

航空宇宙技術研究所報告

TECHNICAL REPORT OF NATIONAL AEROSPACE LABORATORY

TR-1211

三次元Skew配列による並列計算機の性能

原 田 公 一 ・ 岩 宮 敏 幸

1993年8月

航空宇宙技術研究所
NATIONAL AEROSPACE LABORATORY

三次元Skew配列による並列計算機の性能*

原 田 公 一*¹ 岩 宮 敏 幸*¹

Efficiency of a Parallel Computer with 3-Dimensional Skew Matrices

Koichi HARADA and Toshiyuki IWAMIYA

ABSTRACT

To examine the efficiency and applicability of a parallel computer system, a software program was designed and developed which simulates a massive parallel computer system with 128×128 processing elements. The software simulates the computer's operation at the clock level, whereas the assembly language describes its execution procedures. Setting the boundary condition in a fluid dynamics program in a parallel computer often causes the processing elements to freeze because the boundary-grid-point procedures differ from the inner-grid point procedures.

First, the operation was evaluated by comparing the resultant parallel processing time using the test program to the corresponding sequential processing time obtained with a sequential processing element. Next, skew matrices technique was used to avoid freezing of the operation. In addition, efficiency was further evaluated by comparing the parallel processing time using skew matrices to the sequential processing time, and the relation between the data structure and the improved efficiency is described.

Finally, the formulation for maximum efficiency of the massive parallel computer, is presented.

Key Words: Parallel computer, Parallel processing, Numerical simulator

概 要

並列計算機の性能・有用性を調べることを目的とし 128×128 台の演算器を並べた並列計算機を想定し、その動作をクロックレベルで模擬するシミュレータおよび処理手続きを記述するためのアセンブラを開発し下記のことを行い並列計算機の性能評価についてまとめた。

流れ場解析問題を並列計算機で処理する時、境界条件の設定手続きが内部格子点の手続きと異なるため多くの演算器を凍結することがある。しかし3次元問題ではデータ構造の工夫によりその凍結を回避することが可能である。問題は回避した時の性能をどこまで改善できるかにある。

そこで立方体空間内 (Cavity) の流れ場解析プログラムをアセンブラを用いて並列計算機に移植し、そのプログラム命令を並列計算機のシミュレータで追跡し、一台の演算器による処理時間と 128×128 台の演算器によるそれとを比較し、演算性能について調べその問題点を抽出した。

次ぎに境界条件の設定について Skew マトリックスによるデータ構造の改良を行い、その改善度を再度シミュレータで確認した。

さらに並列計算機の性能について定式化を行いデータ構造の改良と性能向上との関係を明らかにした。

* 平成5年3月9日 (received 9 March 1993)

*¹ 数理解析部 (Computational Sciences Division)

1. はじめに

近年、物理現象をスーパーコンピュータを用いて解明する数値シミュレーション技術の進展は目覚ましく、より複雑な現象を精度よく迅速に解析するためにさらに処理性能を大幅に増強した新しいスーパーコンピュータとして並列計算機が市販されるようになってきた。今後は回路の集積度が進みパイプラインの1チップ化が可能になればなおのこと性能向上策としての並列化は避けられないものと思われる。

しかし、並列計算機の処理性能は応用プログラムの性質に大きく依存し、演算器の増設と性能向上とは同一義ではない。特に境界格子点と内部格子点における処理手続きの異なる流れ場解析プログラムでは内部格子点については高効率な演算ができて境界のそれでは性能が大きく低下することが考えられる。並列計算機の性能・有用性を検証するためには、並列計算機を実際に開発し解析プログラムを動作させ、その処理時間により性能評価を行う方法が確かであるが、多額の費用・労力・時間を要する。他方、典型的なルーチンを選びその処理時間を机上で算出する簡便な方法がある。しかし、後者の場合はルーチンを正確に並列計算機の命令に変換しているか、またその命令を追跡する過程が正確に計算されているという保証が無い。そこで、以下に述べる方法で並列計算機の性能を推定することを計画した。

(1) 演算器を128×128台並べた並列計算機を想定する(流れ場解析プログラムでは流れ方向に約100点、境界層方向に約100点を取ることが実用化の目安となる。この数に近い2の冪乗として128×128を採用した)。

(2) 応用プログラムを並列処理に適應させ(計算法の変更およびメモリへの割り付け等を含む)並列計算機に移植する。

(3) 移植したプログラムの命令を並列計算機のシミュレータで追跡し一台の演算器による処理時間(クロック数)と128×128台の演算器によるそれとを比較し、演算性能についてプログラム内のルーチンごとに調べ、その問題点を抽出する。

(4) 次に境界条件の設定についてデータ構造上の改良を行い、その改善度を模擬実験で確認する。

(5) 最後に並列計算機の性能について定式化を行いデータ構造上の改良と性能向上との関係を明らかにし並列計算機の性能評価についてまとめる。

以前、二次元の円柱まわりの流れ場解析プログラム¹⁾による性能評価について述べたのに続き²⁾⁻⁴⁾、三次元非圧縮非定常ナビエーストークス方程式による解析プログラムによる並列計算機の性能評価について報告する。

2. 並列計算機のイメージ

計算機を命令の解釈・制御を行う制御部とデータの加工を行う演算部およびデータの格納を行う記憶部に分割した場合、並列処理の可能な形態としては

- (1) 制御部・演算部から成る処理装置を複数台配し記憶部を共有する形態(演算部に局所的な記憶部を有する形態も含む。図1)
- (2) 制御部・演算部・記憶部から成る計算機を結合し必要に応じてデータを転送する形態(図2)
- (3) 複数の演算部と複数の記憶部とを対応付ける結合回路を有し、原則として1個の制御部により同一処理を行う形態(図3)

がある。

科学技術計算を並列処理で高速化するためには各演算器の負荷を均一にし、同期を取りながら処理することが必要である。

しかし、(1)と(2)では非同期で動作しているため(1)では処理装置の間で、(2)では計算機の間で互いに同期を取るための負荷が避けられずその動作も複雑であるため処理装置・計算機を大規模に増設してもそれに見合う性能向上が明確には判定できないのに対し(3)ではクロックまたは命令レベルでハードウェアによる完全な同期を取るため動作も簡明

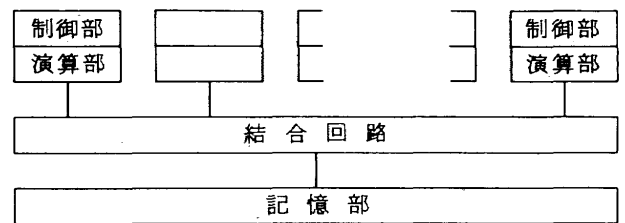


図1 マルチプロセッサ

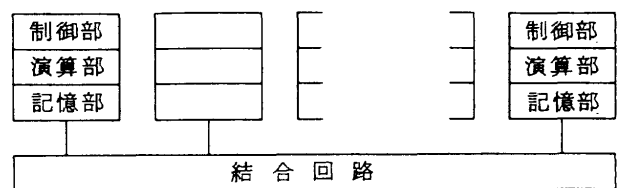


図2 マルチコンピュータ

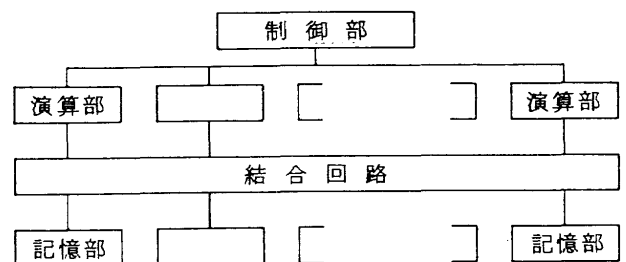


図3 並列計算機

であり増設に見合う性能向上が他に比べて容易に評価できる。このため並列計算機のイメージとして(3)の方式を基本として構成を決めた。

また、アレイデータの演算では演算器を128×128台、記憶部を同数配置し、複数の演算器と複数のメモリとを接続する結合回路として隣接結合し両端をリングとするトラス結合網を採用した。

また入出力処理およびシステム全体を制御・監視する処理装置をスカラ/アレイデータを演算する処理装置から各々独立して設けることとした。

2.1 構成

図4に並列計算機の構成を示す。これはインストラクションメモリ部、制御処理部、データ処理部、入出力処理部、スカラデータメモリ部、アレイデータメモリ部より成る。

なお、三種の各演算器にはコミュニケーションのためのレジスタがあり、制御データ演算器のCMNCR 1/4とスカラデータ演算器のCMNCR 2/5とは1対1で、CMNCR 2/5とアレイデータ演算器のCMNCR 3/6(1, 1)~CMNCR 3/6

(128, 128)とはマスクレジスタ制御下のスキャナを通して1対128×128で接続され互いにデータの送受ができる注)(図5)

またアレイデータメモリ部はアレイデータメモリコントロールユニットと128×128個のメモリ要素から構成されており、各メモリ要素は1語64ビット、読み出し時のメモリアクセス時間を7クロック、書き込み時のメモリアクセス時間を6クロックを想定した記憶装置とする。

さらに、アレイデータメモリコントロールユニットはアレイデータ演算器要素とアレイデータメモリ要素とを対応つける結合網である。ここでは128×128対のデータおよびアドレスを保持するレジスタ (DMBDR 3/6(k, l)) および DMBAR 3 (k, l)) を行列各々隣接結合による上下左右のシ

注) CMNCR はコミュニケーションレジスタを示す。これには3種類 (1/4, 2/5, 3/6) あり各々1, 1, 128×128 個ある。また1/4, 2/5, 3/6としている理由はハードウェアとしては各々1個のレジスタであるが、シミュレータがフォートランで書かれているので整数および実数の処理にあわせて各一対用意してあるからである。

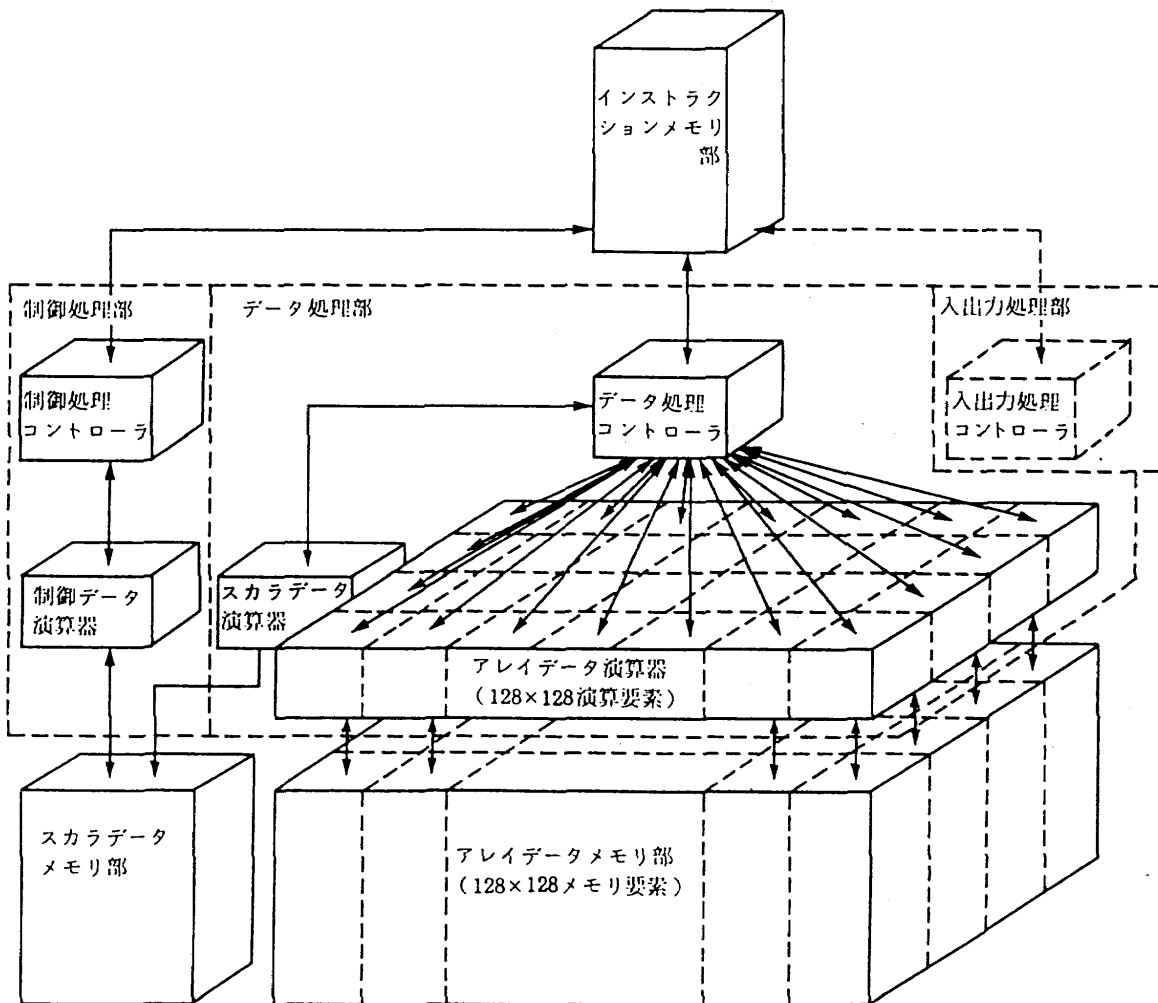


図4 並列計算機の構成

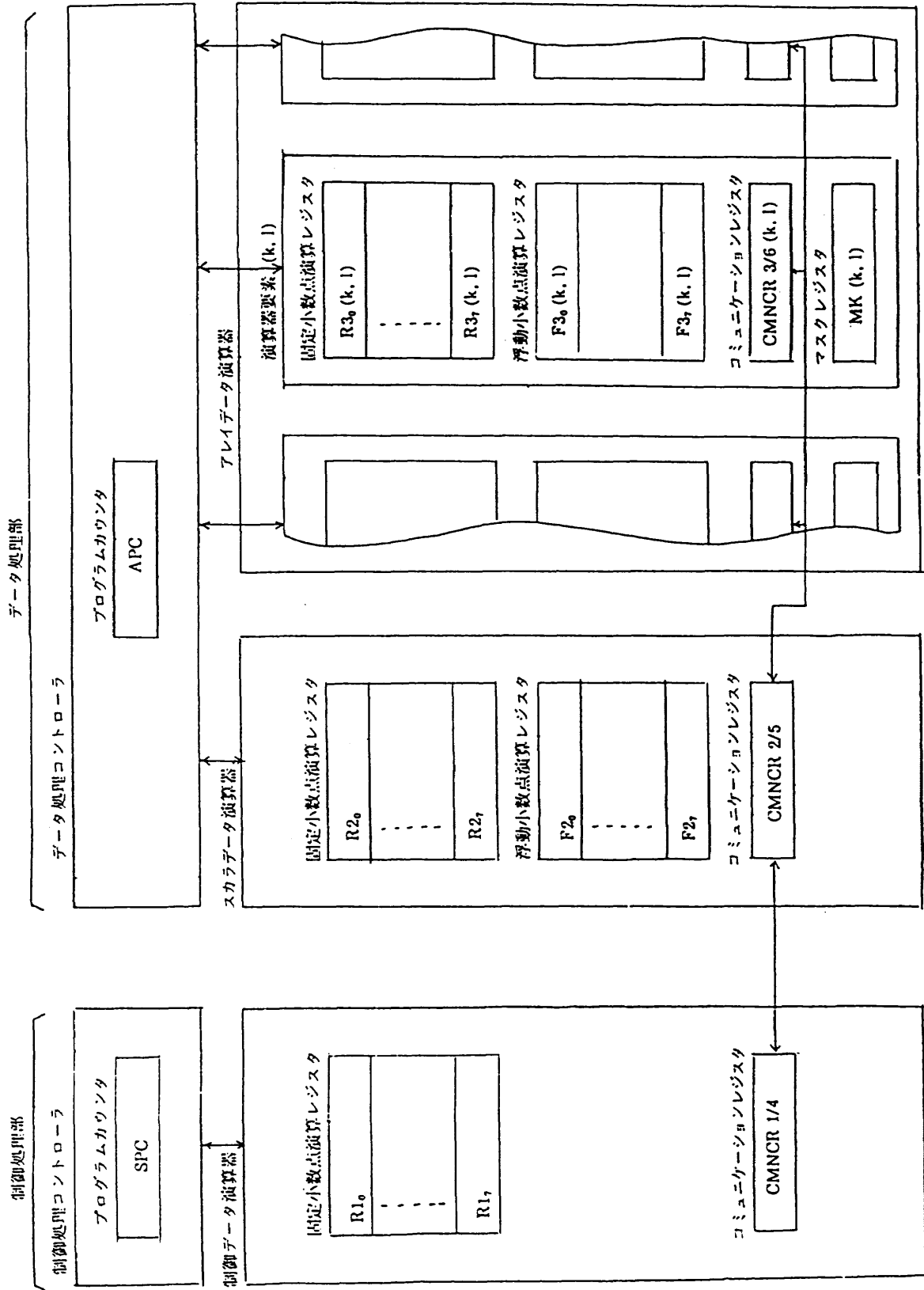


図 5 制御処理部 / データ処理部のレジスタ

フトレジスタで終端がリング結合しているトーラス結合網を構成する(図6)。

メモリ要素からデータを読み出すときには各演算器要素 (k, l) よりアドレスを $DMBAR3(k, l)$ に移し以後データ処理コントローラの制御により1クロック毎に行または列方向に1個だけ移動し、所定の位置 (k', l') に到達した後メモリ要素の読み出しがおこなわれる。7クロック後に読み出されたデータは $DMBDR3/6(k', l')$ に入りそれを $DMBDR3/6(k, l)$ まで行および列を移動させ、データを演算器要素 (k, l) に移す。データを書き込む時にはデータおよびアドレスを演算器要素 (k, l) より $DMBDR3/6(k, l)$, $DMBAR3(k, l)$ に移し以後所定の位置 (k', l') まで行列を1個毎移動した後メモリ要素に書き込む。

