等において最も計算時間のかかる行列解法だけを専用ハードウェア化することを提案し、その回路を設計したので報告する。また、回路のタイミング解析及び動作の確認をハードウェア記述言語 (Verilog-HDL) によって行った。

2. 専用計算機システムについて

本専用計算機システムの概念は次のようなものである。専用計算機はワークステーションやパソコンなどの市販の汎用計算機と共同で計算を行う。この汎用計算機をホストコンピュータと呼ぶ。専用計算機とホストコンピュータは適当なインターフェイスで結合されていて、相互に情報を交換できる。計算は以下のように分担される。専用計算機は、定型で単純だが計算量が多くてスピードが要求される部分を担当する。ホストコンピュータは、計算量は大したことはないが、複雑で頻繁に変更する必要がある部分を担当する。ホストコンピュータではC言語やFortranなどの高級言語でプログラムできるので、手続きが少々複雑でもプログラムすることはそれほど難しくない。また、処理手順の変更も簡単である。ホストコンピュータから見ると、専用計算機はハードウェアでできた高速のサブルーチンに見える。

この概念のもとで専用ハードウェアを設計するに当たって、まず最初にしなければならないのは、行いたい計算の問題点を見極め、どのように専用化するかを考えることである。

半導体デバイスシミュレーションを例にすれば、ここでの問題点はシミュレーションを開始してから終了するまで数日間という計算時間が必要であり、そのほとんどが連立1次方程式の行列解法に費やされることである。従って、行列解法の計算さえ速くできれば、半導体デバイスシミュレーションは高速化できる。専用計算機でハードウェア化するのは、その行列解法の部分だけであり、それで十分である。

[†]〒184 東京都小金井市梶野町3-7-2

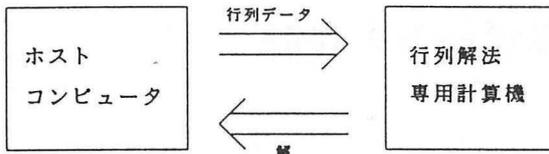


Fig. 1 システムの概念図

Table 1 代表的行列解法

	直接解法	反復解法
解法例	ガウス消去法 コレスキー法	SOR 法 CG 法
精度	良い	悪い
安定性	高安定	不安定
アルゴリズム	簡単	複雑
計算速度	低速	高速

残りの部分は、ワークステーションなどの汎用計算機で計算させる。専用計算機は単に行列解法の計算をするサブルーチンとして扱えるので、今まで使ってきたプログラムをそのまま利用できる。図1にその概念を示す。ホストコンピュータは行列解法の要求が出ると、解くべき行列の各要素を専用計算機に転送する。転送が完了すると専用計算機は、あらかじめハードウェアに組み込まれている計算手続きに従って高速に演算を行う。計算手続きが修了して解がもとまると、その解だけをホストコンピュータへ返す。

3. 行列解法の回路化

3.1. 行列解法

行列解法専用のハードウェアを設計するには、たくさんある解法の中からより良いものを選択しなければならない。表1に代表的な解法とその特徴を示す。

ここで、専用ハードウェア化においてアルゴリズムの複雑さはそのまま回路の複雑さにつながるため、最も簡単で分かりやすいガウス消去法を採用した。

ガウス消去法は、行列を操作して式(1)の形に

する(前進消去)。

$$\begin{bmatrix} 1 & a_{12} & \dots & a_{1,n-1} & a_{1n} \\ 0 & 1 & \dots & a_{2,n-1} & a_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & a_{n-1,n} \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \mathbf{x} = \mathbf{b} \tag{1}$$

ここで、 \mathbf{x} は解ベクトル、 \mathbf{b} は定数項ベクトルを表す。

次に、

$$x_i = b_i - \sum_{j=i+1}^n a_{ij}x_j \tag{2}$$

により、 $i = n-1, n-2, \dots, 2, 1$ の順に解が求められる(後退代入)。

これら前進消去と後退代入で必要な演算は、

$$a' = a/k \tag{3}$$

$$b' = b - k \times a \tag{4}$$

であり、上の二つの演算を交互に繰り返し用いることで解を求める。つまり、二つの演算装置に適切なデータを送り、演算されたデータを所定の位置に戻すという流れ作業を行えばよい。

