

可変成形ビーム描画装置用高速データ変換システムの開発

小山清美¹⁾, 池永修¹⁾, 馬越俊幸¹⁾, 渡辺進²⁾
斉藤玉樹³⁾, 大木慎一郎³⁾, 坂本信二³⁾

- ¹⁾東芝 研究開発センター
川崎市幸区小向東芝町1
²⁾東芝 半導体技術センター
川崎市幸区堀川町580番1号
³⁾東芝マイクロエレクトロニクス
川崎市川崎区駅前本町25の1

あらまし このシステムは可変成形ビーム方式EB描画装置EX-8のマスク描画データを生成する。マスクデータ処理の高速化と高精度化に対応するために、階層処理と並列処理を組み合わせたCADデータからの直接変換バスを開発した。64Mビットメモリの場合、階層処理で図形数を最大5桁圧縮し、4CPU計算機の並列処理で図形演算時間を1/3に短縮した。このシステムにより変換速度が当社従来比で2桁向上し、平均データ変換時間は64Mビットメモリで7分/マスク、130Kゲートのゲートアレイで12分/マスクが得られた。

和文キーワード 電子ビーム, マスク, データ変換, 階層処理, 並列処理, 図形演算

A High Speed Data Conversion System For A Variable Shaped Beam EB Exposure System

Kiyomi Koyama¹⁾, Osamu Ikenaga¹⁾, Shunko Magoshi¹⁾, Susumu Watanabe²⁾
Tamaki Saito³⁾, Shin-inichiro Ooki³⁾, Shinji Sakamoto³⁾

- ¹⁾Toshiba Corporation, Research and Development Center
1 Komukai-Toshiba-cho, Saiwai-ku, Kawasaki 210
²⁾Toshiba Corporation, Semiconductor System Engineering Center
580-1 Horikawa-cho, Saiwai-ku, Kawasaki 210
³⁾Toshiba Microelectronics Corporation
25-1 Ekimae-Honcho, Kawasaki-ku, Kawasaki 210

Abstract

A high speed data conversion system Vishamon has been developed to generate mask pattern data for the variable shaped EB exposure system EX-8. In the data conversion, high speed data processing techniques are adopted using both hierarchical process and parallel process. When a 4-CPU parallel workstation was used, the average data conversion time was 7min for 64Mbit DRAM patterns, and 12min for 130K-gate gate array patterns. Vishamon has achieved a high conversion throughput comparable to the mask exposure rate of EX-8.

英文 key words Electron Beam, Mask, Data Conversion, Hierarchical Process
Parallel Process, Figure Operation

1 はじめに

LSIの大規模化と共にマスクの高精度化が進んでいる。一方では位相シフトや超解像などの新技術の登場によって、光学ステッパを使うリソグラフィ技術が1GビットDRAMの量産まで延命する可能性が強くなってきた¹⁾。高集積LSIのマスク製作技術として、電子ビーム(EB)描画技術の戦略的役割は一層増してきている。

東芝は16MdRAM以降の高精度マスク対応で高速EB描画装置EX-8²⁾を開発した。EX-8は可変成形ビーム(VSB)、ベクタ走査、ステージ連続移動の描画方式を採用し、平均5マスク/H(16Mクラス)の描画スループットを達成した。

EX-8のマスク描画データを供給するために、データ変換システムVishamonが開発された³⁾。本開発では、EX-8の描画スループットと同等の変換速度を実現するため、階層処理と並列処理を組み合わせた高速データ技術を開発した。

2 Vishamonデータ変換

2.1 設計データ入力

VishamonではCADの設計データ(GDS II形式)から直接EX-8のマスク描画データを作成する。中間データから変換する場合に較べて、設計データの精度を落とさずにデータ変換が行え、また設計-マスク描画のターンアラウンド時間が短縮できる

設計データ入力では幅付き線分、多角形等の図形は台形表現に変換するが、階層処理がし易いように階層構造は保ったままファイルに格納する。

2.2 図形演算

図1に主な図形演算処理を示す。以下に示すように、これらの処理には種々の目的がある。

図形演算の主な目的はリソグラフィ工程で発生する不都合な物理現象の補正である⁴⁾。例えば重なり除去処理では、CADデータでは許された図形同士の重複を除去する。これによって多重描画による寸法変動が回避され、高精度マスク製作が可能になる。またマスク製作やウェーハプロセスでは寸法変換差が発生するが、リサイズ処理では予めマスク図形の太めや細め