2.2 命令の実施

並列計算機の命令はインストラクションメモリに格納され各処理部より読み出され解読・実行される。

全命令は64ビットで構成されその実行手順は各命令ごと

に決められており、一例として図7にアレイデータの加算命令(AA)の実行手順を示す。ここで

() はレジスタおよび記憶装置の内容

APC はデータ処理部のプログラムカウンタ

ADADR 2 はスカラデータのアドレス専用の加算器

AIR は命令を保持するレジスタ

ADADR 3 (k, l) はアレイデータのアドレス専用の加算器

R3 (k, l) はアレイデータの固定少数点演算レジスタ

ADDER 3/6 (k, l) はアレイデータ演算器回路

その他は一時的に命令・データ・アドレスを保持するレジスタである。各タイミングの概略は

タイミング0~7 ; 命令を読み出す動作

タイミング8 ; 命令を解読する動作

タイミング9 ; 命令を解読し各演算器に制御信号を伝達する動作

タイミング10 ; オペランド番地の計算

タイミング11 ; オペランド番地の設定と結合網の

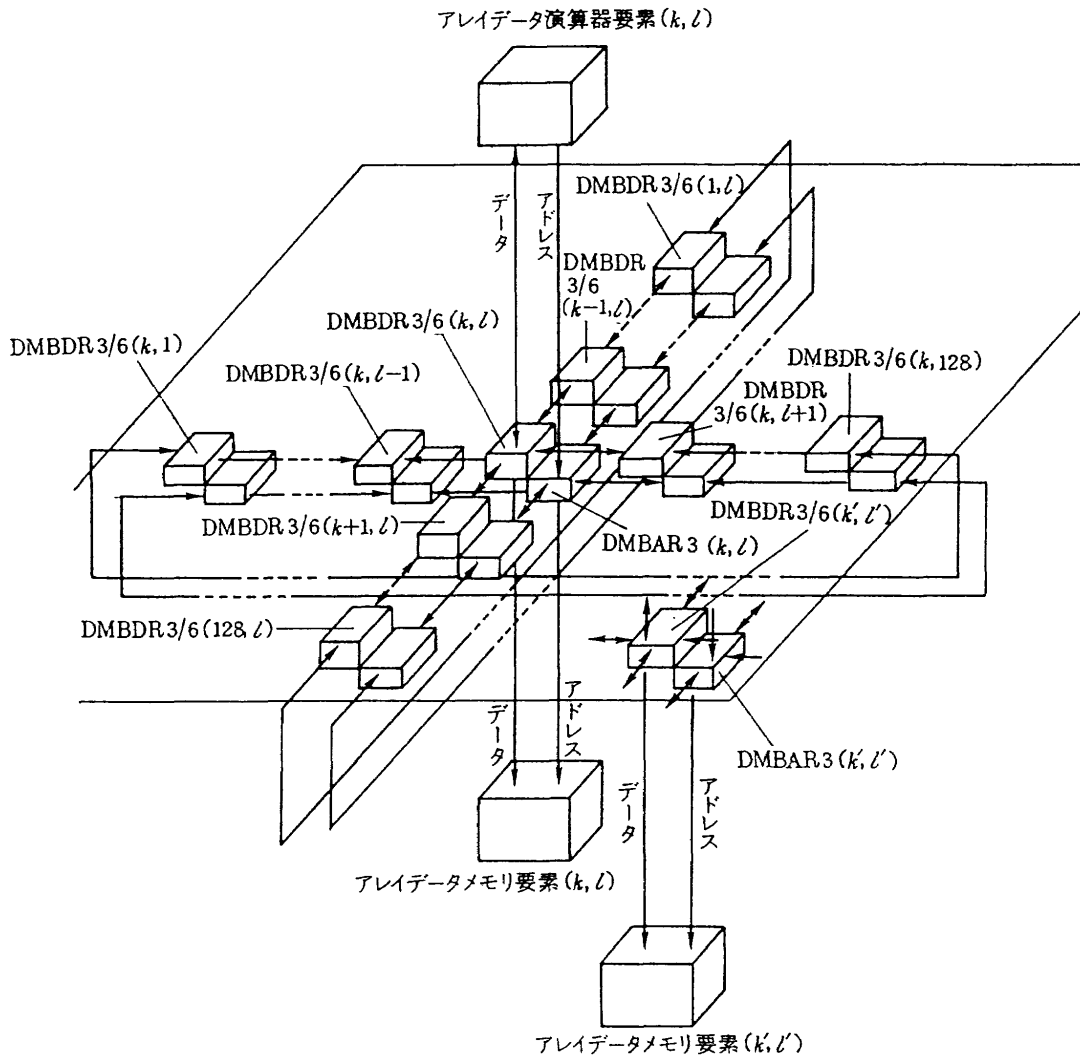


図6 アレイデータメモリコントロールユニットの構成

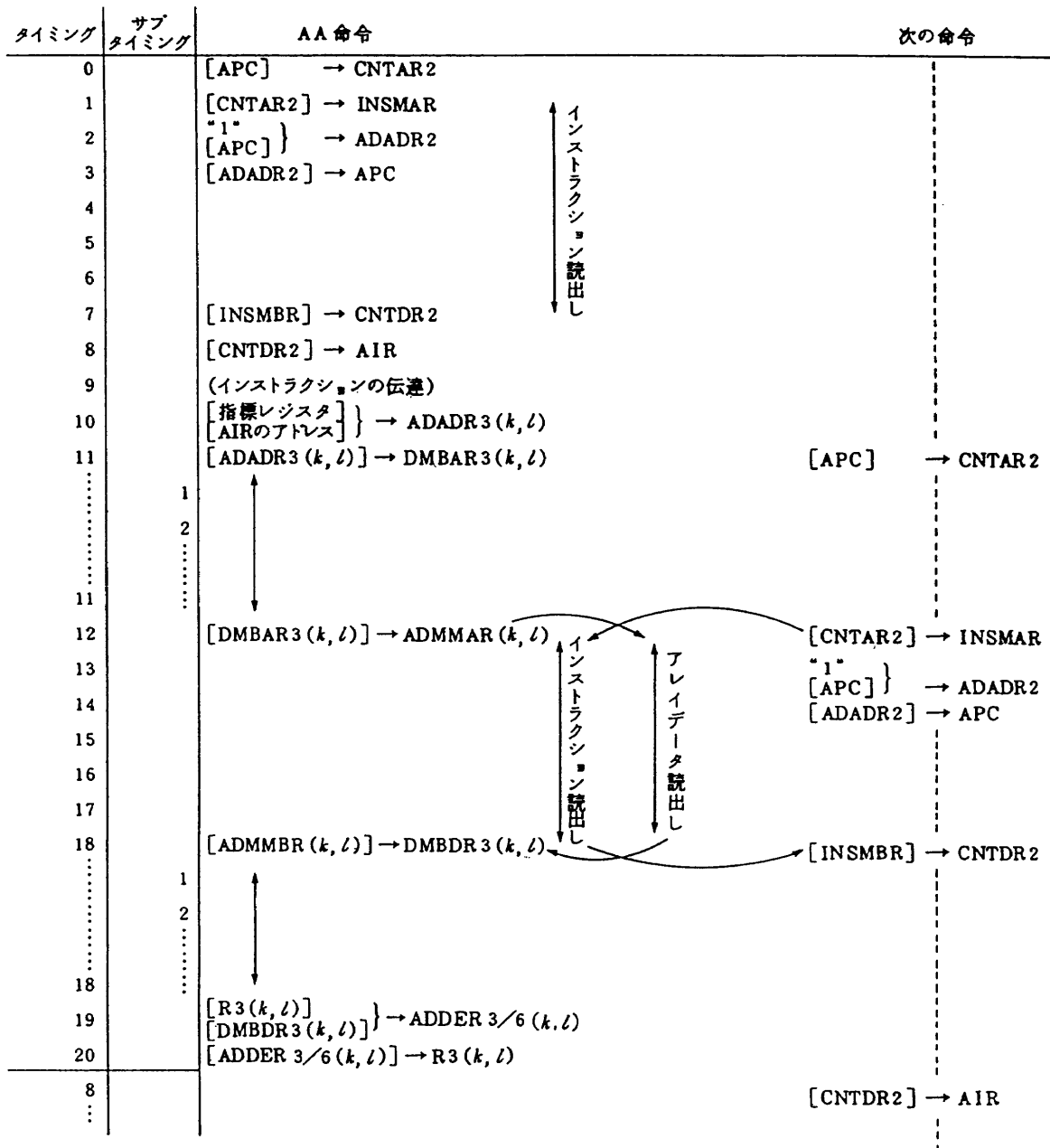


図7 Add Array (AA) 命令のタイムチャート

動作によるメモリ要素の選択

タイミング12～17; オペランドの読み出し

タイミング18 ; オペランドの逆転送

タイミング19～20; 配列演算の実行

を示す。またタイミング11～18で次の命令を先取りする様子も示されている。

さらに、メモリの競合が起きタイミングが進められない時にはメモリが解除されるまで待ちになる。

3. アセンブラとシミュレータ

計算機は機械語の命令しか理解できないため応用プログラムから演算結果を得るためには処理手順をその機械語に変換する必要がある。しかし直接機械語に変換することは多大の労力が必要であり誤りも起き易い。また、高級言語

で記述し直接機械語に翻訳できれば理想ではあるが並列計算機用のコンパイラを開発しなければならず並列計算機の有用性を検証する手続きとしてはあまりにも大ききである。そこで前記二通りの手段の間であるアセンブラを開発した⁵⁾。

このアセンブラは並列計算機のプログラムイメージが図8の構成になることを前提とする。

3.1 アセンブラの動作

このアセンブラの入出力ファイルの関係を図9に示す。

記号命令で書かれた原始プログラムはファイル“VP. ASM1”に格納されている。アセンブラはこれを並列計算機の機械語に変換し、命令列をファイル“VP. OBJ1”に、スカラデータ初期値をファイル“VP. OBJ2”に、アレイデ

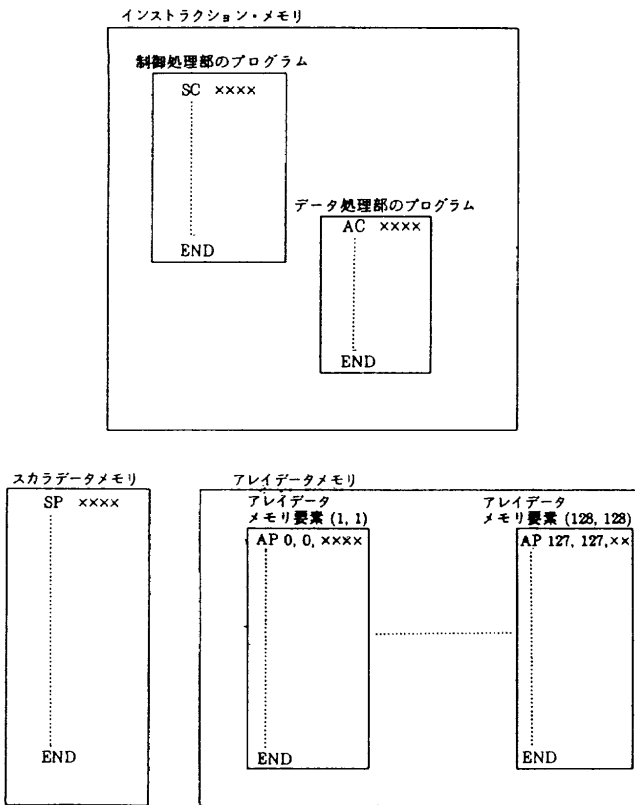


図 8 プログラムイメージ

ータ初期値を各アレイデータメモリ要素ごとにファイル“VP. OBJ3”に区分データセットとして出力する。

これらのファイルは並列計算機の動作を忠実に模擬するシミュレータの入力となる。

3.2 シミュレータの動作

並列計算機の各部動作・効率を検討するため、このシミュレータは命令を実施する手順をクロックごとに忠実に追跡するように作成されており演算結果をも得ることができる⁶⁾。

図10にその入出力ファイルの関係を示す。

シミュレータが起動されると命令列、スカラ/アレイデータの初期値が読み込まれた後、制御処理部が起動される。以後、1クロックずつ時刻を進ませ、それに応じて制御処理部の命令が解読・実行され、演算器・メモリ等間でデータが移動・加工される状況が模擬される。

制御処理部がデータ処理部起動命令を解読するとアドレス部から求めたデータ処理部命令の先頭アドレスをデータ処理部に送り起動する。以後、両処理部は独立して処理を進め、各処理部の命令が解読・実行される。そして各自の命令列に停止命令を解読するとその部を停止し、両処理部

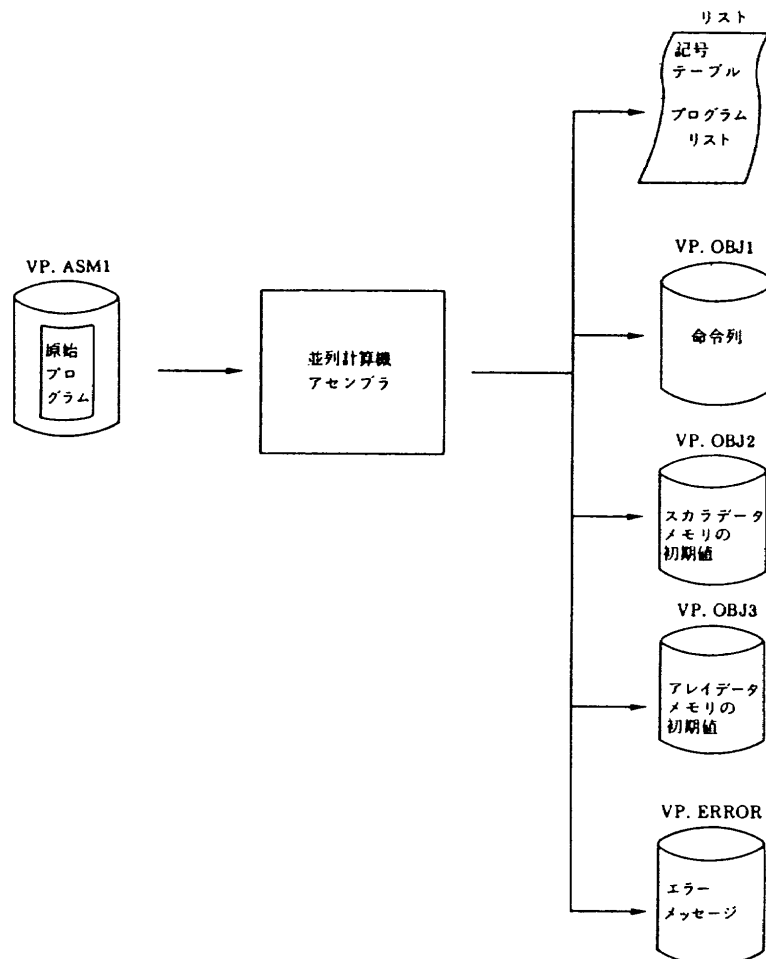


図 9 アセンブラの入出力ファイル

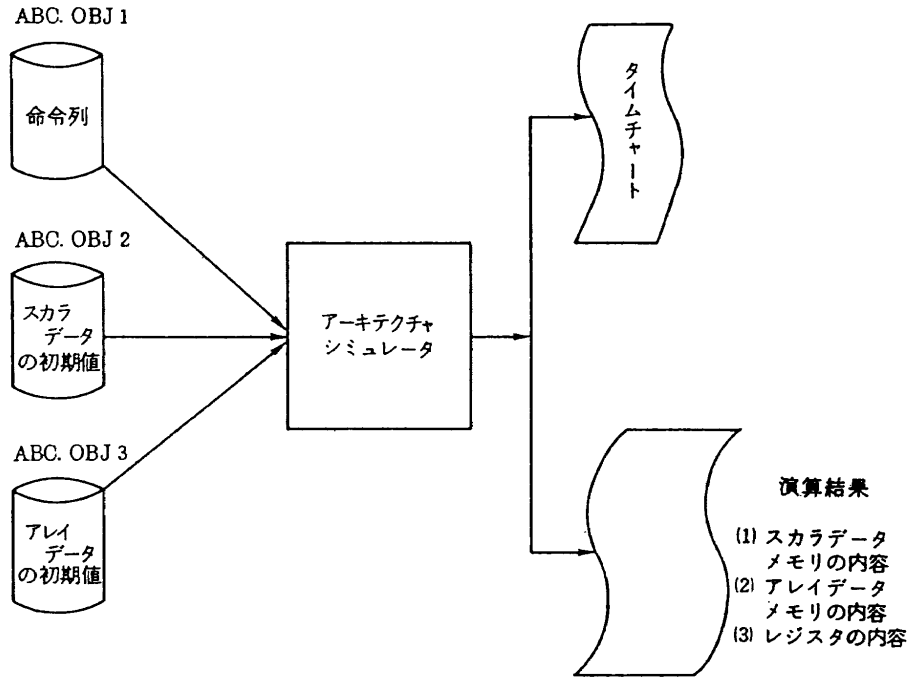


図10 シミュレータの入出力ファイル

が停止した時点で模擬が終了しタイムチャートと演算結果が出力される。

4. 立方体空間内の流れ場解析プログラム

性能を評価するにあたり、三次元非定常非圧縮ナビエ・ストークス方程式による単位立方体空間 (cavity) で上面の流速が1である流れ場解析プログラムを使用する。以後図11で面 ABCD を下面、面 EFGH を上面、面 CGFD を前面、面 AEHB を後面、面 ADFE を左面、面 CBHG を右面と言う。

このプログラムは圧力を p 、流速を x, y, z 方向の流速の要素を u, v, w とした $V(u, v, w)$ 、 t を時刻として式

$$\begin{cases} dV/dt + (V \cdot \nabla) V \\ \quad = -\nabla p + 1/Re \cdot \nabla^2 V \end{cases} \quad (1)$$

$$\nabla V = 0 \quad (2)$$

を境界条件

$$\begin{cases} x = 0, 1 \text{ で } u = v = w = 0 & (3) \\ y = 0 \text{ で } u = v = w = 0 & (4) \\ y = 1 \text{ で } u = 1, v = w = 0 & (5) \\ z = 0, 1 \text{ で } u = v = w = 0 & (6) \end{cases}$$

のもとに差分方程式に変換し、数値計算するものである^{7),8)}。

4.1 解法の概略

時間刻みを Δt とし、 x, y, z 方向を N 等分してその刻み幅を h とすると

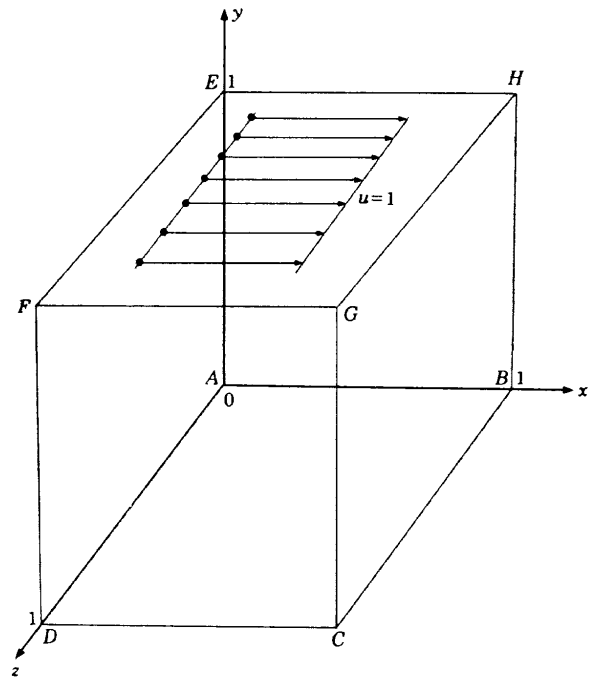


図11 立方体空間図

$$\begin{cases} \Delta x = \Delta y = \Delta z = h = 1/N \\ t = n \cdot \Delta t \\ x = i \cdot h \quad (i = 0 \sim N) \\ y = j \cdot h \quad (j = 0 \sim N) \\ z = k \cdot h \quad (k = 0 \sim N) \end{cases}$$

となり

$$\begin{cases} i = 0 \text{ および } N \\ j = 0 \text{ および } N \\ k = 0 \text{ および } N \end{cases}$$

は境界である。

ここで時刻 t での V, p を V^n, p^n , 時刻 $t+dt$ でのそれらを V^{n+1}, p^{n+1} とすると(1)と(2)の差分方程式は以下の形になる。

$$V^{n+1} + dt \cdot \nabla p^{n+1} = V^* \quad (7)$$

$$V^* = V^n + dt[-(V^n \cdot \nabla) V^n + 1/Re \cdot \nabla^2 V^n] \quad (8)$$

$$\nabla V^{n+1} = 0 \quad (9)$$

ここで V^* は V^n より計算でき、その要素を u^*, v^*, w^* とする。

(7)と(9)より下記の式を得る。

$$\nabla^2 p^{n+1} = 1/dt \cdot \nabla V^* \quad (10)$$

式(10)はポアソン方程式であり、また偶奇数番ごとに8個の系を構成するため p^n を p^* の初期値、 m を反復数、 i, j, k を $1 \sim N-1$ とし

$$p_{i,j,k}^{*(m+1)} = p_{i,j,k}^{*(m)} + DELTA_{i,j,k} \quad (11)$$

$$DELTA_{i,j,k} = Const \cdot [(p_{i+2,j,k}^{*(m)} + p_{i-2,j,k}^{*(m)} + p_{i,j+2,k}^{*(m)} + p_{i,j-2,k}^{*(m)} + p_{i,j,k+2}^{*(m)} + p_{i,j,k-2}^{*(m)})/6 - p_{i,j,k}^{*(m)} - 2h^2 \cdot A_{i,j,k}/3]$$

$$A_{i,j,k} = (u_{i+1,j,k}^* - u_{i-1,j,k}^* + v_{i,j+1,k}^* - v_{i,j-1,k}^* + w_{i,j,k+1}^* - w_{i,j,k-1}^*) / (2h \cdot dt)$$

を反復計算により丸めを 10^{-6} とし収束した $p^{*(m+1)}$ を p^{n+1}

として解いたものである。ここで Const は減速パラメータで 0.95 とする。

このため、 N を奇数とし、各境界面の外側に格子点を設け、 $-1, N+1$ の番号をつける (図12)。この p^{n+1} と(7)より、

$$V^{n+1} = V^* - dt \cdot \nabla p^{n+1} \quad (12)$$

を求める。また、(11)を解くためには境界・外側の格子点で $p^{*(m)}$ を設定しなければならない。それには境界条件から $x=0, 1$ で

$$dp/dx = d^2 u/dx^2 / Re \quad (13)$$

として

$$\begin{cases} p_{0,j,k}^{*(m)} = p_{1,j,k}^{*(m)} - B_{0,j,k}^n \\ B_{0,j,k}^n = (u_{2,j,k}^n - 2u_{1,j,k}^n + u_{0,j,k}^n) / (Re \cdot h) \\ p_{N,j,k}^{*(m)} = p_{N-1,j,k}^{*(m)} + B_{N,j,k}^n \\ B_{N,j,k}^n = (u_{N,j,k}^n - 2u_{N-1,j,k}^n + u_{N-2,j,k}^n) / (Re \cdot h) \end{cases}$$