3.2. 回路設計

どんな計算をするのかが決まったら、大まかな構成図を作る。図2に、その構成を示す。

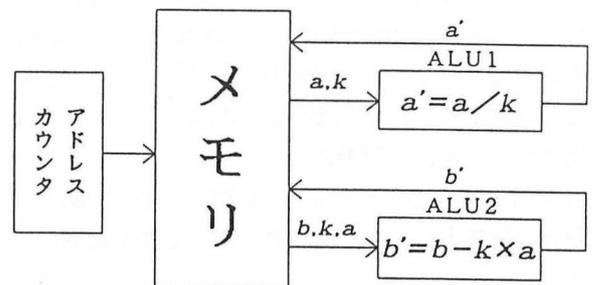


Fig. 2 構成図

この構成図において、メモリからのデータの流れを追ってみる。例えば、ALU1を使ってある行を k で割る作業を行うとする。まず、最初のサイクルでアドレスカウンタはメモリの読み出しアドレスとして k が格納されているアドレスを与え、ALU1へ k の値がセットされる。次のサイクルでは、行の1

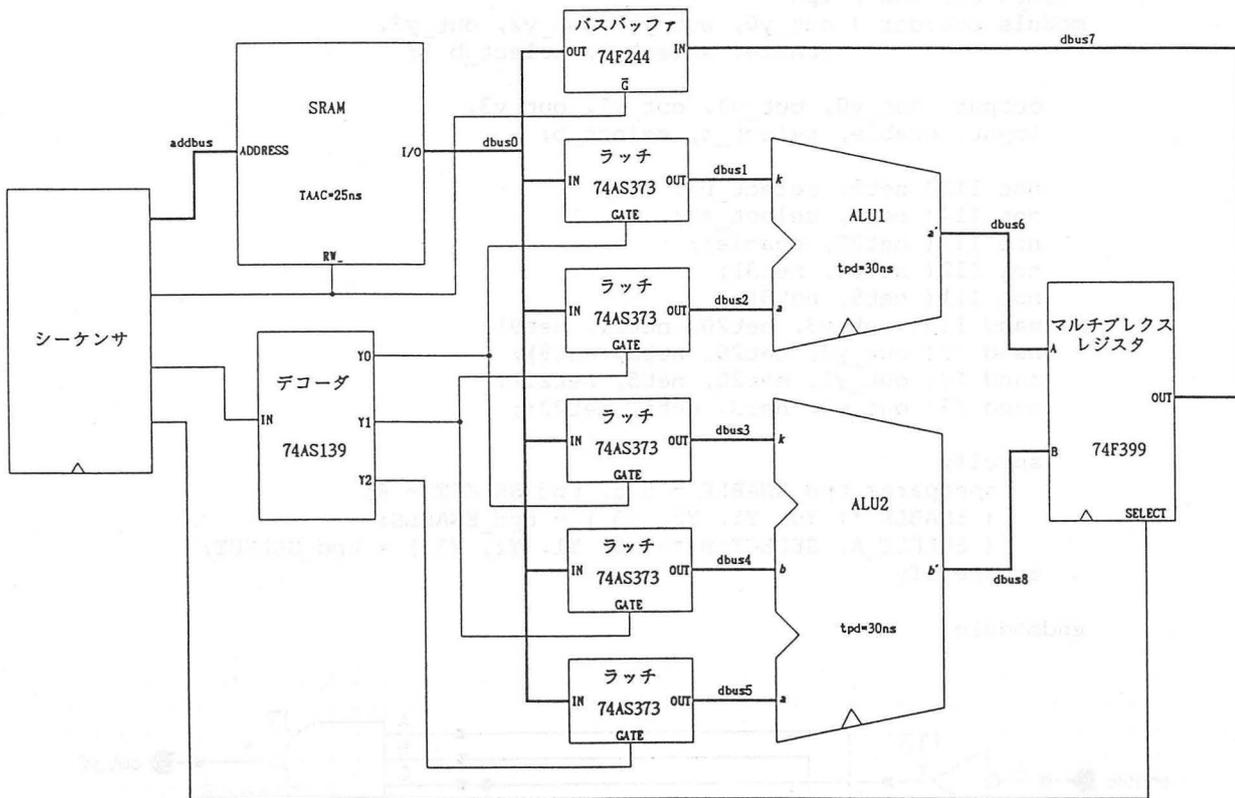


Fig. 3 ブロック図

番目の値 (つまり, k によって割られる値) が格納されているアドレスが与えられ, メモリから読み出されて ALU1 の a としてセットされる. そして次のサイクルでは, ALU1 で演算された値 a' をメモリへ格納する. この時アドレスカウンタは, 先ほどの a が格納されていたアドレスを与え (無変化), メモリを読み出しサイクルから書き込みサイクルへと変化させる. これで a から a' へ更新される. 以下, 行の 2 番目の値の読み出し, 書き込みと同様の作業が繰り返される. ALU2 についても同様である.