を行って補正する。

層間演算はCAD処理時間を短縮するための処理である。メモリLSI等の設計では他の層の合成によって得られる層は設計を省略して設計や検証の効率化を図る。層間演算は省略された層の論理合成に使われる。1層を合成するのに複数層の論理演算が必要になるためCAD上で行うと時間がかかる。層間演算では代わりにデータ変換の中で高速に処理する。

図形演算処理はビットマップハードウェアで比較的容易に実現できる。このため、殆どの丸ビーム方式描画装置自身が重なり除去、白黒反転機能を備え、リサイズ機能を持つものもある。一方、EX-8のようなVSB方式描画装置ではハードウェアの実現は困難で、その分データ変換の負荷は重くなる。

2.3 VSB変換

VSB変換では図2に示す処理が行われる。これらの処理は図形演算の場合よりEX-8の描画装置に依存した処理であると言える。

倍率変換は縮小投影型ステッパ用に拡大マスクを作成するために、或いはチップサイズを縮小して生産コストを下げる場合などに使う。また鏡面反転はステッパの結像条件に合わせて鏡像のマスクパターンを作成するために用いる。

領域分割は、ベクタ走査、ステージ連続移動の描画方式による描画を可能にするため、電子ビームのサブフィールド偏向領域単位でパターンを分割する処理である。図3にEX-8の描画方式を示す。

基本図形分割やフォーマットでは図形を台形分割したり、VSB描画フォーマットに並べ替えたりして、EX-8の描画ハードウェアが解読可能な形に整える。フォーマット時には階層情報をフレームに振り分ける、フレーム分割が行われる。任意角図形の基本図形分割では45度台形近似を行う。

VSB変換ではこの他、描画装置内のパターンメモリのオーバフローを無くすため、複数フレームで参照されるデータを共通定義してデータ圧縮を行う(共通データ定義)。また、描画解像度以下の微小図形が発生しないように図形毎に分割位置を調整する(マージン分割)。

3 高速化戦略

Vishamonの高速化戦略を図化したものを図4に示す。まず階層処理では、メモリなどの設計データがセル（図形や他のセルへの参照が定義された領域）の階層構造を成して作られているのを利用し、種となるセルを抽出し、セル毎に独立に演算を施す（セル単位演算）⁹⁾。種セルしか処理しないため図形数が圧縮でき、大規模データが短時間で処理できるようになる。

一方、ロジックやゲートアレイの様に階層規模が小さい設計データでは並列処理による高速化を行う。階層は全展開して領域に区切り、複数CPU計算機で並列処理する。同様に階層処理のセル単位演算も並列処理して、更に時間短縮する。

また、メモリ、ロジックに関わらず、フレーム分割後のVSB変換処理（フォーマットリングなど）はフレーム単位で並列処理する。

4 階層最適化

4.1 階層前処理

階層構造を持つ設計データが全て階層処理のセル単位演算に適している訳ではなく、不具合が発生する。その例を図5に示す。この不具合はセルが他のセル或いは図形と重複することによって発生するため、重複しているセルを展開する方法をとる。これをセル間重複除去と呼ぶ。図6に示すようにセル間重複除去では階層の組み替えが行われ、副作用として図形数が増加する。他にリサイズをセル単位演算するにはドーナツ処理¹⁰⁾と呼ばれる前処理を行う。

大規模メモリ製品に対しては、繰り返しパターンの前処理をいかに効率よく行うかが高速化の鍵となる。このため、以下の前処理機能を導入した。

アレイセルでセル同士が重複している場合、セル間重複除去ではアレイを全展開してしまい、図形数が大幅増加する。このため事前にアレイの構成変更を行って重複を除去し、全展開を回避する（アレイ正規化）。アレイセルが図形や他のアレイセルと重複を持つ場合も同様の危険があるため、重複部分のみを展開して、残りのアレイを保存する処理を行う¹¹⁾（アレイ部分展開）。更に、セル同士が重複を持たないアレイセルの場合も、リサイズ処理による重複を考え