$y=0, 1$ で

$$dp/dy = d^2 v/dy^2 / Re \quad (14)$$

として

$$\begin{cases} p_{i,0,k}^{*(m)} = p_{i,1,k}^{*(m)} - B_{i,0,k}^n \\ B_{i,0,k}^n = (v_{i,2,k}^n - 2v_{i,1,k}^n + v_{i,0,k}^n) / (Re \cdot h) \\ p_{i,N,k}^{*(m)} = p_{i,N-1,k}^{*(m)} + B_{i,N,k}^n \\ B_{i,N,k}^n = (v_{i,N,k}^n - 2v_{i,N-1,k}^n + v_{i,N-2,k}^n) / (Re \cdot h) \end{cases}$$

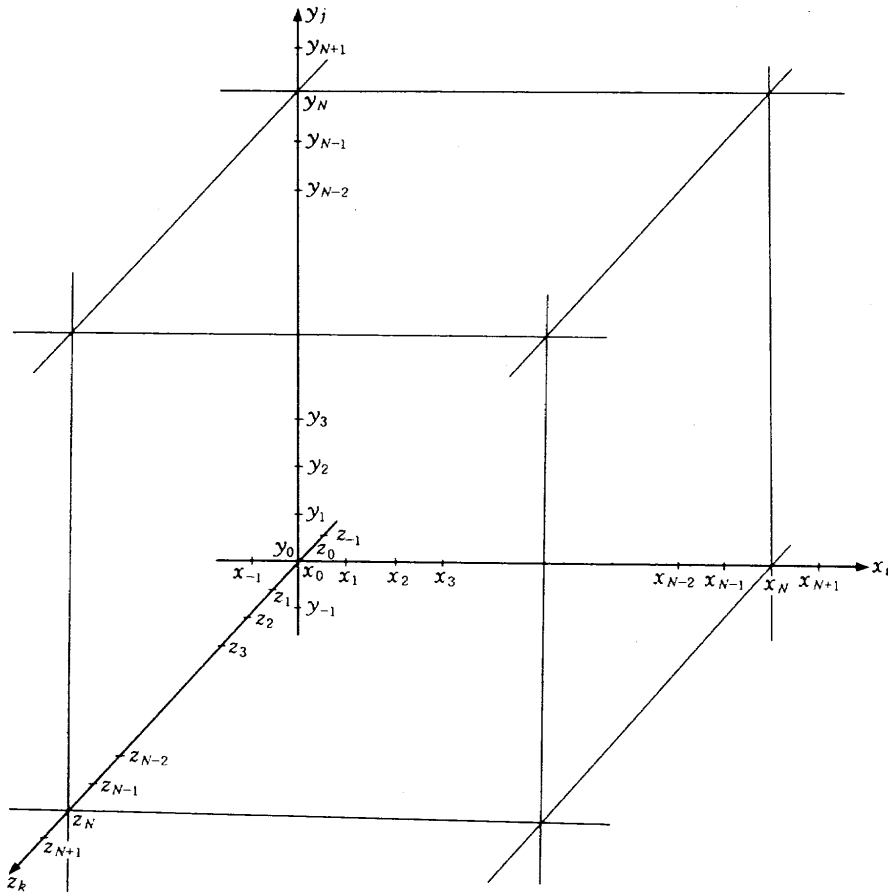


図12 格子点の番号

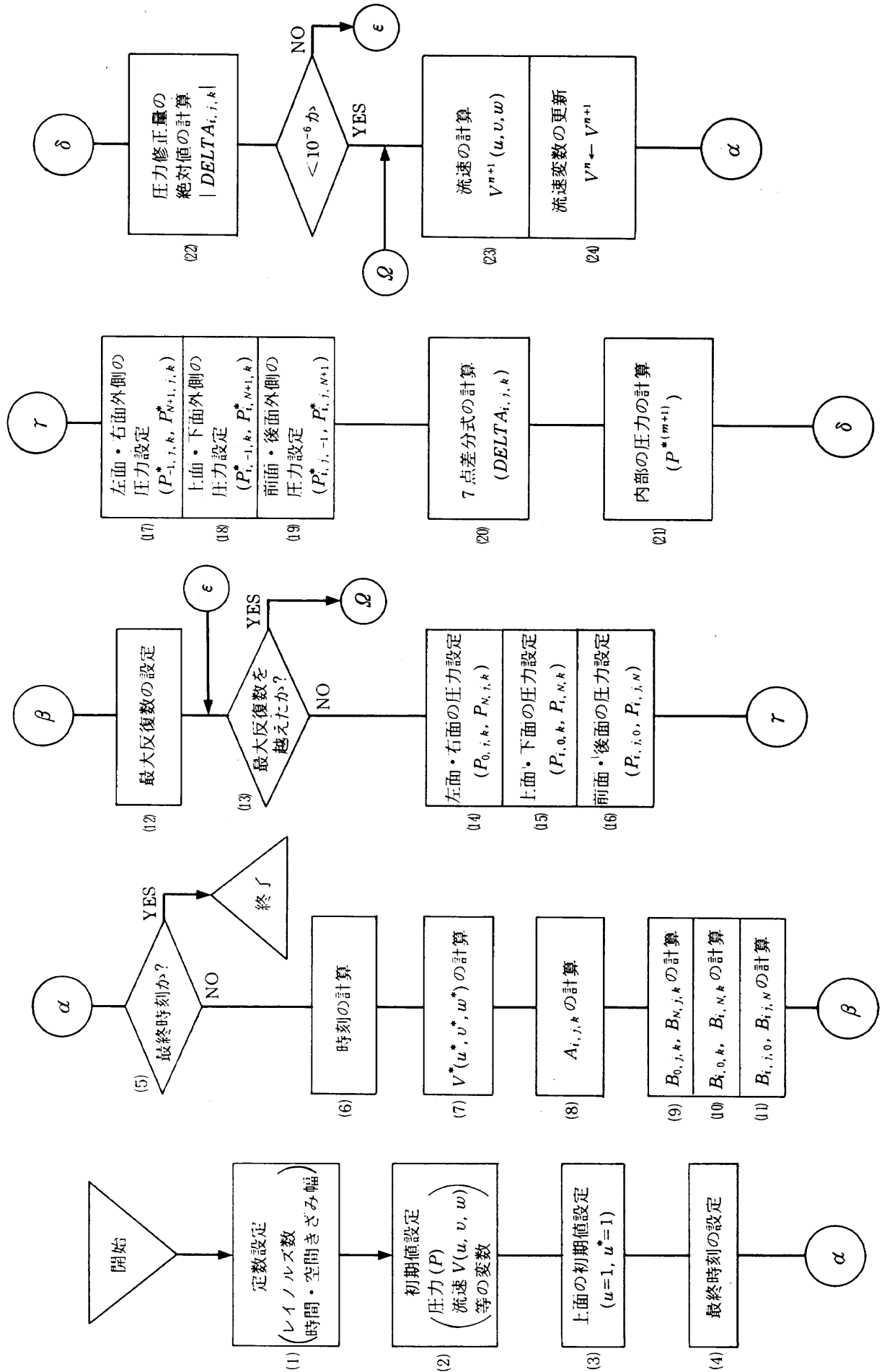


図13 フローチャート

また, $z=0, 1$ で

$$dp/dy = d^2 v/dy^2 / Re \tag{15}$$

として

$$\begin{cases} p_{i,j,0}^{*(m)} = p_{i,j,1}^{*(m)} - B_{i,j,0}^n \\ B_{i,j,0}^n = (w_{i,j,2}^n - 2w_{i,j,1}^n + w_{i,j,0}^n) / (Re \cdot h) \\ p_{i,j,N}^{*(m)} = p_{i,j,N-1}^{*(m)} + B_{i,j,N}^n \\ B_{i,j,N}^n = (w_{i,j,N}^n - 2w_{i,j,N-1}^n + w_{i,j,N-2}^n) / (Re \cdot h) \end{cases}$$

を用いる。ここで i, j, k は $1 \sim N-1$ である。また, 外側の格子点について V^* の境界条件を V^{n+1} のそれに等しいとして,

$$\begin{cases} u_{0,j,k}^* = u_{0,j,k}^{n+1} = 0 \\ u_{N,j,k}^* = u_{N,j,k}^{n+1} = 0 \\ v_{i,0,k}^* = v_{i,0,k}^{n+1} = 0 \\ v_{i,N,k}^* = v_{i,N,k}^{n+1} = 0 \\ w_{i,j,0}^* = w_{i,j,0}^{n+1} = 0 \\ w_{i,j,N}^* = w_{i,j,N}^{n+1} = 0 \end{cases}$$

を(12)に用いて

$$\begin{cases} p_{-1,j,k}^{*(m)} = p_{1,j,k}^{*(m)} \\ p_{N+1,j,k}^{*(m)} = p_{N-1,j,k}^{*(m)} \\ p_{i,-1,k}^{*(m)} = p_{i,1,k}^{*(m)} \\ p_{i,N+1,k}^{*(m)} = p_{i,N-1,k}^{*(m)} \\ p_{i,j,-1}^{*(m)} = p_{i,j,1}^{*(m)} \\ p_{i,j,N+1}^{*(m)} = p_{i,j,N-1}^{*(m)} \end{cases}$$

を得て設定する。ここで i, j, k は $1 \sim N-1$ である。

4.2 フローチャート

図13にプログラムの開始により初期値の設定を示す。

- ① プログラムの開始により初期値の設定を行う。
..... (1), (2), (3), (4)
- ② 時刻 t を Δt 進める。 (5), (6)
- ③ V^n より V^* を計算する。 V^* の境界は V^n と同じとする。
..... (7)
- ④ V^* より ∇V^* を計算する。 (8)
- ⑤ 境界条件より壁面圧力 $p^{*(m)}$ の修正量 B^n を計算する。
..... (9), (10), (11)
- ⑥ ポアソン方程式を反復計算する最大数を設定する。
..... (12), (13)
- ⑦ 表面圧力 $p^{*(m)}$ を⑤の修正量 B^n を用い計算設定する。
..... (14), (15), (16)
- ⑧ 外側に想定した格子点の圧力 $p^{*(m)}$ を計算設定する。
..... (17), (18), (19)
- ⑨ 7点差分の計算式より内部格子点 $p^{*(m)}$ の修正量 $DELTA$ を計算する。 (20)
- ⑩ 内部点の $p^{*(m)}$ を計算する。 (21)
- ⑪ 圧力 $p^{*(m)}$ の修正量 $DELTA$ の絶対値が 10^{-6} 以下かを判定し, 以上であれば⑦に戻り繰り返す。以下であれば $p^{*(m)}$ を p^{n+1} とする。 (22)
- ⑫ V^{n+1} を計算し, V^n を更新する。 (23), (24)
- ⑬ 以後②に戻り繰り返す。

これにより解いた格子数とレイノルズ数および Δt との関係を表1に示す。表中×印は解がえられなかった場合で, △は解が奇数番と偶数番で不自然と思われる場合, ○は解

表1 レイノルズ数とメッシュ数

レイノルズ数 メッシュ数	10	100	400	1000	3200
13 ³	$\Delta t = 0.0150$ ○	$\Delta t = 0.0150$ △	$\Delta t = 0.0150$ △		
25 ³		○	$\Delta t = 0.0150$ △		
43 ³			$\Delta t = 0.0150$ ○	$\Delta t = 0.0150$ △	
63 ³				$\Delta t = 0.0075$ ○	X
83 ³				$\Delta t = 0.0075$ ○	$\Delta t = 0.0025$ △
127 ³					$\Delta t = 0.00125$ ○

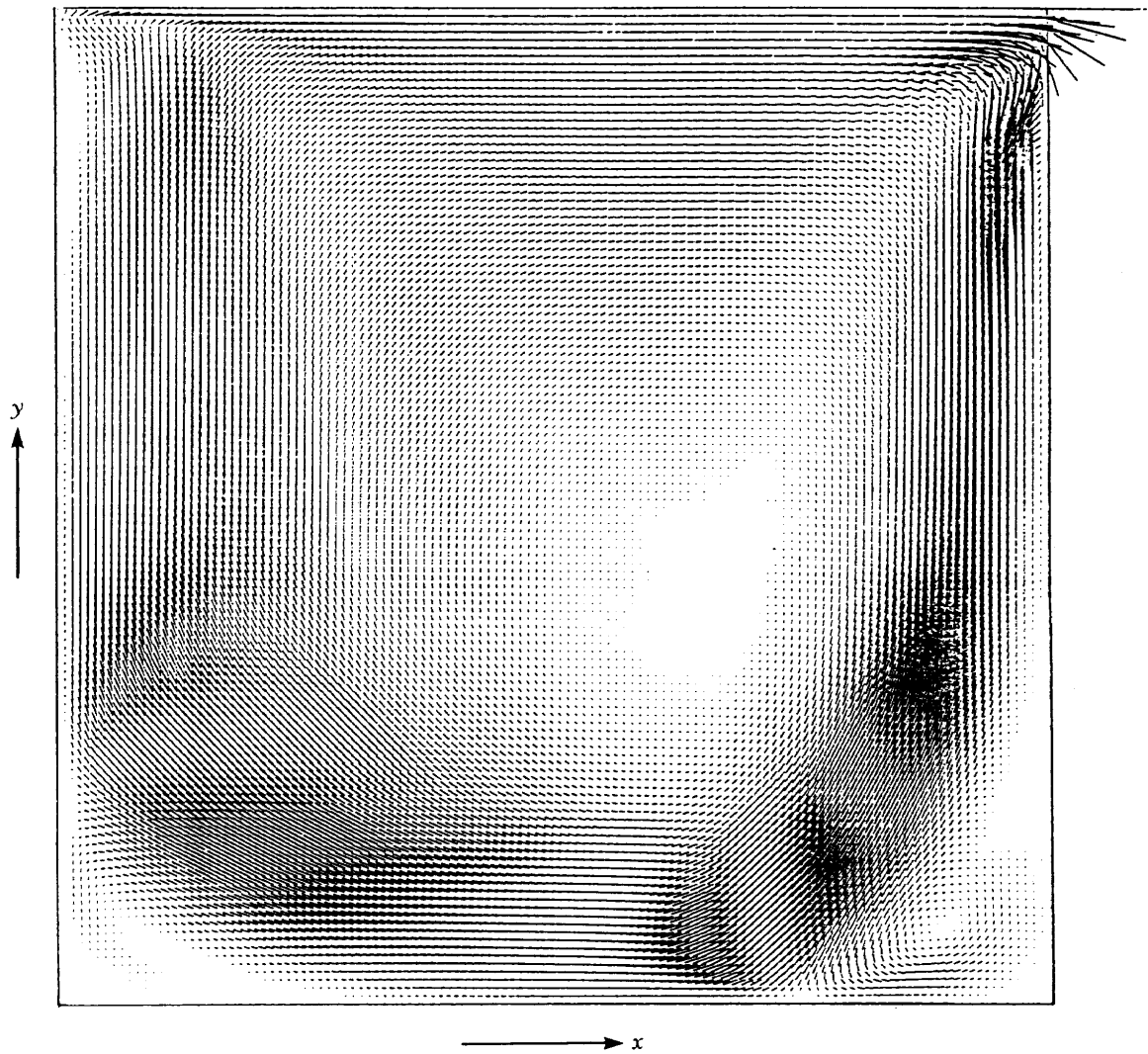


図14 計算例 ($z=0.5$ の面)

が得られた場合である。

その計算例を図14に示す。ここで

$$Re=3200, h=1/125, \Delta t=0.00125, t=125.00$$

である。なお、 $p^{*(m)}$ は相対値で求まるため全格子点で補正を要するが実際の計算では反復数が100,000回以上でもあまり変動しなかったので省略する。

5. 並列計算機への移植と模擬

5.1 移植

並列計算機に流れ場解析プログラムを移植するために以下の方針を建てた。

(1) 流れ場解析プログラムをフローチャートに示すように初期値設定・圧力計算・流速計算・境界条件の設定等の処理単位に分割し、スカラデータ演算器による逐次処理と128×128台のアレイデータ演算器による並列処理との両方を移植し、その各々について処理時間を比べる(表2)。

(2) フォートランからアセンブラへの変換は以下の方法でおこなった。

- ① 出来るだけ簡単な翻訳方法を採用最適化の手法を用いない。これについては次節で再度述べる。
 - ② 変数名をアセンブラのラベルとして用いる。
 - ③ DO ループの翻訳では始めに反復数の初期設定をし、検査してからループ内の計算に入り反復数を数えて再び検査部に戻る。
 - ④ DO ループが重なった時は③をその都度適用する。
- (3) 立方体を x, y, z 方向に各々125等分し位置の番号 $-1 \sim N+1$ を変数の指標の1~127に対応させる(図15)。図16に並列処理で7点差分の計算部の変換例を示す。図中'MASK1'は図17に示すマスクである。図中'0'はマスクの設定を表し、それにより演算が抑制されることを意味する。
- (4) 並列処理独特の問題には以下のものがある。
- ① 格子点と演算器の対応
格子点と演算器・記憶装置との対応は並列処理に適合するための基本条件であり、単に格子の分解能ばかりでなく反復法・反復数に与える影響が大きい。
 x, z 平面に128×128台の演算器・記憶装置を並べ各記憶

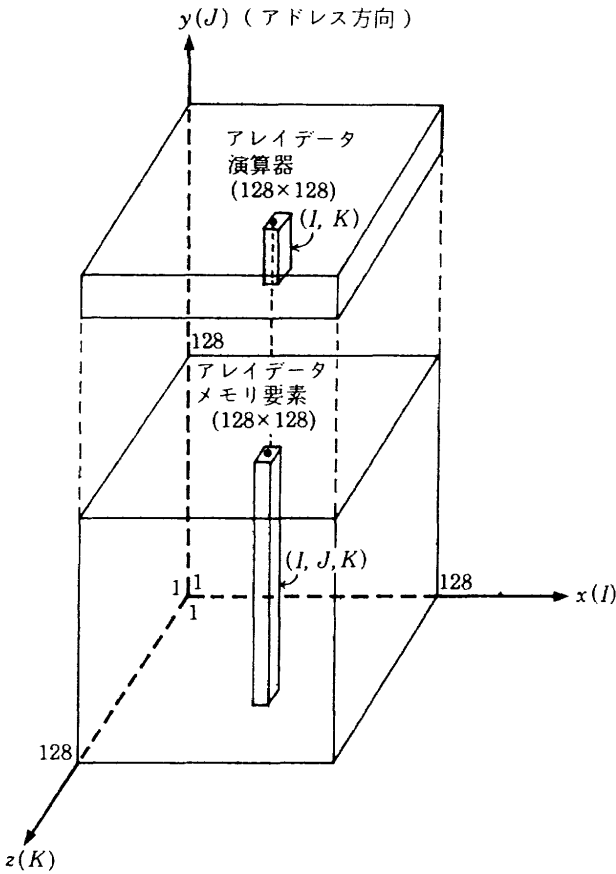


図15 変数と演算器の対応

装置には y 方向のデータを配置する。格子数は 127×127 であるため、 $128 \times 128 - 127 \times 127$ 台の演算器は使用されない。

並列処理の反復計算法としてヤコビ法を用いる。図18にヤコビ法によるこの解析プログラム計算時のポアソン方程式の反復計算数を示す。ヤコビ法では初期の反復数が極端に多い。しかし以後、急激に減少する。 $t=46.7$ 以降は10回、9回、1回の反復数が循環して現れ、その平均は6~7回である。

② 並列処理とマスク操作

ポアソン方程式の反復計算時における各格子点での変数の修正量が丸め誤差以内であるかを判定する場合、逐次処理では各点の修正量を条件と比較し、収束でなければただちに反復計算にもどるのに対し、並列処理では 127×127 格子点の収束判定が同時に可能であるが反復計算を続行するかどうかを判定するためには各演算器の判定信号を一カ所で監視・判定する必要がある。

図19にその回路の概念を示す。ここでは各演算器にマスク $MK(k, l)$ を設定しておき収束条件を満たさない時には払う。マスクが払われた演算器の $CMNCR3/6(k, l)$ の内容は $CMNCR2/5$ に送られるので、その内容が“空”であれば全て収束条件をみたしていることになる⁹⁾。