ここで注目すべき点は, すべてのデータの流れはアドレスカウンタとメモリの読み出し/書き込みの制御線, ALU1 と ALU2 を選択する制御線などによって, 制御されていることである. アドレスカウンタはハードウェア化されたプログラムであるシーケンサとして働いたのである. また, 基準になるサイクル = クロックにすべてが同期して動作する同期式回路である.

次に詳細な回路設計を行う. 構成図から各ブロックにどんな IC・LSI を使うかを定める. 各 IC ごとにデータの入出力にかかる時間 (伝搬遅延) が決まっ

ているので, データの受け渡しをするタイミングを考えて, つじつまが合うようにしなければならない.

図 3 に, 設計した回路のブロック図を示す. ここでは, 計算の精度を確保するために 64bit の倍精度浮動小数点データを扱う. よってすべてのデータバスは 64bit であり, ALU1 と ALU2 は 64bit の浮動小数点演算器である. また, 各ブロックを構成する素子は, 現在市販されているもので最も遅延の少ないものを選択した. 例えば, 64bit ラッチブロックは, 74 シリーズ IC の 8bit ラッチで最も高速な 74AS373 を 8 個使って実現している. シーケンサブロックにおいては, アドレスを生成しあらゆる制御信号を発信するところであり, 複雑な論理回路になるので PLD や FPGA などのプログラム可能な IC で実現する.

4. 論理シミュレーション

設計した回路の動作を確認し, タイミング解析をするために, ハードウェア記述言語 (HDL) を用いて各素子をモデル化してシミュレーションを行った.

```

`timescale 1ns / 1ps
module decoder ( out_y0, out_y1, out_y2, out_y3,
                enable, select_a, select_b );

    output out_y0, out_y1, out_y2, out_y3;
    input enable, select_a, select_b;

    not I15( net5, select_b);
    not I14( net3, select_a);
    not I13( net20, enable);
    not I12( net21, net3);
    not I11( net9, net5);
    nand I10( out_y3, net20, net21, net9);
    nand I9( out_y2, net20, net3, net9);
    nand I8( out_y1, net20, net5, net21);
    nand I7( out_y0, net3, net5, net20);

    specify
        specparam tpd_ENABLE = 5.5, tpd_SELECT = 6;
        ( ENABLE *> Y0, Y1, Y2, Y3 ) = tpd_ENABLE;
        ( SELECT_A, SELECT_B *> Y0, Y1, Y2, Y3 ) = tpd_SELECT;
    endspecify

endmodule