て、リサイズ量ののりしろに含まれる図形を追加した9種類のセルでアレイを再構成しておく（アレイ最適化）。

アレイ正規化、アレイ最適化の例を図7、8に示す。

4.2 処理法の選択

Vishamonでは設計パターンのレギュラリティ（ファクタ）⁷⁾を計算して自動的に処理法を決定する。レギュラリティは

$$\frac{\text{階層を全展開した時の図形数}}{\text{未展開セルの図形数の総和}}$$

で表され、パターンの規則性が高くなるほど大きい値を示す。また、階層前処理の副作用によって次第に減少する。

まず設計データ入力時のレギュラリティと基準値 T_1 との比較によって選択する。これにより、メモリ等パターンの繰り返しが多い製品では階層処理が、またロジックやゲートアレイ等不規則なパターンで構成された製品では全展開処理が選択される。

また、途中セル間重複除去等によりレギュラリティが基準値 T_2 （ $<T_1$ ）以下に減少した場合には、全展開処理へ切り替える。これにより、オーバーヘッドが増えて階層処理の効率が逆に低下するような場合、展開処理への切り替えが自動的に行われる。

処理法選択のフローを図9に示す。この方法によって、メモリからロジックまで各製品に対して最適な処理法がとられる。

5 並列処理

並列処理は密結合のマルチCPU計算機を使って行う。ここでは、重複除去、リサイズ、白黒反転等の図形演算、及び領域分割などのVSB変換を対象とした並列処理を中心に説明する。既に述べたように、フレーム分割後のVSB変換処理はフレーム単位で並列処理を行なう。

並列プロセス起動にはファイルのオープン／クローズ、メモリの割付け／解放といったオーバーヘッドが伴う。一方、大規模LSIではセル数は数百～数千個に及ぶため、オーバーヘッドが増えないようにセルをグループ化し、ユニット単位で並列プロセスを起動する（階層処理の場合）（図10）。更に、ユニットの図形数を均一化してプロセス負荷をバランスさせる。こ

れによってCPU利用率に偏りが無くなり、最短のターンアラウンド時間が得られるようになる⁸⁾。

全展開処理ではフラットレベルに階層展開し、チップ領域を決まった大きさに分割してユニットを作成する。リサイズが含まれる場合、ユニット境界部での不具合を無くするため境界をオーバーラップさせる。

並列プロセスの図形演算はスキャンライン法⁹⁾で行う。本アルゴリズムは同一X軸上に並んだ図形に対して処理時間が $O(n \log n)$ となる特性がある(X方向スキャンの場合)。このため、処理時間が増えないようにユニット内部でセルをX方向に一列に配置している。一方、全展開処理のユニットではスキャンライン法にワークリスト法⁹⁾を組み合わせ、対象図形の抽出時間が短縮できるようにした。

6 インプリメンテーション

図11にVishamonデータ変換の処理フローを示す。全体制御部は変換自動化の機能を持つ。ここでは従来データ変換のジョブデックに相当するシェルスクリプトを自動生成する。これに従って変換処理部のデータ変換が実行される。データ変換プログラムはC言語で記述し、実行はワークステーション(EWS)で行う。

プログラムの実行ではUNIX OSから多くの支援を受ける。全体制御部で生成されたシェルスクリプトはUNIXのコマンドアナライザ(C-Shell)で解読する。変換処理部では、図形演算部及びVSB変換部に於けるユニット、或いはフレーム単位の並列プロセス起動はUNIXのマルチタスキング機能(fork)を使って行われる。また変換中のファイルアクセスでは、階層構造ファイルやフラット形式ファイルを仮想記憶上に構成し、またUNIXの動的メモリ管理機構(malloc)を使う。

7 性能評価

7.1 階層処理の効果

(1)メモリ製品の場合

最初に、メモリ製品に対する階層処理の効果をも64MビットDRAMを使って評価した。

図12は階層処理による図形数圧縮効果を示す。処理図形数は設計図形数に対して3~5桁圧縮された。

次に、このメモリのデータ変換時間を測定した。測定は単体CPU EWS(25MIPS)上で行った。図13に各層(レイヤ)の変換時間を示す。セル間重複除去後のレギュラリティは各層とも基準値 T_2 以上で、このため図形演算は階層処理で行われた。平均データ変換時間は目標を上回る9分/マスクが得られた。

(2)ロジック製品の場合

次に、ロジック製品とゲートアレイ製品(130Kゲート)で測定を行った。結果を図14に示す。ロジック製品のレギュラリティは設計データ入力時に基準値 T_1 以上で、セル間重複除去後に T_2 以下に減った。このため最初に階層処理が選択され、途中から全展開処理に切り替わった。一方、ゲートアレイでは設計データ入力時のレギュラリティが T_1 以下で、最初から全展開処理が選択された。