図16 プログラム変換例

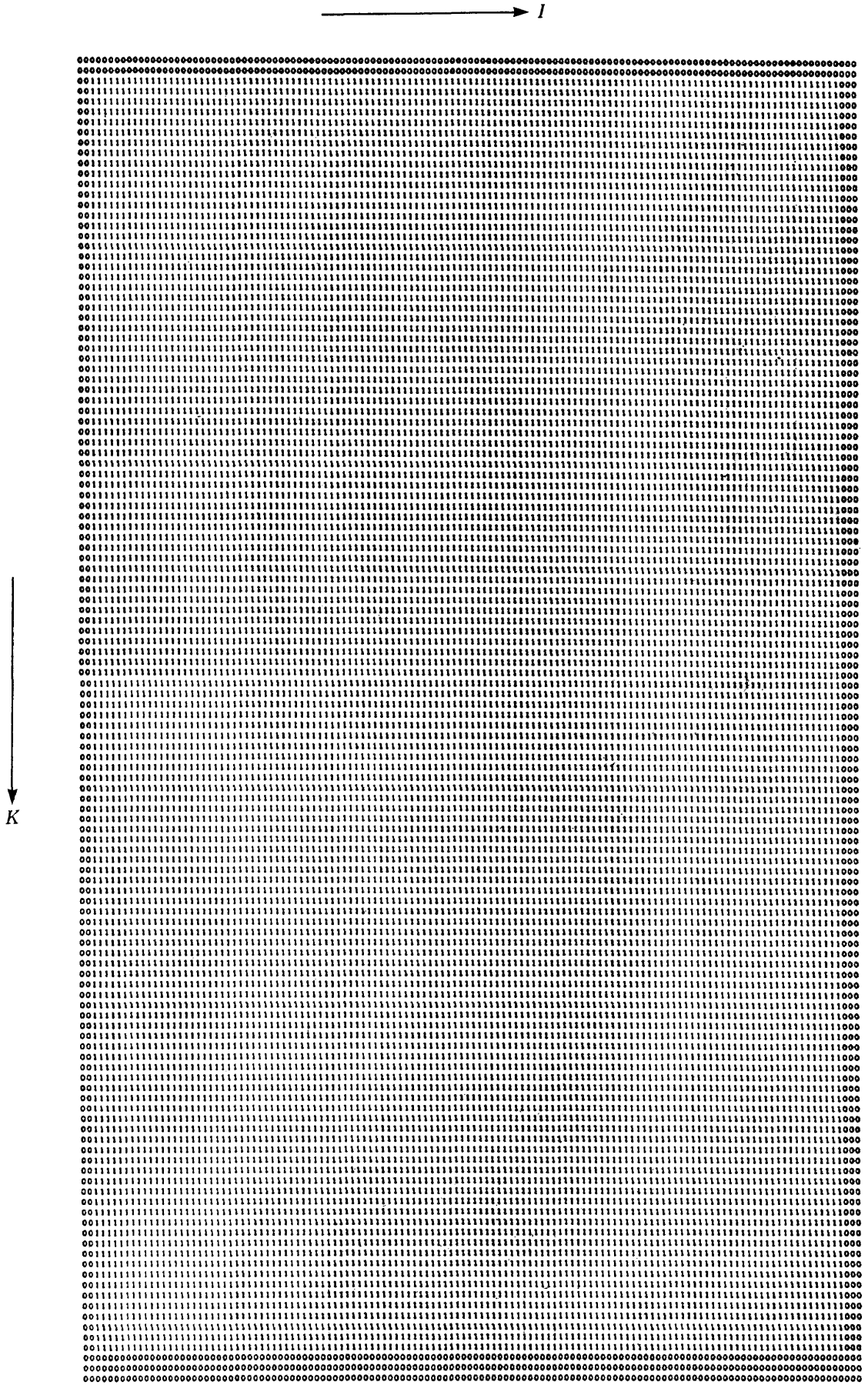


図17 内部格子点と上下面の格子点および上下面の
外側の格子点の演算マスク (J=1~127)

③ 境界条件とマスク操作

この解析プログラムでは、流速・圧力の計算時に上下前後左右の格子点の値を用いるため境界では流速の値を更新しない。またポアソン方程式で圧力を計算する時には内部格子点、表面の格子点と外側の格子点では計算手続きが異なり処理を拒否するマスク操作が必要である。

5.2 模擬

一台の演算器による逐次処理と128×128台の演算器による並列処理との相対的処理効率を見極めることを目標とし、解析プログラムの全体を各処理手続きごとに分割しシミュレータで模擬した時のクロック数を1台の演算器のスカラデータ演算器による逐次処理の場合と128×128台の演算器

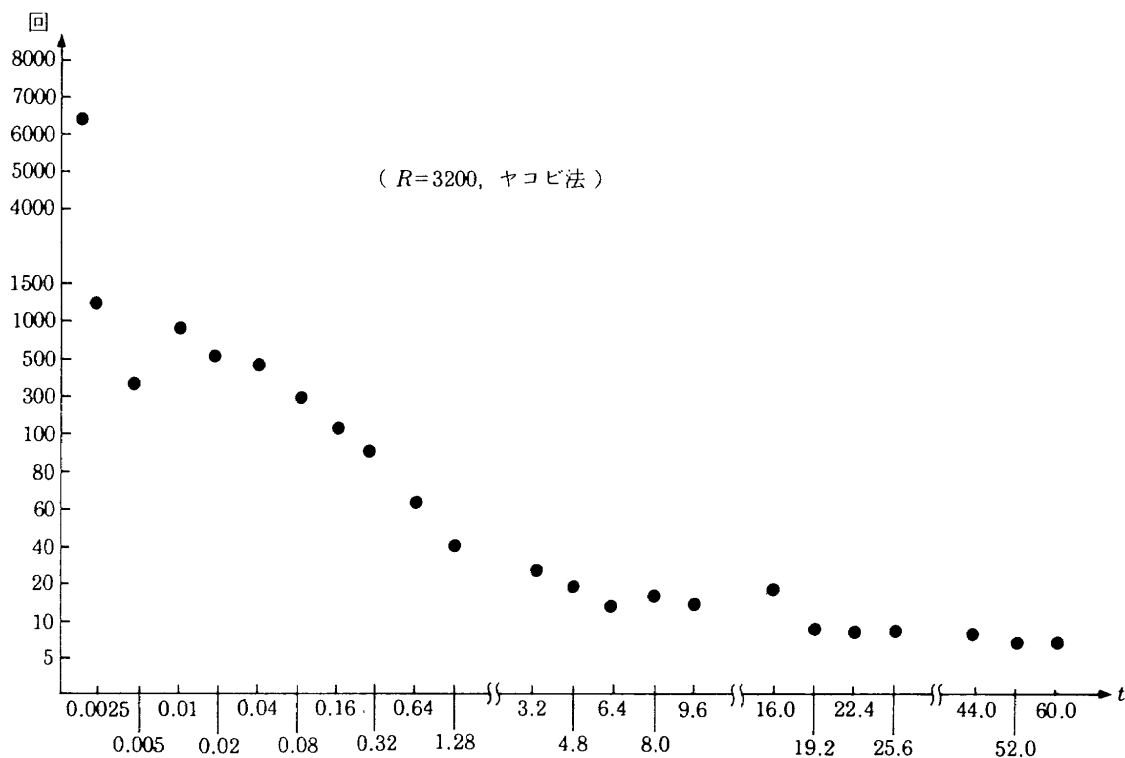
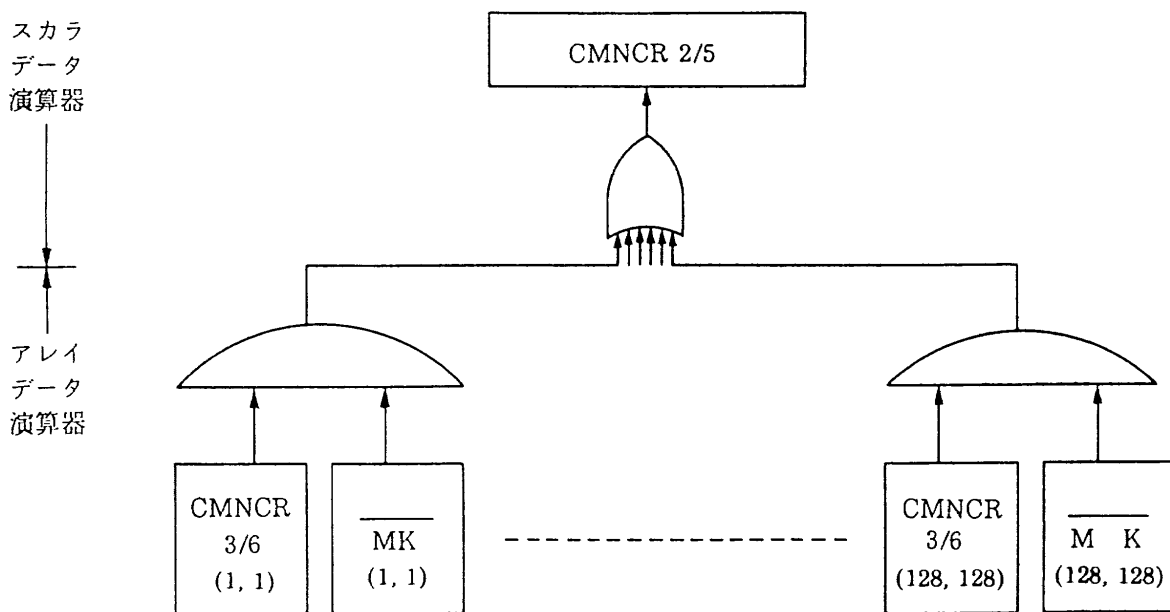


図18 ポアソン方程式の計算での反復回数



CMNCR 2/5, CMNCR 3/6 ; コミュニケーションレジスタ
MK ; マスクレジスタ

図19 収束判定回路

による並列処理の場合とで比較した値を表2に示す。

以下に模擬の結果について考察を加える。

(1) プログラムの各ルーチンでの効率は演算器の数に見合う部分と極端に低下する部分に分かれる。

効率が極端に低いところは表面・外側の格子点の計算部分の中で左右前後の境界面の格子点およびその外側格子点での計算ルーチンである。この時使用するマスクは図20～図21のようになり、マスクによる演算器の抑制のため大部分の演算器は動作していない(図中 '0' はマスクの設定を

表す)。

また、効率の高い部分は内部の格子点の計算および上下面の格子点とその外側の格子点での計算である。ここでは図17に示すようにマスクによる演算の抑制は少ない。

(2) 演算器の数以上に効率の良いルーチンがある。

その部分のプログラムを観察すると逐次処理では演算そのもの以外にループの計算およびアドレスの計算がかなりの計算時間を要しているのに対し並列処理ではループも一重でアドレス計算も少ない。このため逐次処理に比べ有利

表2 処理時間の比較

	並列処理	逐次処理	比
初期設定			
(1)定数設定	619	593	1
(2)初期値設定	24,484	707,277,361	28,887
(3)上面の初期値設定	805	2,543,767	3,159
(4)最終時刻の設定	44	42	1
時刻を Δt 進める計算			
(5)最終時刻の判定	68	68	1
(6)時刻の計算	74	69	1
(7) V^* の計算	223,135	5,557,723,252	24,907
(8) A の計算	34,553	629,890,243	18,229
(9) $B_{0,j,k}, B_{N,j,k}$ の計算	42,425	7,182,014	169
(10) $B_{i,0,k}, B_{i,N,j}$ の計算	225	7,182,014	31,920
(11) $B_{i,j,0}, B_{i,j,N}$ の計算	42,425	7,182,014	169
(12)最大反復数の設定	21	21	1
ポアソン方程式の計算			
(13)反復数の判定	68	68	1
(14)左面・右面の圧力設定	26,435	3,332,347	126
(15)上面・下面の圧力設定	107	3,332,347	30,571
(16)前面・後面の圧力設定	26,435	3,332,347	126
(17)左面・右面外部の圧力設定	22,253	2,407,142	108
(18)上面・下面外部の圧力設定	69	2,407,142	32,974
(19)前面・後面外部の圧力設定	22,253	2,407,142	108
(20)7点差分式の計算	55,978	1,668,606,365	22,662
(21)内部圧力の計算	15,119	245,341,321	16,227
(22)圧力修正量の絶対値の計算および丸めの判定	25,595	345,465,013	13,497
(23)流速の計算	53,395	1,322,389,960	24,766
(24)流速変数の更新	18,440	380,776,129	20,649
(25)終了時の処理	79	79	1

(ポアソン方程式の計算での比較 $2,276,631,234/194,312=11,716$)
 (時刻を Δt 進める計算での比較 $10,188,957,018/609,073=16,728$)

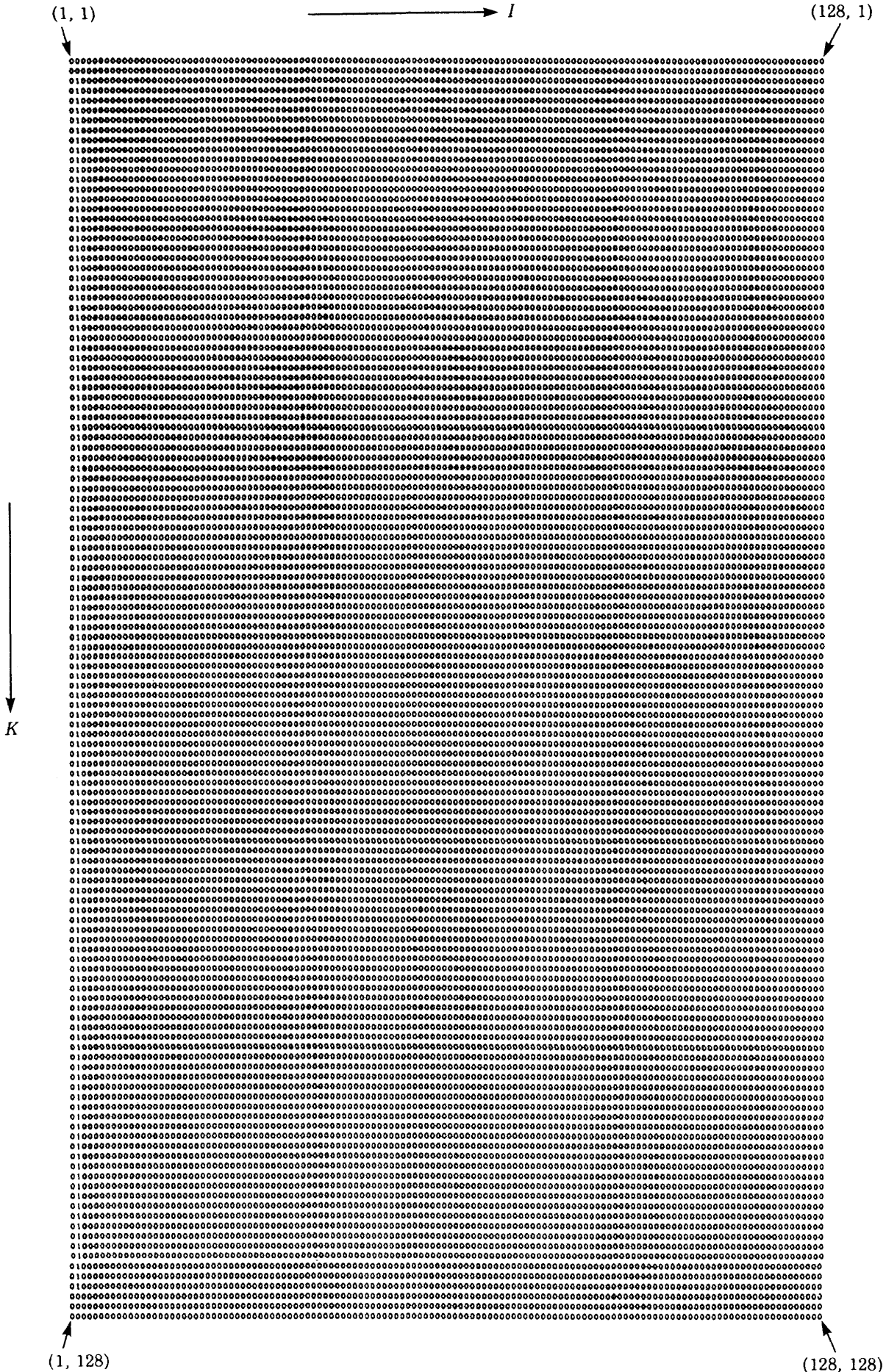


図20(1) 左面格子点の演算マスク (J = 3 ~ 125)

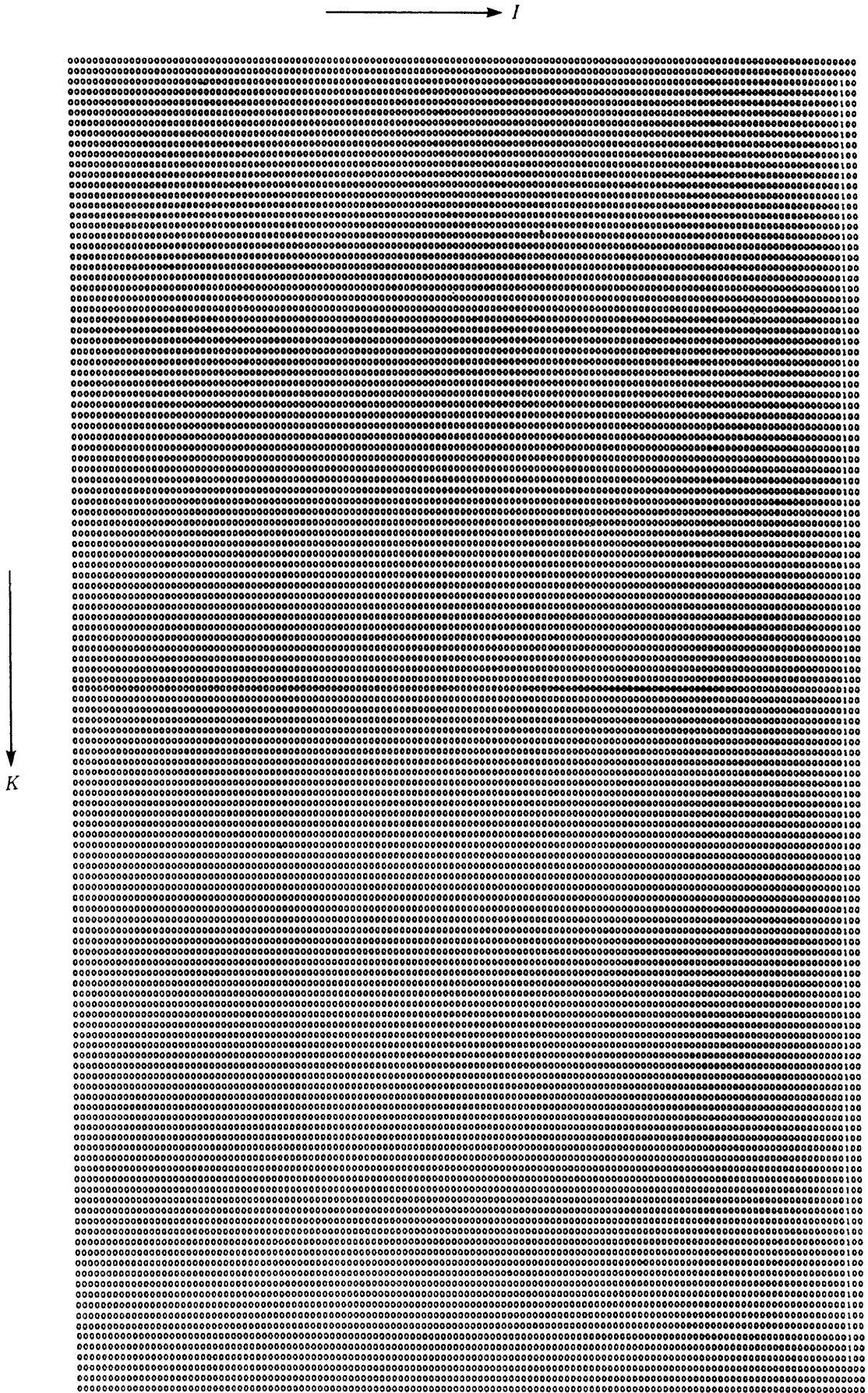


図20(3) 右面格子点の演算マスク (J=3~125)