```

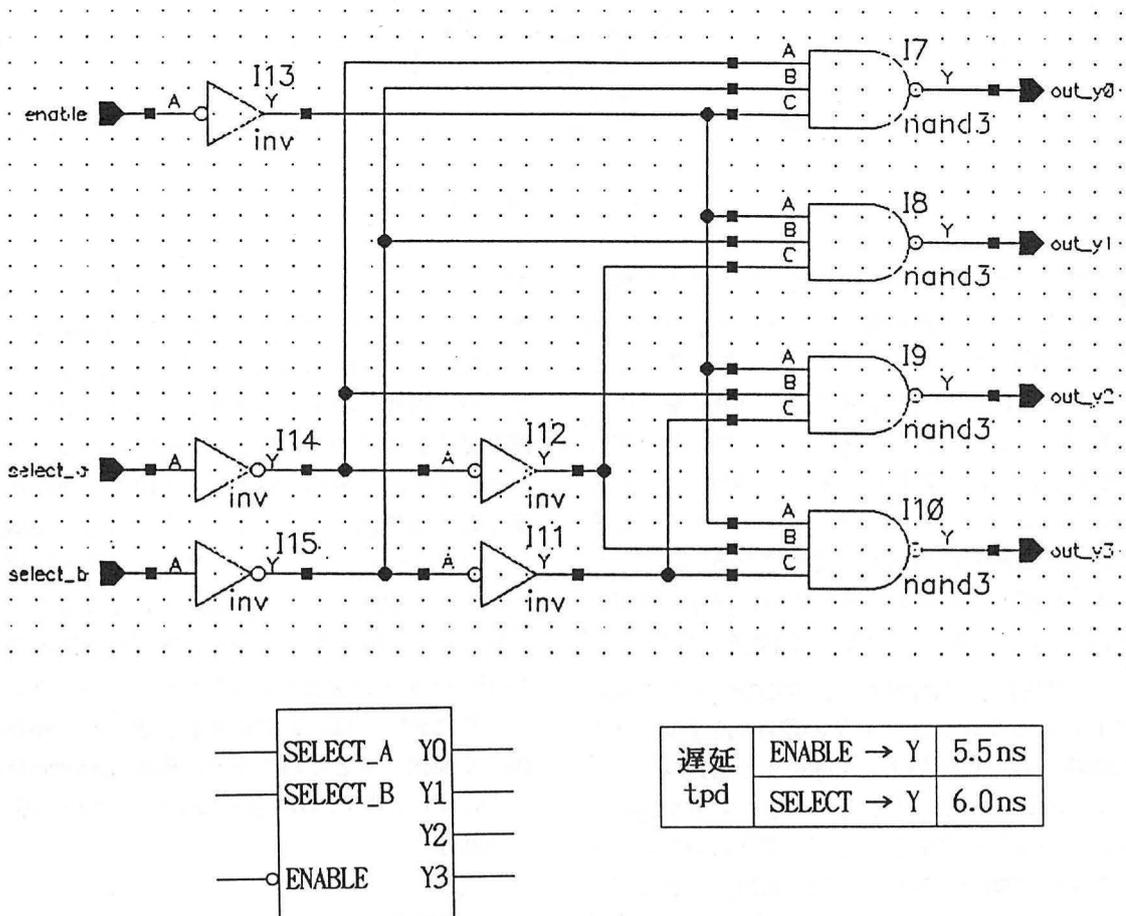


Fig. 4 デコーダの記述

4.1. モデル化

構成するすべての素子について、ハードウェア記述言語の Verilog-HDL で記述し、モデル化した。図4にデコーダICの記述例を示す。記述はゲートレベルで行い、素子内部でのゲートの接続をそのまま記述した。ただし、ALU1, ALU2, シーケンサはゲートレベルではなく、もっと抽象度の高いレジスタ転送レベルで記述した。入力端子から出力端子への伝搬遅延、データのセットアップ/ホールド時間、最小パルス幅などの設定は、データシート(3), (4)の値を用いた。また、システムへデータを転送し、結果を受け取るというホストコンピュータのモデル化も行った。

4.2. シミュレーション方法

シミュレーションは Cadence Design System 社の Verilog-XL シミュレータで行った。テストデータとして図5に示す3×3行列を与える。

4.3. シミュレーション結果

シミュレーションの結果は、オシロスコープまたはデジタル・アナライザと同じような感覚で任意のネットの信号値を波形表示し、信号値の任意の時間での値をモニタすることができる。

図6に行列データの転送中の波形を示す。データの転送はホストコンピュータで制御されるので、システムクロック (clock) と同期しない。ホストはデータ (INPUT) とアドレス (ADDR) とメモリの書き込み制御 (RW-) を送り、データはちよつとの遅延の後データバス 0(dbus0) へ乗り、メモリへ書き込まれていることが分かる。

図7に演算開始の様子を示す。ホストは最後のデータの転送が終わると専用機へ計算の開始を合図する (CALSTR の立ち上がり)。合図が入るとシーケンサはクロックに同期してアドレスを発生する (addbus)。ここからは、ホストと切り離される。始めにアドレス 2 番地の値 4 をアドレス 1 番地の値 2 で割る演算を ALU1 で行っていることが確認できる。

図8に結果の出力を示す。専用機で演算がすべて修了すると、ホストへ合図を出す (CALEND)。ホストはメモリの解が格納されているアドレスにアクセスして、解を引き出している様子が分かる。これで動作の確認ができた。