共に図形演算の処理時間がメモリの場合より長い。平均データ変換時間はロジックで14分/マスク、ゲートアレイで24分/マスクであった。

7.2 並列処理の効果

並列処理の効果を4CPU EWS計算機を使って測定した。ゲートアレイ製品の1層の測定結果を図15に示す。並列度4の場合、図形演算時間は1/3に、VSB変換は1/2に短縮した。また全体の処理時間は1/2に減少した。(この測定ではチューニングを殆ど行っていない)。

次に64MビットDRAMとゲートアレイのデータ変換時間を4CPU計算機を使って測定した。平均データ変換時間はメモリが7分/マスクに、ゲートアレイは12分/マスクに短縮した。

7.3 従来データ変換との比較

Vishamonの性能を当社の従来データ変換と比較した(図16)。測定は64Mメモリの2層(3Poly, 1Al)について行った。従来データ変換の測定では4CPUミニコンを使用した。Vishamonの測定は単体CPUワークステーションで行った。MIPS性能は同等である。図16に示すように、V

i s h a m o n の性能が 2 桁近く上回った。

8 まとめ

V i s h a m o n データ変換システムでは、G D S II 形式の C A D データを直接変換して E X - 8 の描画データを作成する。大規模 L S I のデータ変換を E X - 8 の描画スループットと同等の速度で行うために、階層処理と並列処理を組み合わせた高速処理法を開発した。64Mビット D R A M では階層処理により図形数が最大 5 桁圧縮された、平均データ変換は 9 分 / マスクに短縮された。また、4 C P U E W S の並列処理により図形演算時間が最大 1 / 3 に減少し、ゲートアレイの平均変換時間が 1 2 分 / マスクに短縮された。また、当社従来比で性能が 2 桁向上した。

参考文献

- 1) G. Owen et al., "1/8th μ m Optical Lithography", Proc. of EIPB'92 (1992)
- 2) K. Koyama et al., "CAD Oriented EB Data Conversion System Vishamon", Proc. of EIPB '92 (1992)
- 3) T. Takigawa et al., "A High Accuracy and High Throughput Electron Beam Reticule Writing System for 16M Dynamic Random Access Memory Class And Beyond Devices", J. of Vac.Sci. Tech., pp.1877-1881(1990)
- 4) D.W.Grobman, "An Overview of Pattern Data Preparation For Vector Scan Electron Beam Lithography", Proc. of ICCC, pp.558-561 (1980)
- 5) K. Koyama et al., "Integrated Data Conversion for The Electron Beam Exposure System EX-7", J. of Vac. Sci. Tech., pp.2061-2065(1988)
- 6) K. Koyama et al., "Shape Data Operations for VSB EB Data Conversion Using CAD Tools", Jpn. J. Appl. Phys., pp.2329-2332 (1989)
- 7) T. Whitney, "A Hierarchical Design-Rule Checking Algorithm", LAMBDA, First Quarter, pp.40-43(1981)
- 8) A. L. DeCegama, "Parallel Processing Architecture and VLSI Hardware", Prentice-

Hall (1989)

- 9) H. S. Baird, "Fast Algorithms for LSI Artwork Analysis", Proc. 14th DAC, pp.301-311 (1977)