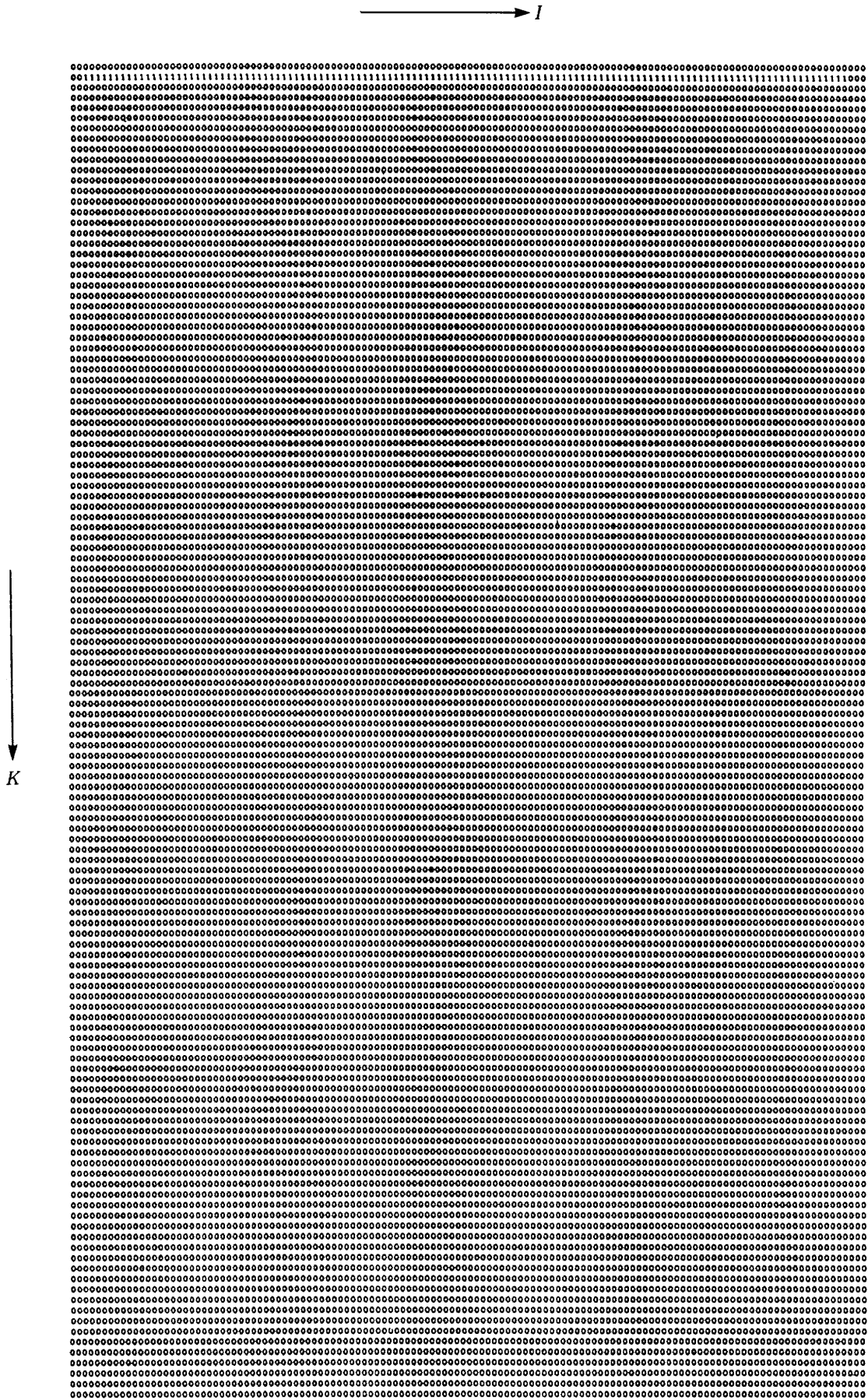


図20(3) 後面格子点の演算マスク (J=3~125)

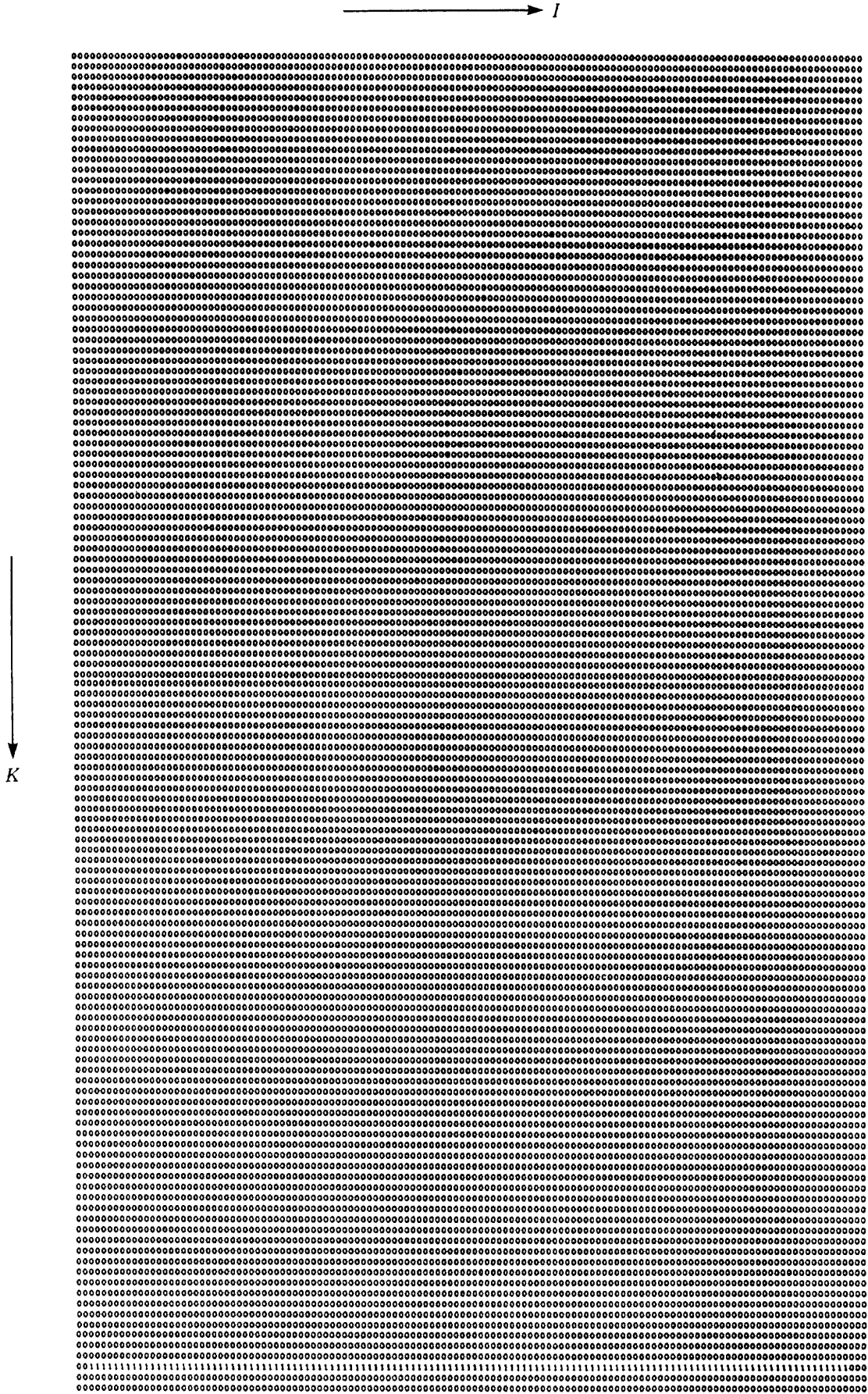


図20(4) 前面格子点の演算マスク (J=3~125)

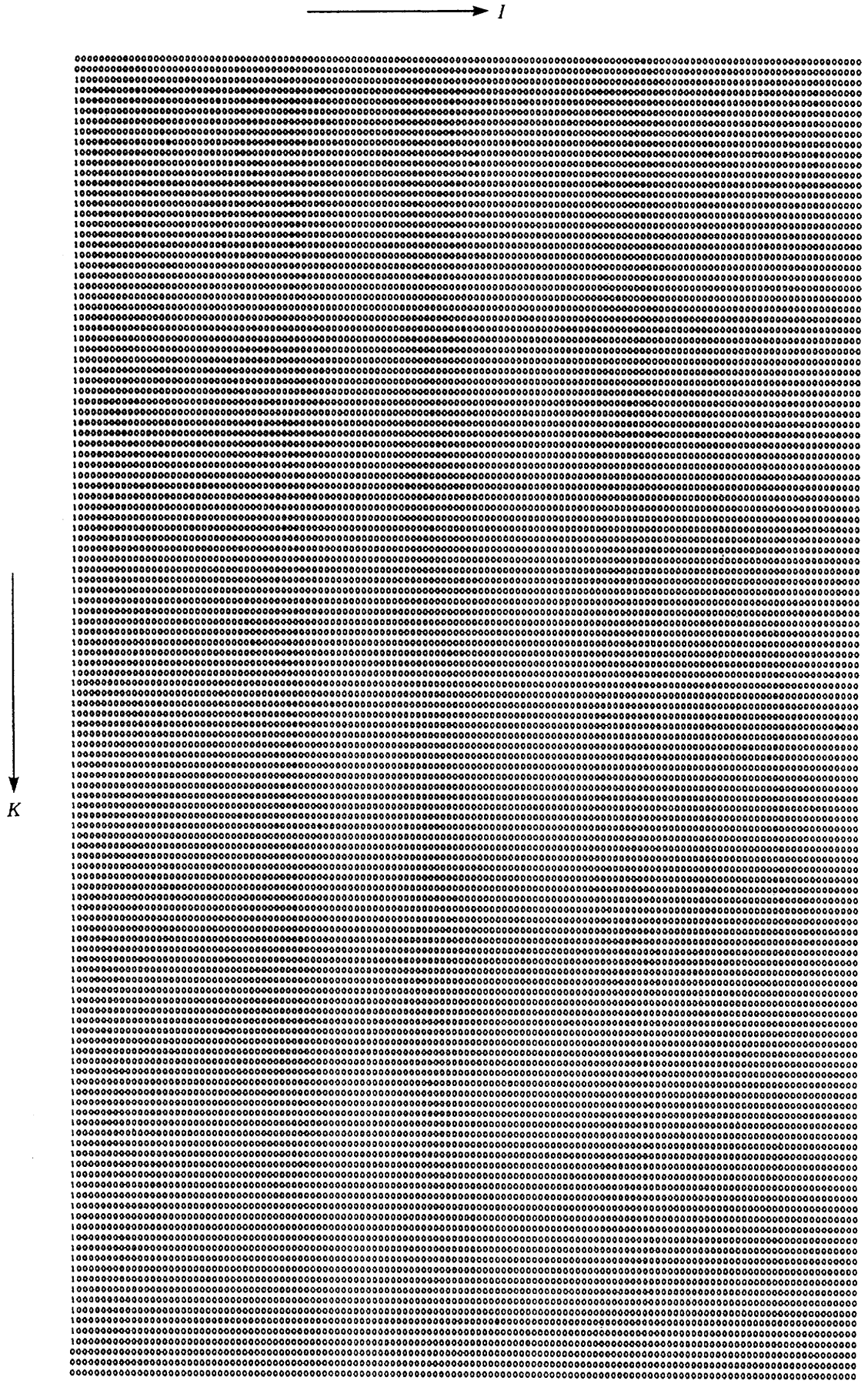


図21(1) 左面の外側格子点の演算マスク (J=3~125)

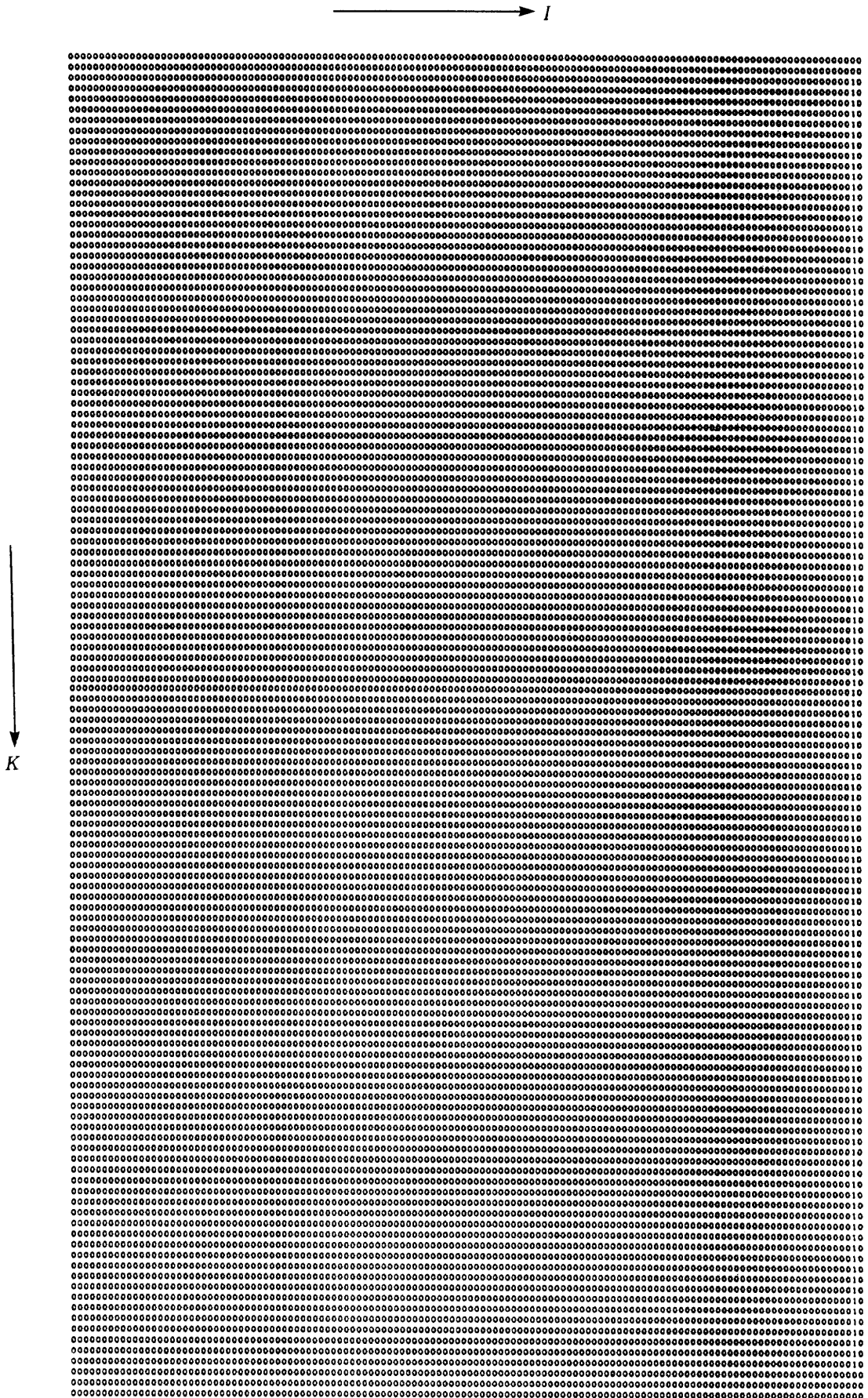


図21(2) 右面の外側格子点の演算マスク (J=3~125)

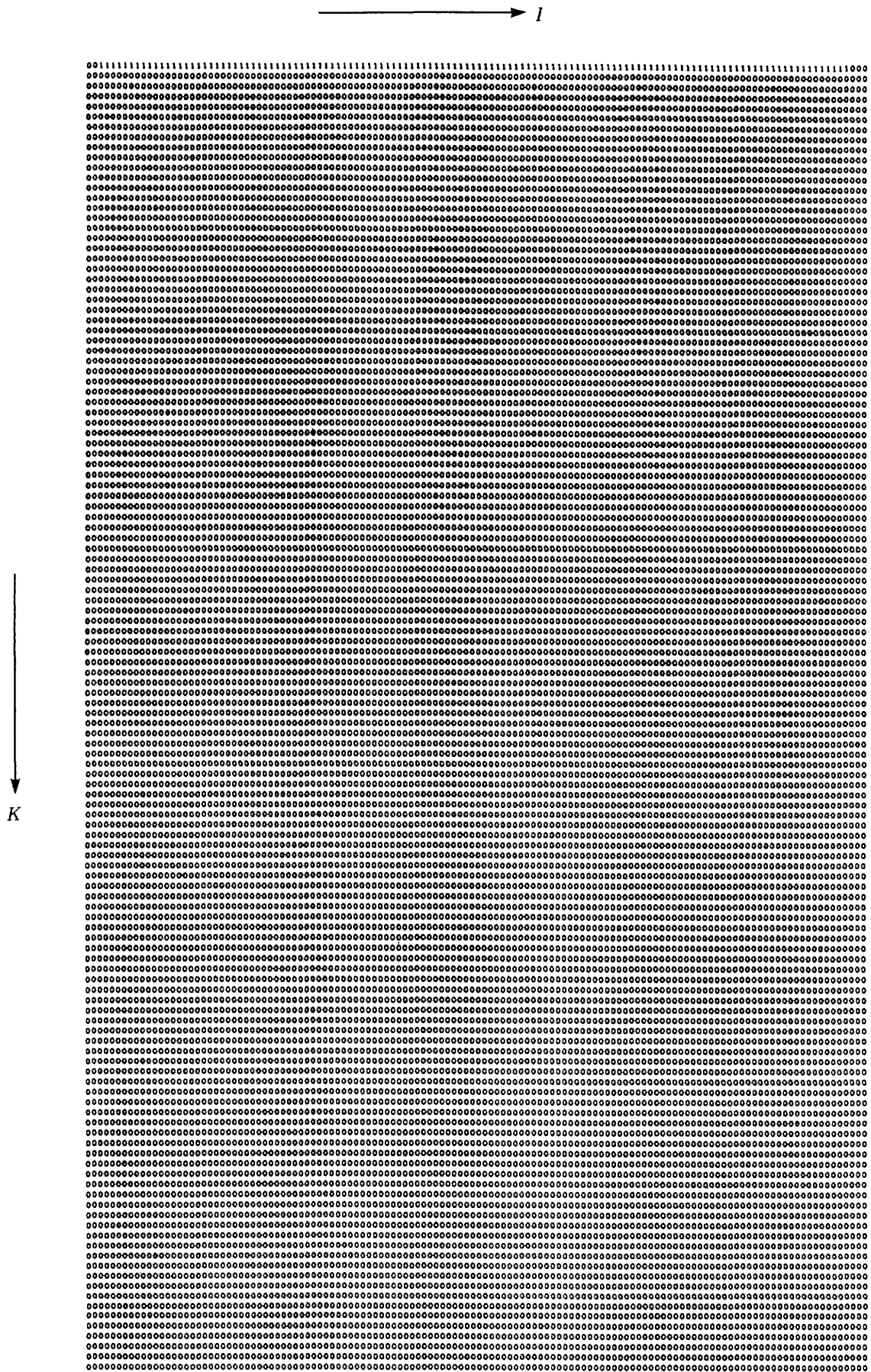


図21(3) 後面の外側格子点の演算マスク (J=3~125)