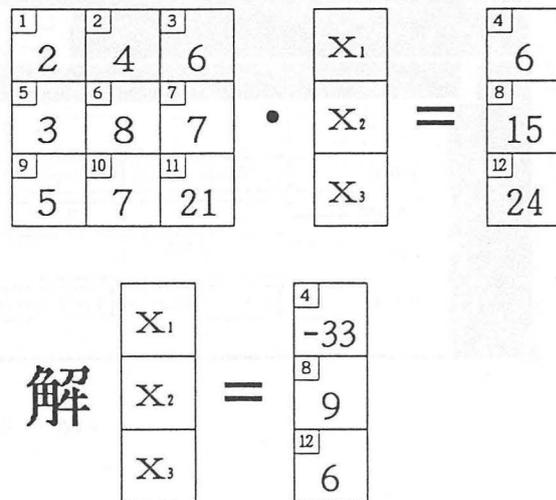


Fig. 5 テストデータ

Table 2 ワークステーションとの比較

900×900 行列の解法	
SPARCserver670MP CPU time	255 s
専用計算機 14.7MHz	75 s

次にタイミング解析の結果、システムクロックは 14.7MHz 以下にしなければならないことが分かった。これ以上になるとデータのセットアップ時間が確保できなくなり、正しい動作が期待できない。

最後にワークステーションと計算速度について比較した結果を表2に示す。これで大幅な計算時間の短縮が期待できる。

5. まとめ

計算機の高速化の一手法である専用計算機の構築に着目して、半導体デバイスシミュレータやその他あらゆるシミュレーションで行われている行列解法を専用ハードウェア化することを提案し、その回路を設計した。ハードウェア記述言語の Verilog-HDL を用いて回路をモデル化し、論理シミュレーション及びタイミング解析を行った。その結果として、大幅な計算時間の短縮が期待でき、実際のハードウェア化に向けた手がかりをつかむことができた。