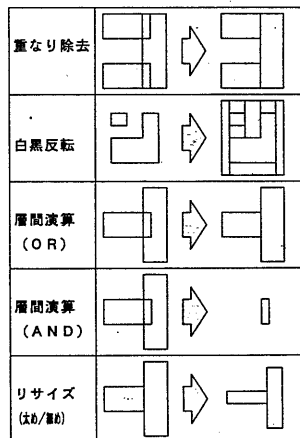


図 1 図形演算の処理

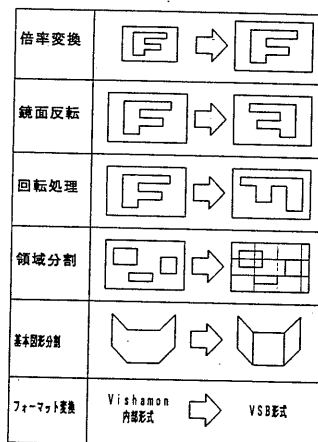


図 2 V S B 変換の処理

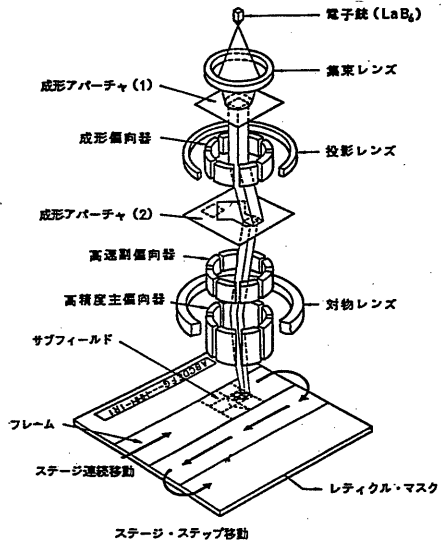


図3 E X - 8 の描画方式

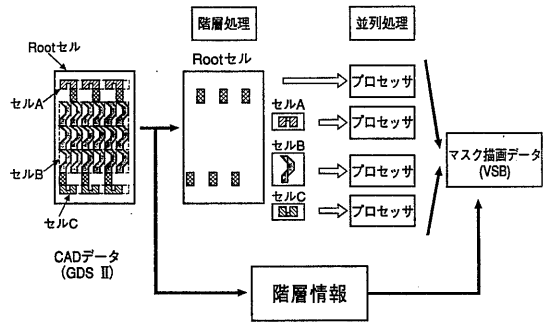


図4 高速化戦略

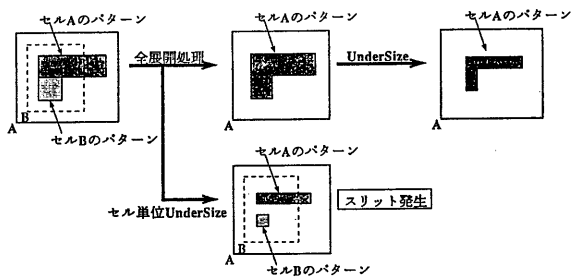


図5 セル単位演算による不具合の例

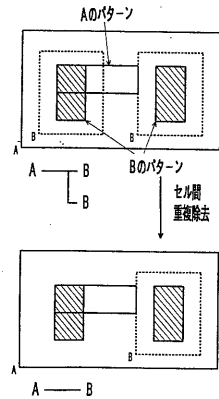


図6 セル間重複除去

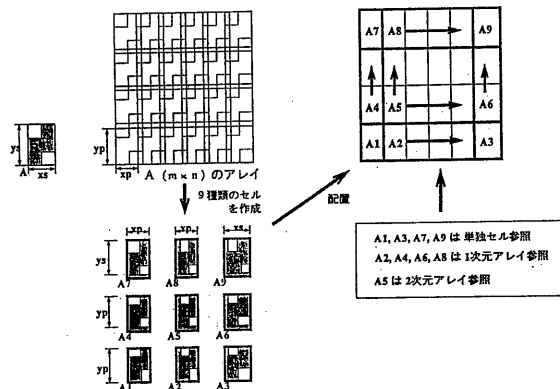


図7 アレイ正規化処理

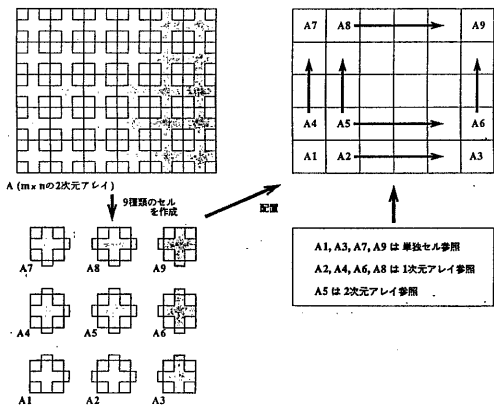


図8 アレイ最適化処理