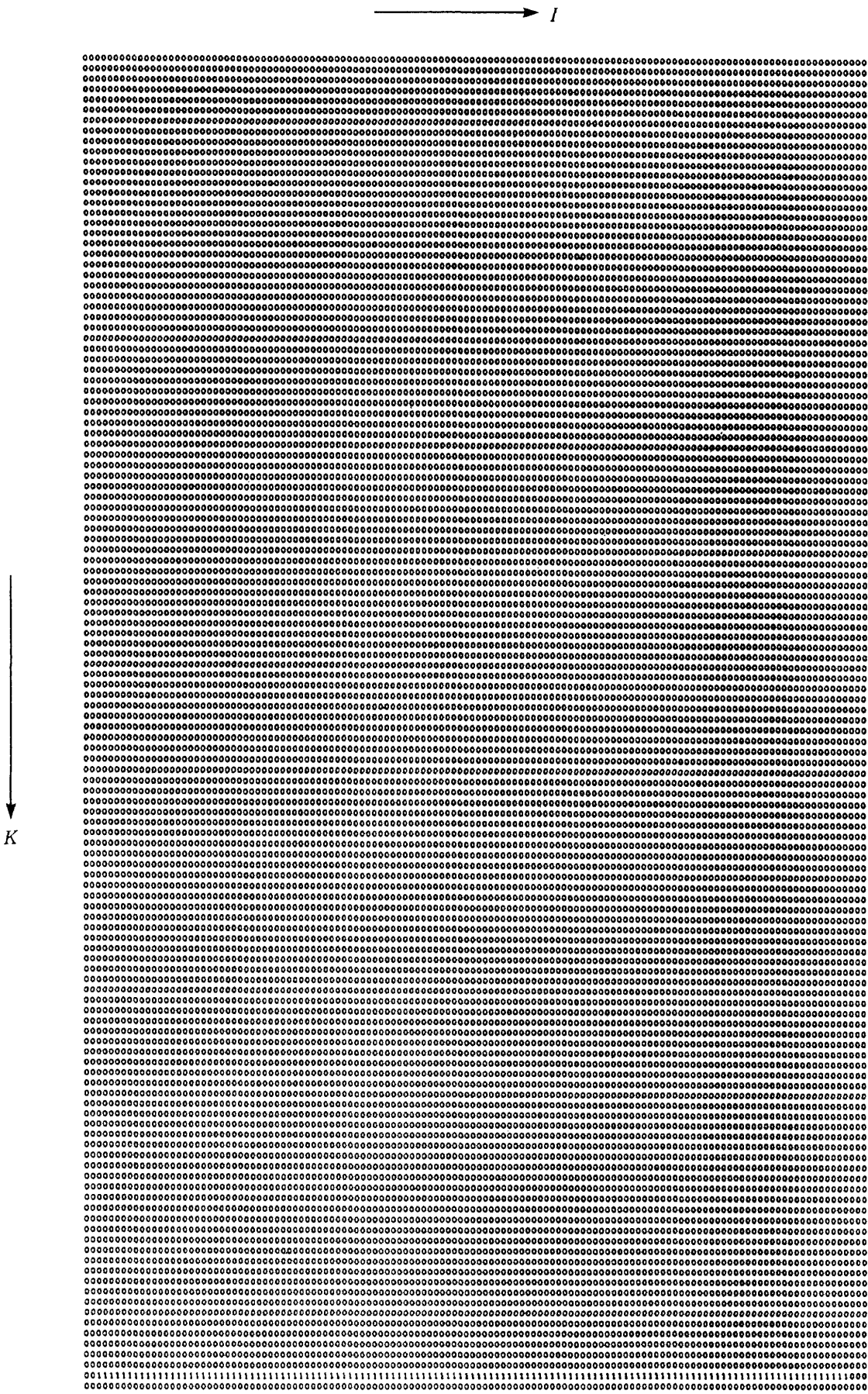


図21(4) 前面の外側格子点の演算マスク (J=3~125)

になっている。

(3) 収束判定のルーチンでは期待 (128×128倍) の約8割 (13,497倍) しか性能が出ていない。その主な原因は、この内容に収束判定の手続が含まれているためである。逐次処理では収束を満たさない格子点を発見すればただちに反復計算にもどることができ、127×127の全点を調べる必要がないのに対し並列処理ではアレイデータ演算器からスカラデータ演算器に集められた信号を基に収束判定する必要がありこれが処理時間の約2割を占める。したがって、より一層の性能向上を達成するためには格子点での収束判定回路および命令系の最適化を図り、かつプログラム内での計算比重を増すことが考えられる。図18に示したように収束のための反復回数は始めは多く以後減少して行くことから初期には100回単位で収束判定を行い、以後前回の反復回数を参考に収束判定手続の頻度を可変にすることにより実効上の性能向上を図ることができると思われる。ただし、時刻が進むにつれてその効果は薄れてゆくことは注意する必要がある。

(4) ポアソン方程式の解法部分では、処理手順をスカラ演算機で処理する場合と比較すると約11700倍にまた、時刻を dt 進める計算ではポアソン方程式の解法部分を1回通るとして約16700倍になることがわかる。しかしポアソン方程式の解法部分は図18に示すように複数回通過するため効率はポアソン方程式の解法部分の効率に近づく。したがって境界・外側格子点における効率を高めることが必要となる。

他方、逐次処理時間と並列処理時間との比較において‘公平’な比較であるかを検討することも必要である。これについて以下の項目を挙げるができる。

(5) 性能が優れている部分は並列処理が優れている以上に逐次処理の変換が適当でない場合も考えられる。前述したようにここでは最適化の手法を採っていない。並列処理手続きの最適化による性能改善よりも逐次処理手続きの最適化による性能改善の効果は命令数の多いだけ大きいと思われる。したがって両者に最適化(例えば loop unrolling)

を行えば並列処理がもたらす相対的性能は低下することが考えられる。しかし、最適化については計算機の構成ごとにその手法が異なり‘真の最適化’の判定は困難である。

(6) ここでは並列処理と逐次処理の両者でポアソン方程式の解法にヤコビ法を用いているが、各処理の演算法には最適な演算法があるのではないかという疑問がある(例えば並列処理にはゼブラ法、逐次処理にはSOR法等)。しかし、丸めを有する時の緩和法の保証について未解決な部分があり、計算機の構成に依存する部分もあり普遍的なこととして論ずることは危険と思われる。したがって、最も‘原始的’なヤコビ法で対処した。

上記の事項から並列処理で性能が何倍になったという数字も一つの目安であることを認識することが大事である。

6. Skew マトリックスによる性能向上

境界における性能向上のためには演算器を出来るだけ抑制しないことであり、そのためには境界・外側の格子点を各演算器に対応させれば良いであろう。このため Skew マトリックスを用いることとする。

6.1 データ構造の変換

図12に示す各格子点について $-1 \sim N+1$ を変数の指標の $2 \sim 128$ に対応させる。これにより立方体の表面は指標が 3 と 127 に対応する。また、図15に示す通常の演算器とデータの関係を以下のように変換し、各境界・外側の格子点を各演算器に対応付ける。

(1) 格子点を I 方向に J にしたがって1つずつずらせる。128を越えて張り出した部分(斜線部)をトーラス結合網を用いて巡回させる(図22)。

これにより、左右面およびそれらの外側の各格子点に演算器が1対1に対応する。

(2) 次に K 方向に J にしたがって1つずつずらせる。128を越えて張り出した部分(斜線部)をトーラス結合網を用いて巡回させる(図23)。

これにより、前後面およびそれらの外側の各格子点に演

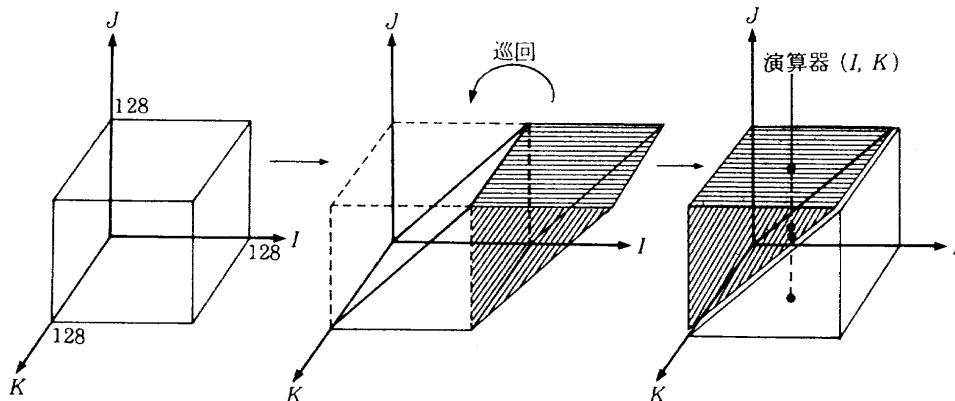


図22 Skew マトリックスによる対応(1)

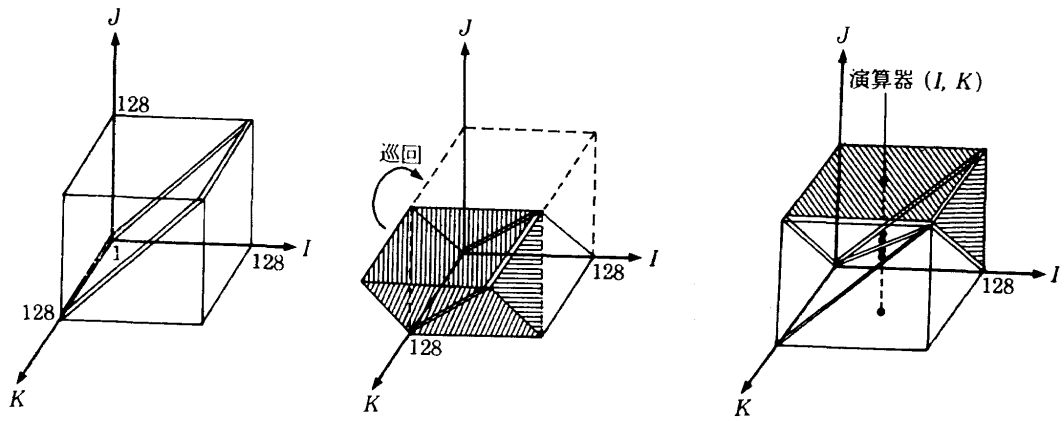


図23 Skew マトリックスによる対応(2)

表3 処理時間の比較 (Skew マトリックス)

	並列処理 (Skew)	逐次処理	比		
初期設定	(1)定数設定	619	593	1	
	(2)初期値設定	24,484	707,277,361	28,887	
	(3)上面の初期値設定	784	2,543,767	3,244	
	(4)最終時刻の設定	44	42	1	
時刻を Δt 進める 計算	(5)最終時刻の判定	68	68	1	
	(6)時刻の計算	74	69		
	(7) V^* の計算	233,505	5,557,723,252	23,801	
	(8) A の計算	40,272	629,890,243	15,460	
	(9) $B_{0,j,k}, B_{N,j,k}$ の計算	314	7,182,014	22,872	
	(10) $B_{i,0,k}, B_{i,N,k}$ の計算	297	7,182,014	24,181	
	(11) $B_{i,j,0}, B_{i,j,N}$ の計算	314	7,182,014	22,872	
	(12)最大反復数の設定	21	21	1	
	ポアソン方程式 の計算	(13)反復数の判定	68	68	1
		(14)左面・右面の圧力設定	184	3,332,347	18,110
		(15)上面・下面の圧力設定	138	3,332,347	24,147
		(16)前面・後面の圧力設定	184	3,332,347	18,110
(17)左面・右面外部の圧力設定		150	2,407,142	16,047	
(18)上面・下面外部の圧力設定		108	2,407,142	22,288	
(19)前面・後面外部の圧力設定		150	2,407,142	16,047	
(20)7点差分式の計算		62,658	1,668,606,365	26,630	
(21)内部圧力の計算		19,854	245,341,321	12,357	
(22)圧力修正量の絶対値の計算 および丸めの判定		26,702	345,465,013	12,937	
時刻を Δt 進める 計算	(23)流速の計算	59,102	1,322,389,960	22,374	
	(24)流速変数の更新	23,176	380,776,129	16,429	
終了時の処理	(25)終了時の処理	79	79	1	

(ポアソン方程式の計算での比較 $2,276,631,234/110,196=20,659$ 倍)
時刻を Δt 進める計算での比較 $10,188,957,018/467,339=21,802$ 倍)

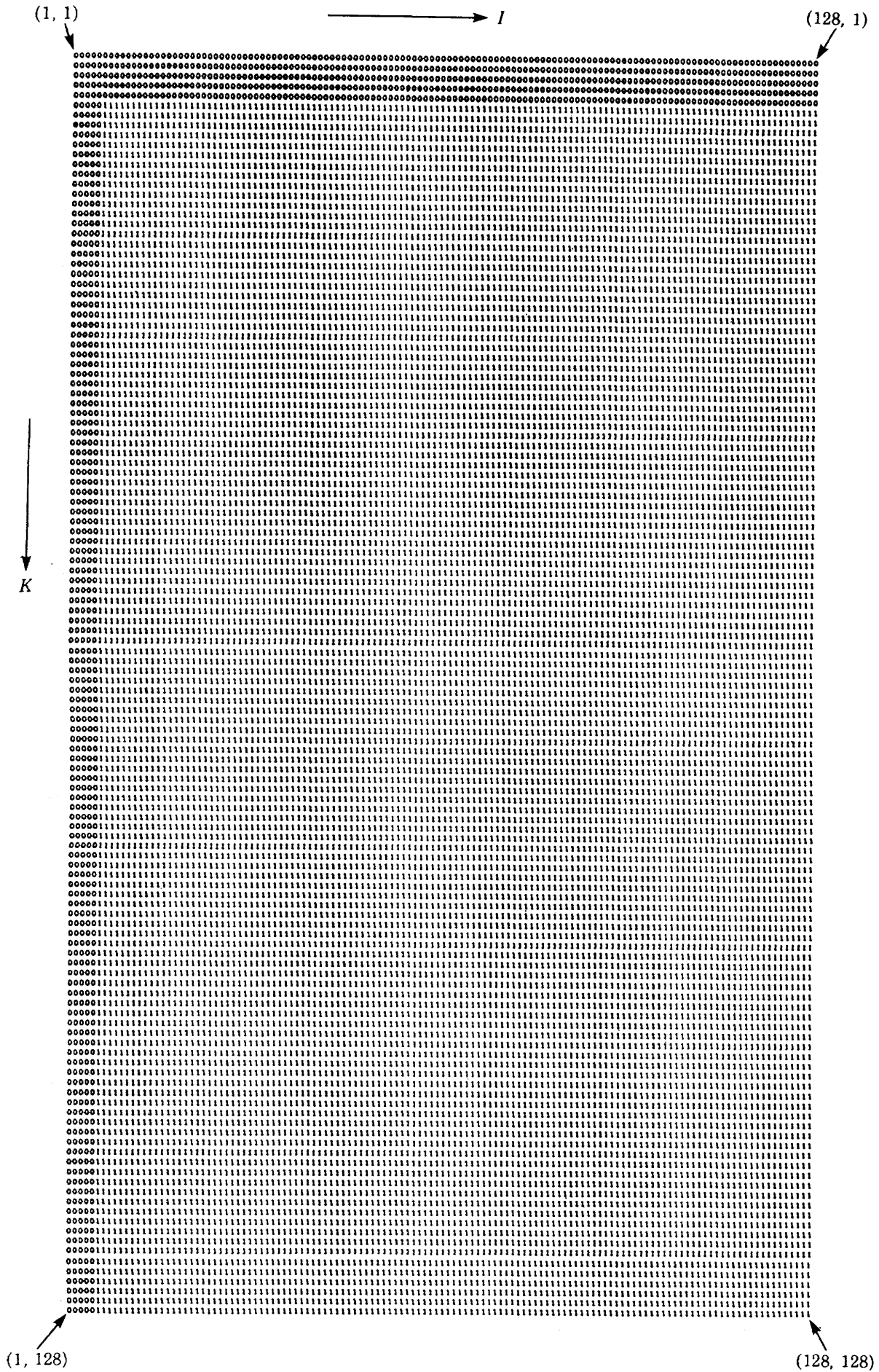


図24(1) 下面格子点の演算マスク (J=3)

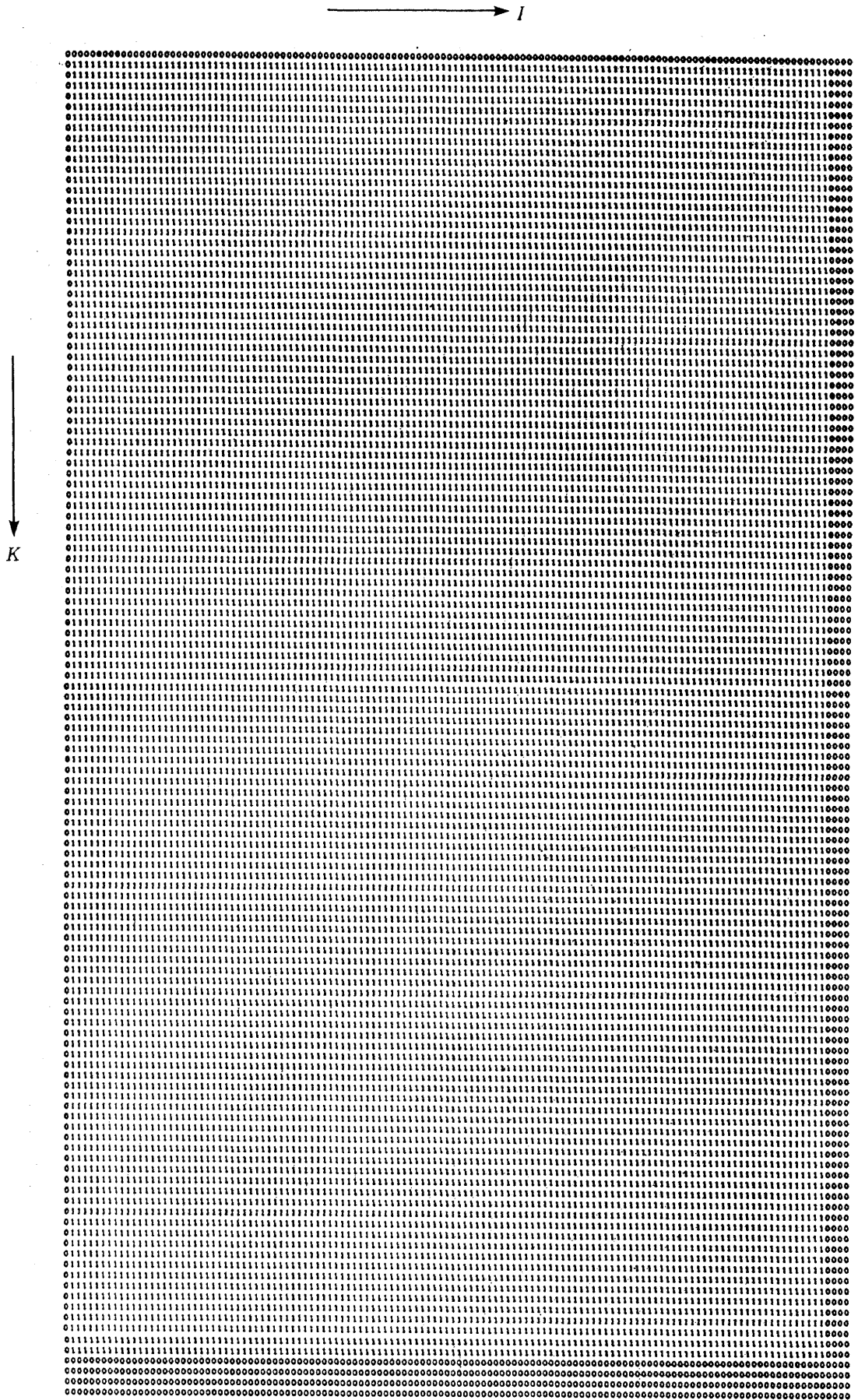


図24(2) 上面格子点の演算マスク (J=127)

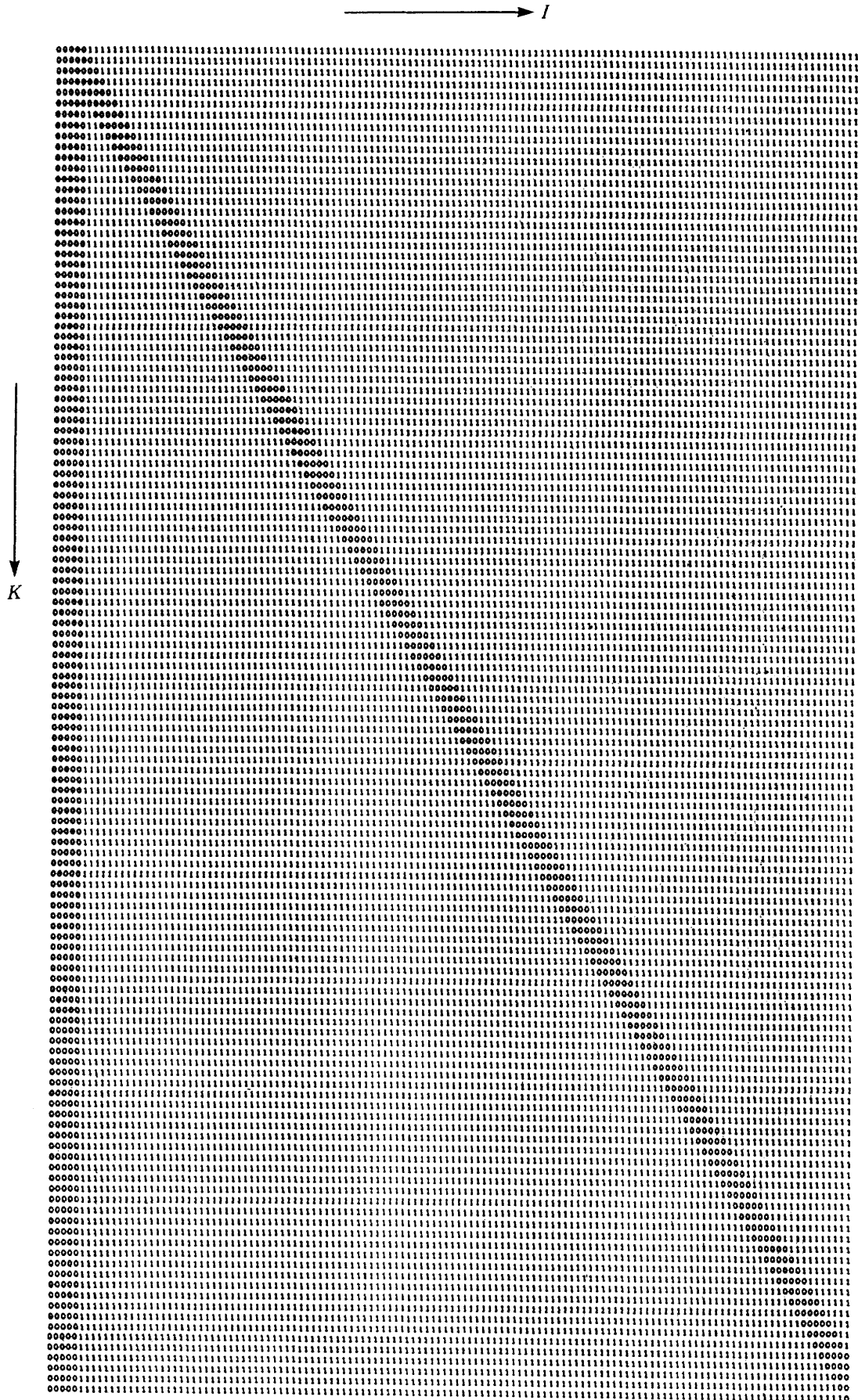


図24(3) 左面格子点の演算マスク (J=1-2)