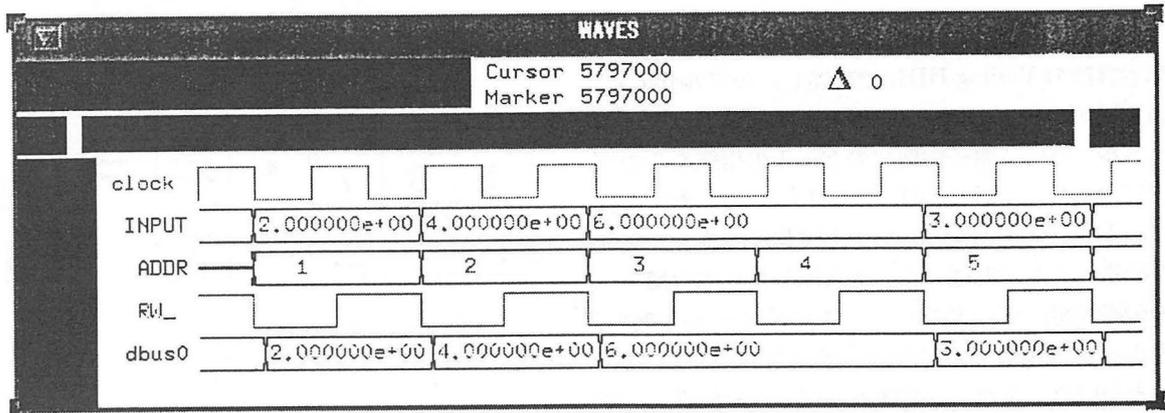


Fig. 6 行列データの転送

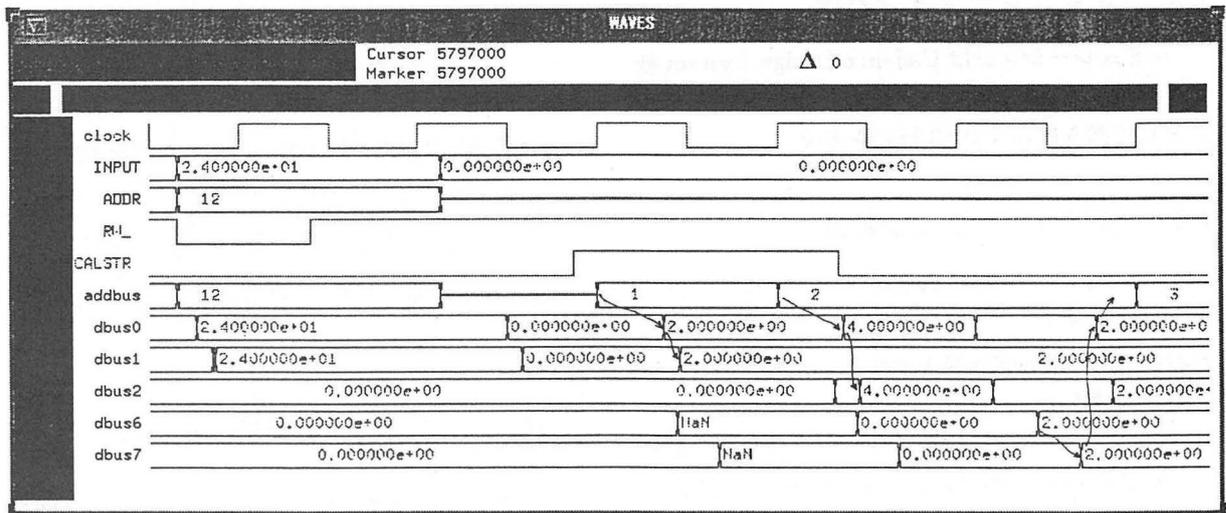


Fig. 7 演算の開始

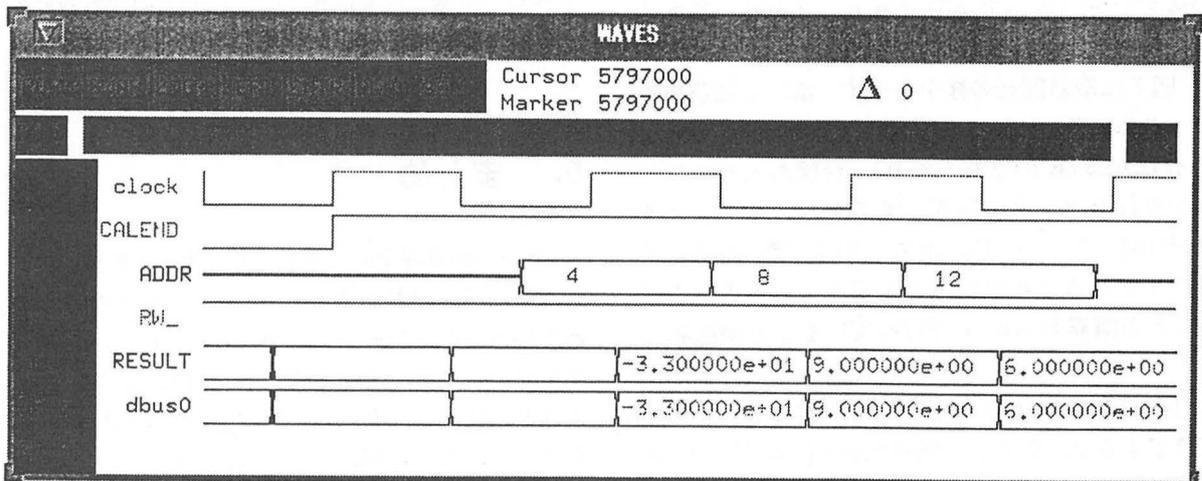


Fig. 8 結果出力

参考文献

- 1) 杉本大一郎, 専用計算機によるシミュレーション,
朝倉書店,1994.
- 2) T. Ito, J. Makino, T. Ebisuzaki and D. Sugimoto, "A special-purpose N-body machine GRAPE-1," Comp. Phys. Comm. , vol. 60, pp. 187-194, 1990.
- 3) '93 最新 74 シリーズ IC 規格表, CQ 出版社, 1993.
- 4) '93 最新 メモリ IC 規格表 [1] (RAM 編), CQ 出版社, 1993.

キーワード*

専用計算機, ガウス消去法, ハードウェア記述言語, 論理シミュレーション
.....

Summary

Design of a Computing Board for Gaussian Elimination Using Hardware Description Language

Kanemitsu OHTA

College of Engineering, Hosei University †

Ryo DANG

College of Engineering, Hosei University†

This paper describes the design of a computing board for Gaussian elimination. It is designed as a back-end processor to carry out the matrix operation in the solution of a system of linear equations. All other relevant calculations are performed on a host computer connected to this board.

Operation of the designed circuit is confirmed and its performance evaluated using a logic simulator based on the hardware description language Verilog-HDL.

Key Words

dedicated computer, Gaussian elimination, hardware description language, logic simulation

†3 - 7 - 2, Kajino-cho, Koganei-shi, Tokyo 184, Japan