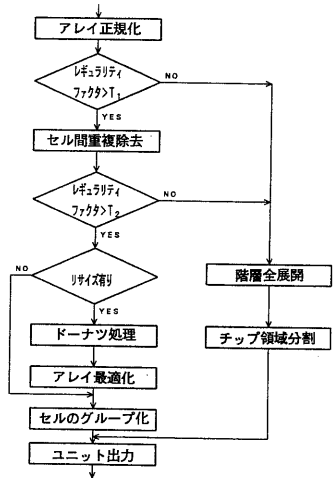


図9 処理法選択のフロー

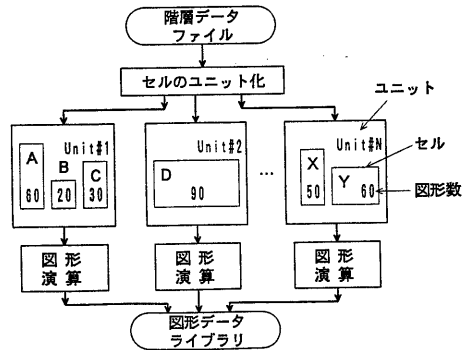


図10 ユニットの並列処理

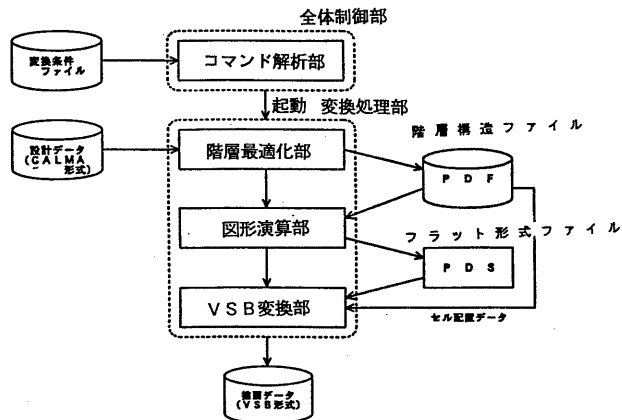


図11 データ変換の処理フロー

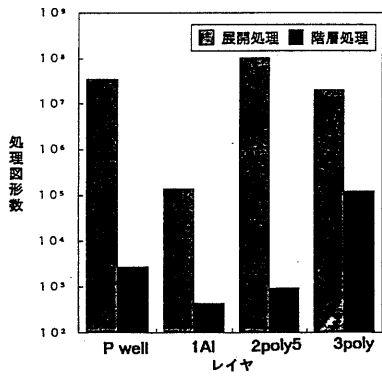


図12 階層処理による図形数圧縮効果

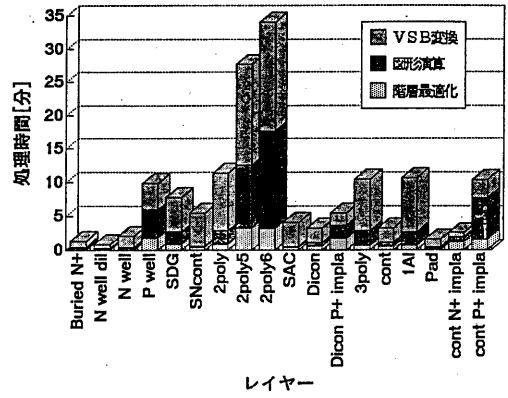


図13 64 Mメモリの変換時間

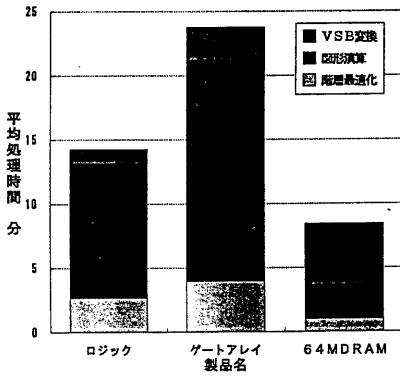


図14 各種製品の変換時間

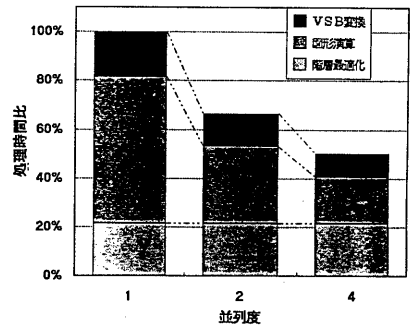


図15 並列処理の効果

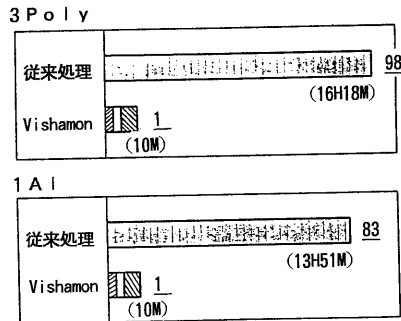


図16 従来データ変換との比較