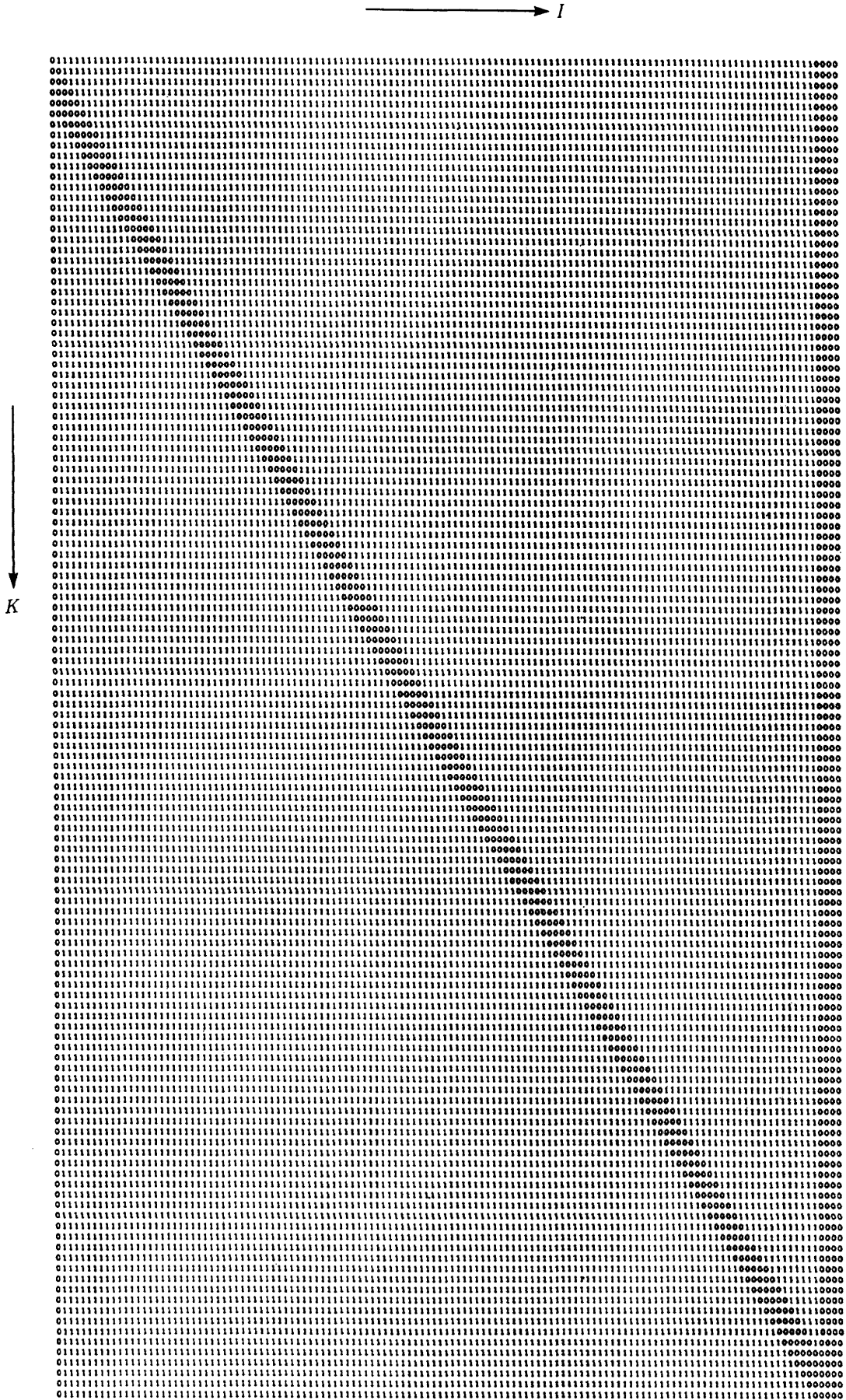


図24(4) 右面格子点の演算マスク (J=I+2)

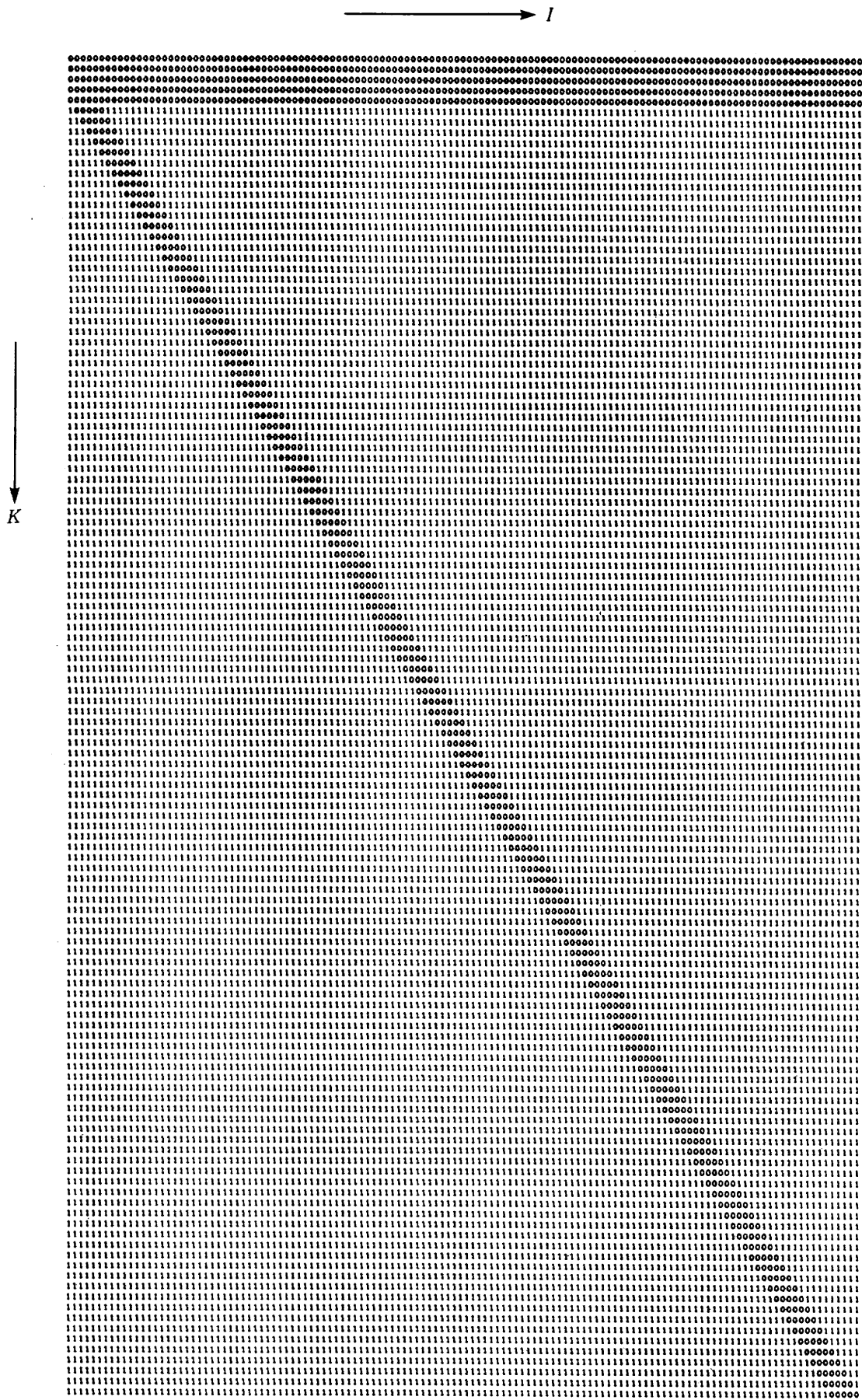


図24(5) 後面格子点の演算マスク (J=K-2)

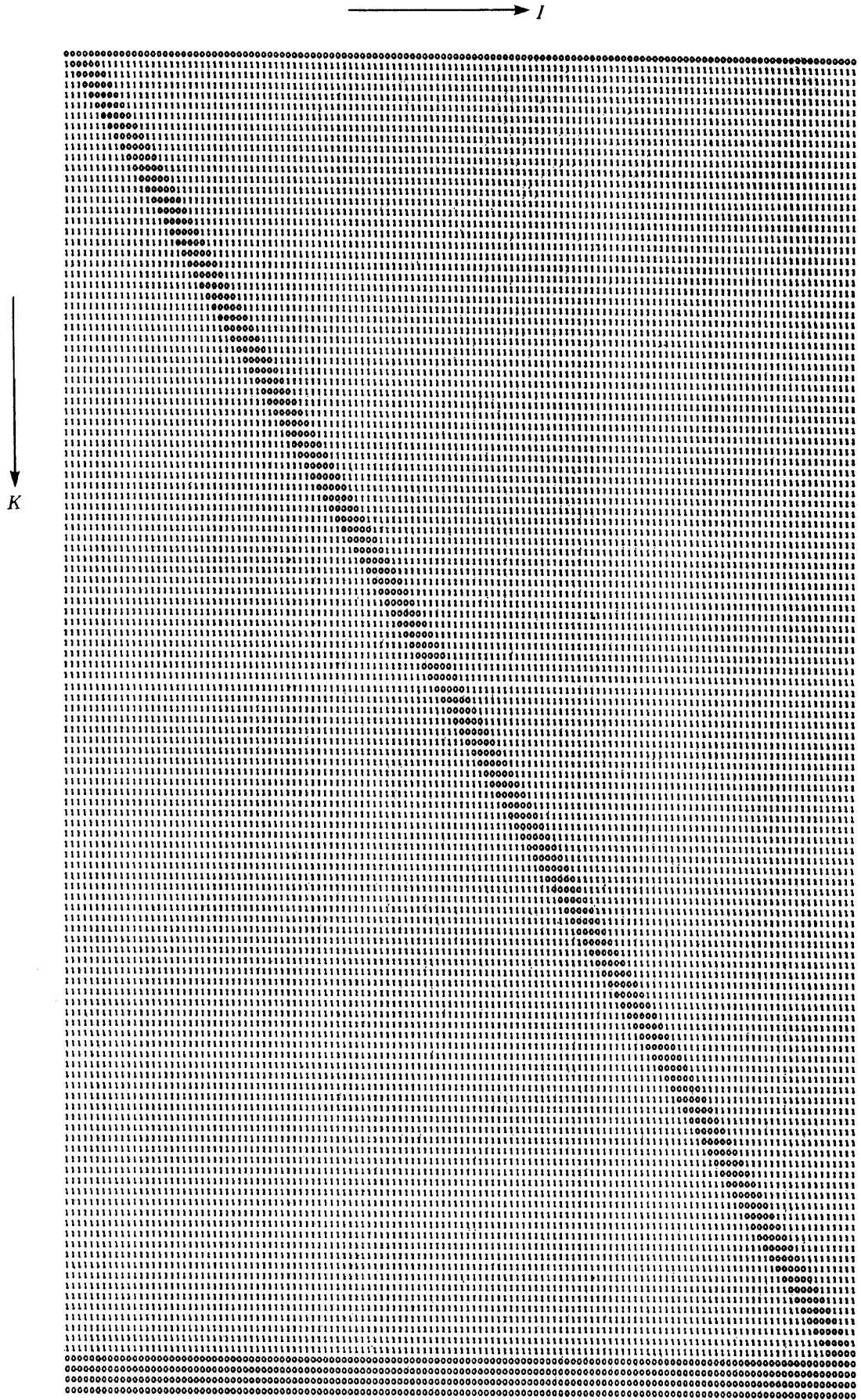


図24(6) 前面格子点の演算マスク (J=K+2)

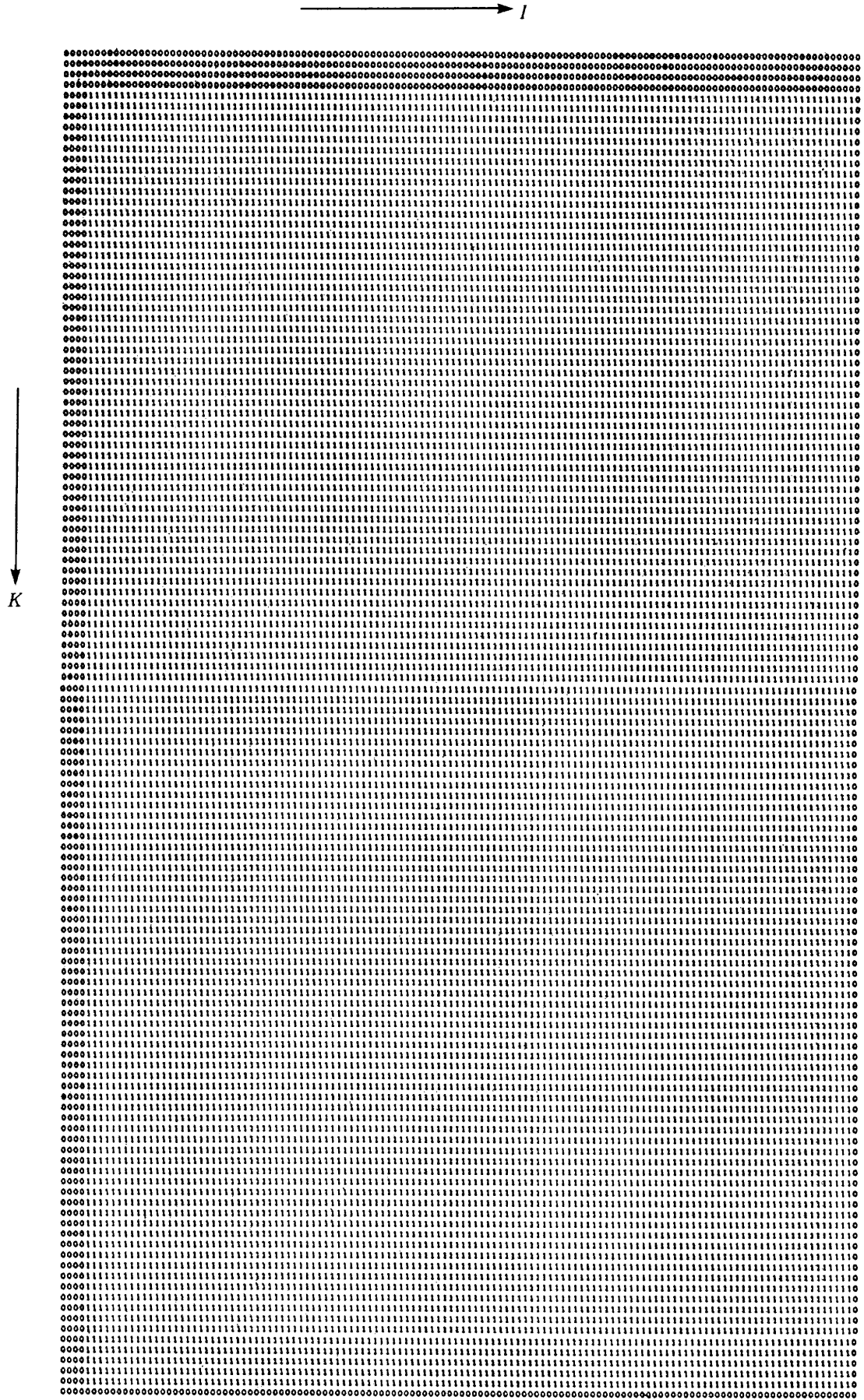


図25(1) 下面の外側格子点の演算マスク (J=2)

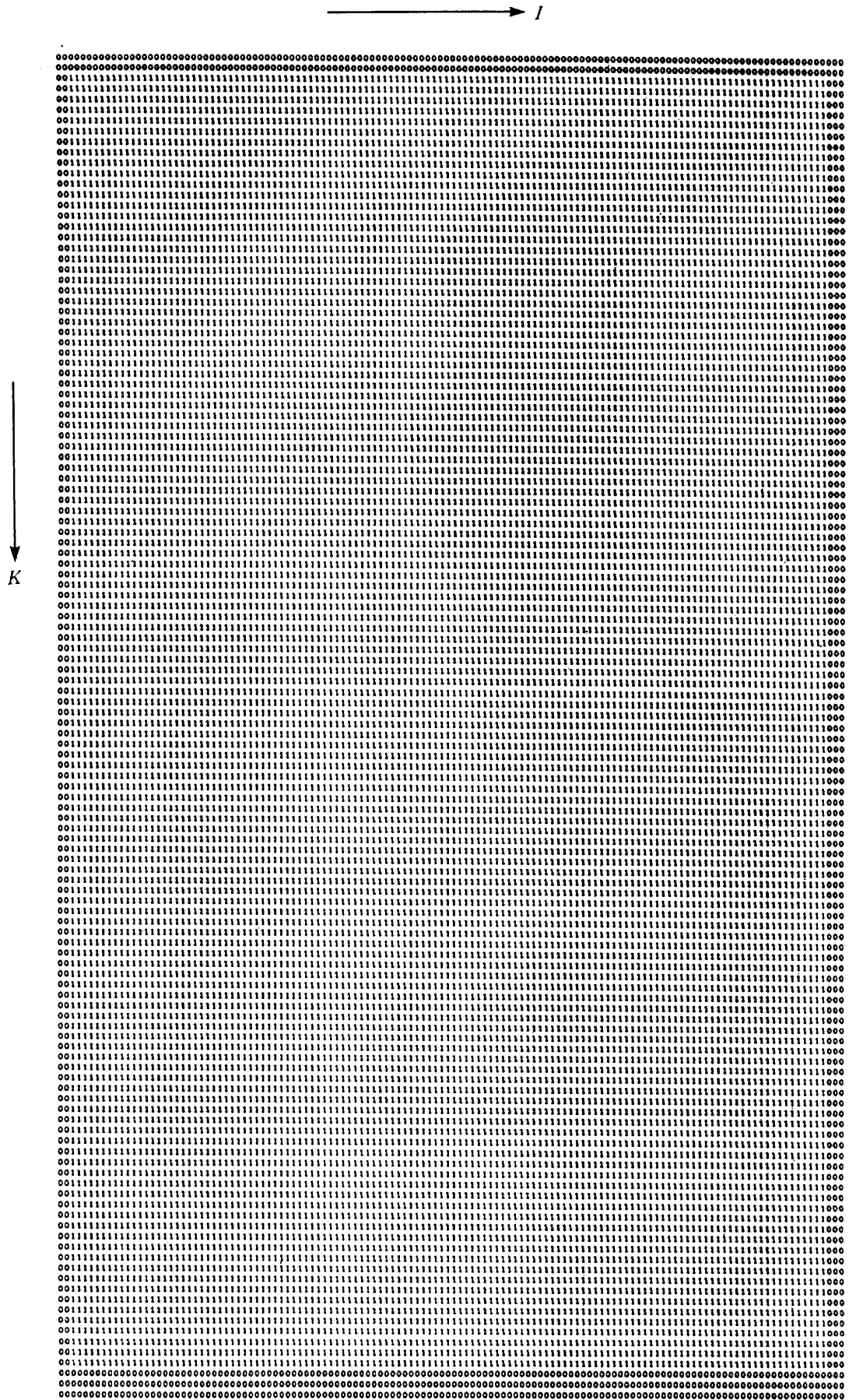


図25(2) 上面の外側格子点の演算マスク (J=128)

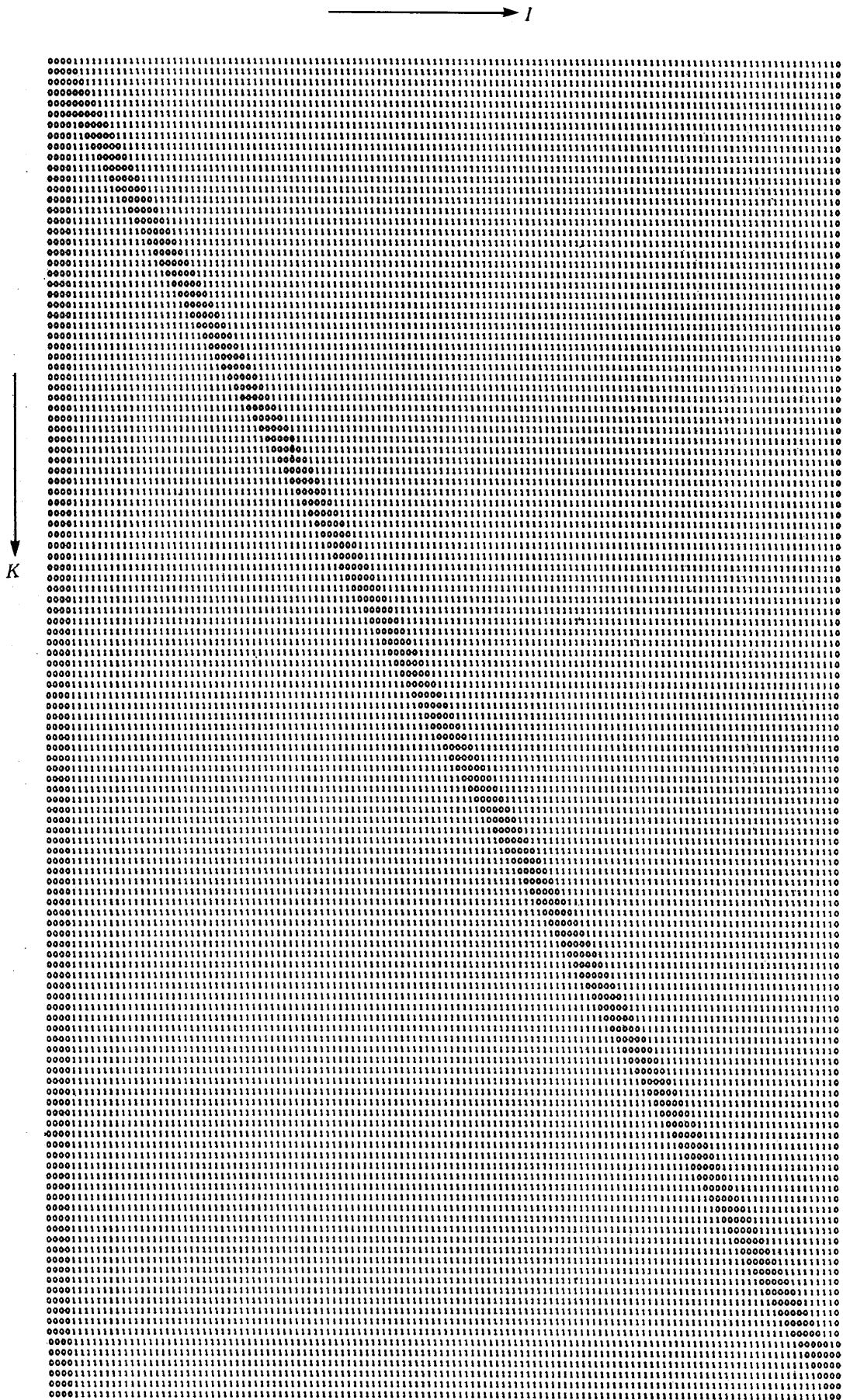


図25(3) 左面の外側格子点の演算マスク (J=I-1)

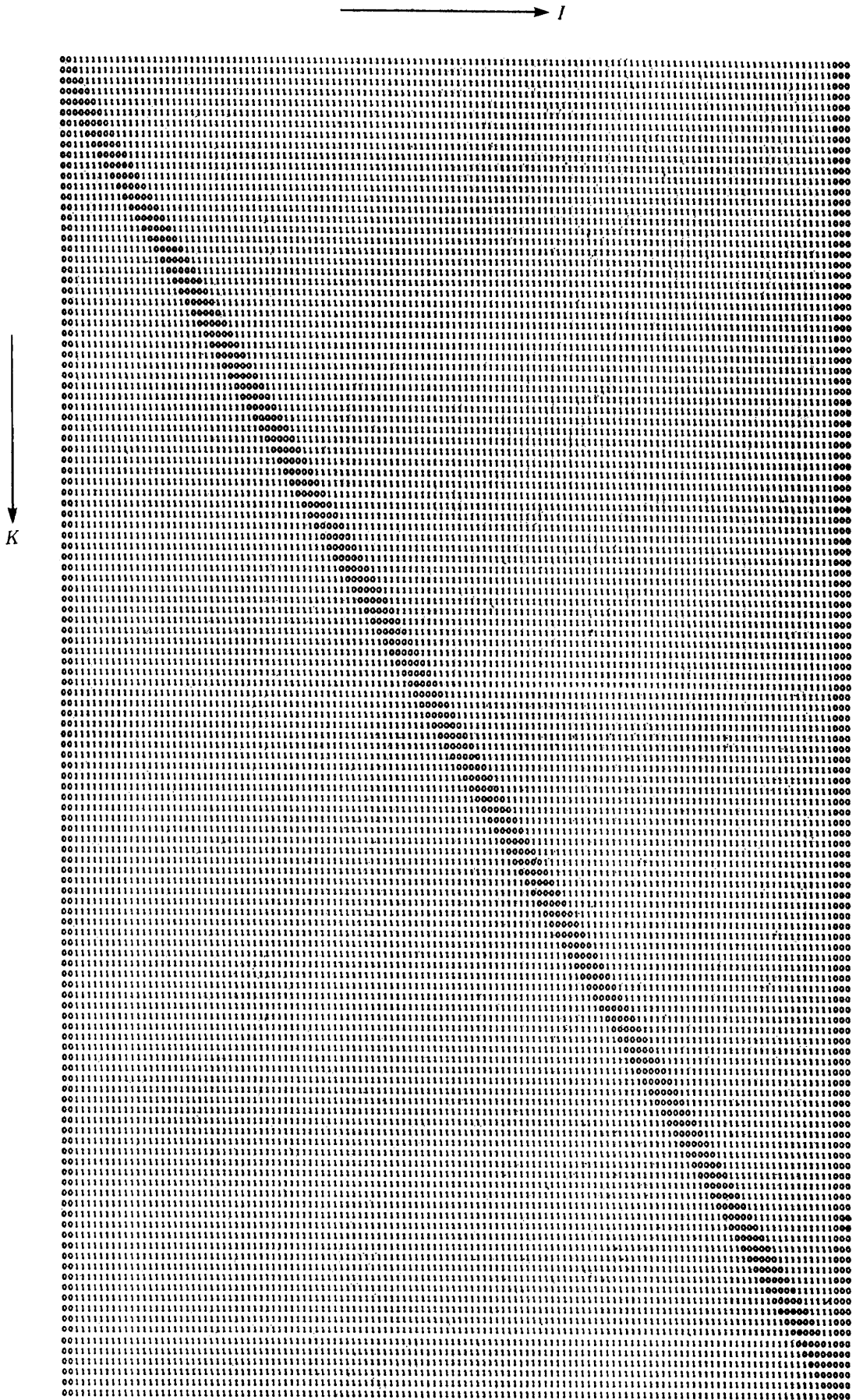


図25(4) 右面の外側格子点の演算マスク (J=I+1)

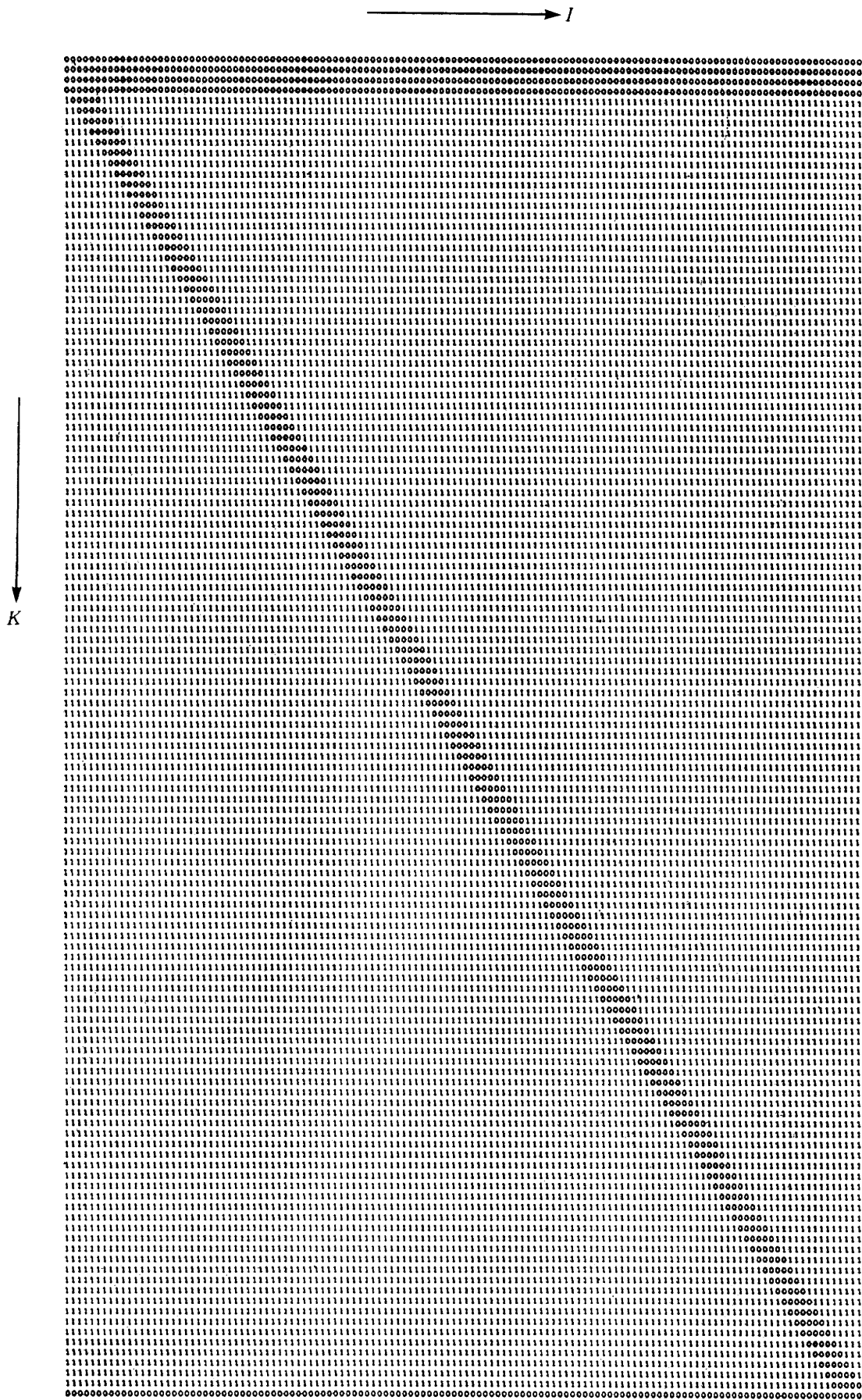


図25(5) 後面の外側格子点の演算マスク (J=K-1)



図25(6) 前面の外側格子点の演算マスク (J=K+1)

算器が1対1で対応し、結果として立方体の6面上の各格子点およびその外側の格子点に対し演算器が1対1で対応する。このときの変換は A, B を正整数として A/B の正の余りを $R[A, B]$ とすると

$$(I, J, K) \rightarrow (R[(I+J-1), 128]+1, J, R[(K+J-1), 128]+1)$$

である。

(3) 境界の格子点の演算に用いるマスクを図24に示す。また外側の格子点のそれを図25に示す。図中 J は対応する格子点の位置を示す指標である。また図26に左面格子点演算のためのマスクの原形を図27に I 方向にずらした時の断面を示す。これをさらに K 方向にずらして図24(3)が得られる。

(4) 内部格子点の演算では図28に示すマスクを J にした

がい1つずつ I, K 方向に移動させながら演算を進める。

6.2 考察

解析プログラムを同様に各ルーチンごとに分割し逐次処理と並列処理との処理時間を表3に示す。

この表を前述の表2と比べると表2で極端に性能の低い部分(9), (11), (14), (16), (17), (19)は大幅に向上している。図29に(14)のルーチンを示す。この処理時間は J に依ること無くこの計算機の構成と手続きがもたらす固定値である。また性能の高い部分(7), (8), (10), (15), (18), (20)~(24)は逆に少し低下している。図30に(20)のルーチンを示す。図16の場合に比べマスクの移動と‘ずれ’によるデータ転送の負荷が低下の原因である。また、処理時間は J の数に依存していることで格子数と演算器を増大させればこれらの部分の比重が

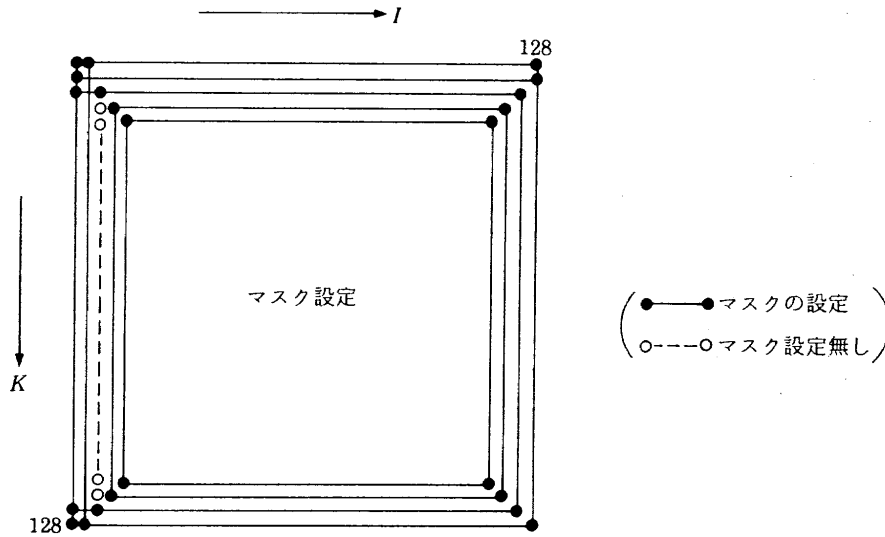


図26 マスクの原形

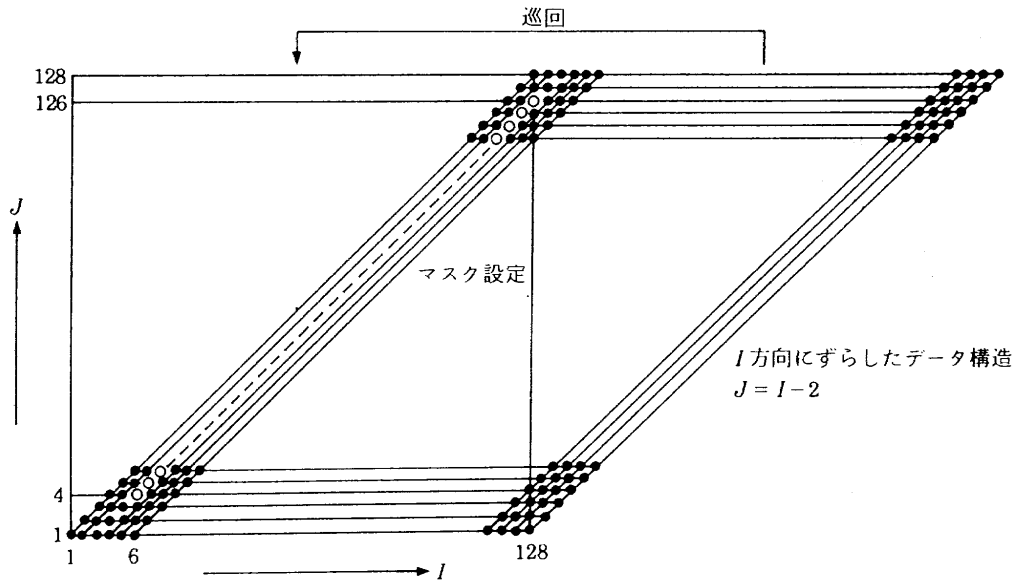


図27 I方向にずれたマスクの状態

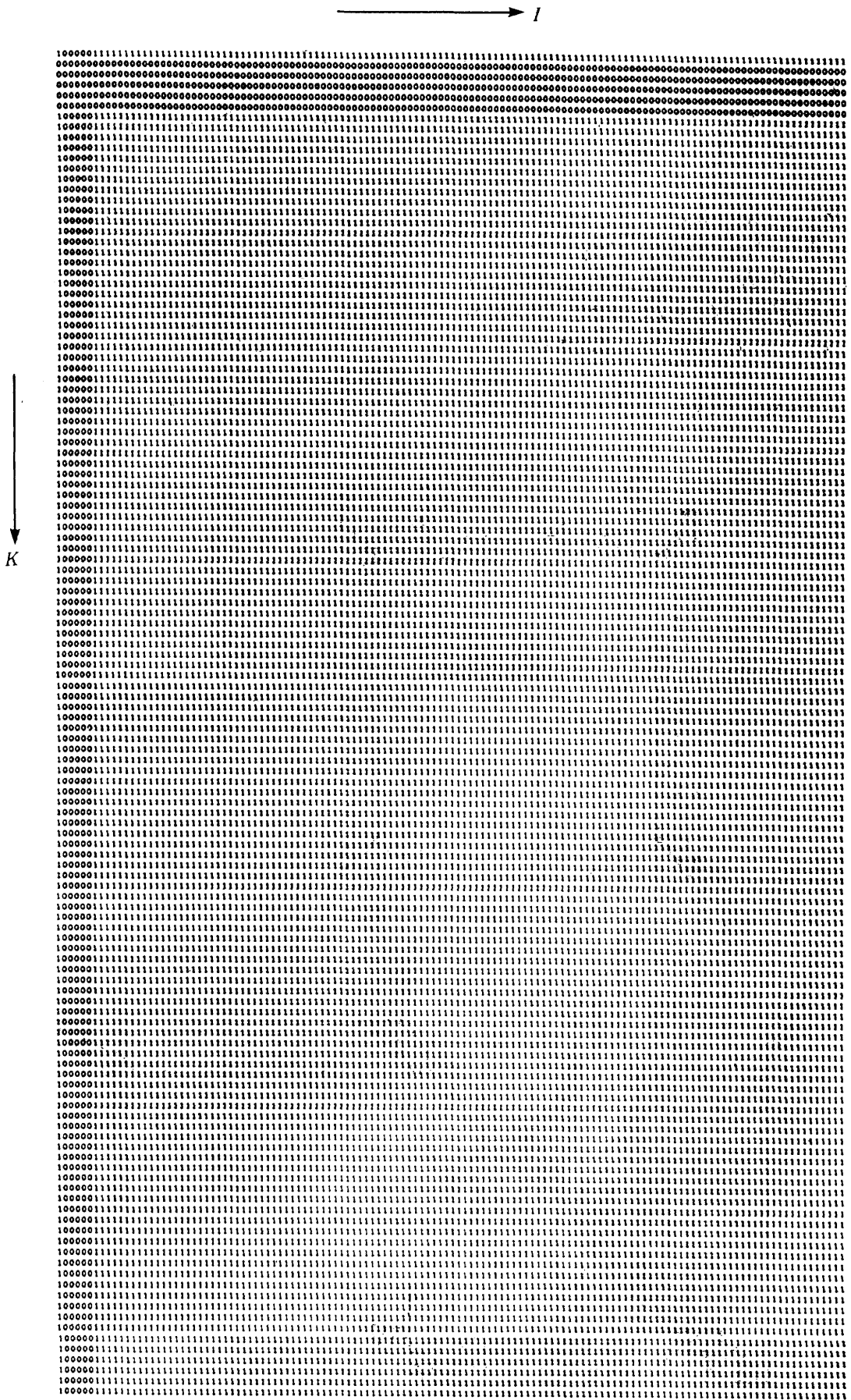


図28(i) 内部格子点の演算マスク (J=4)

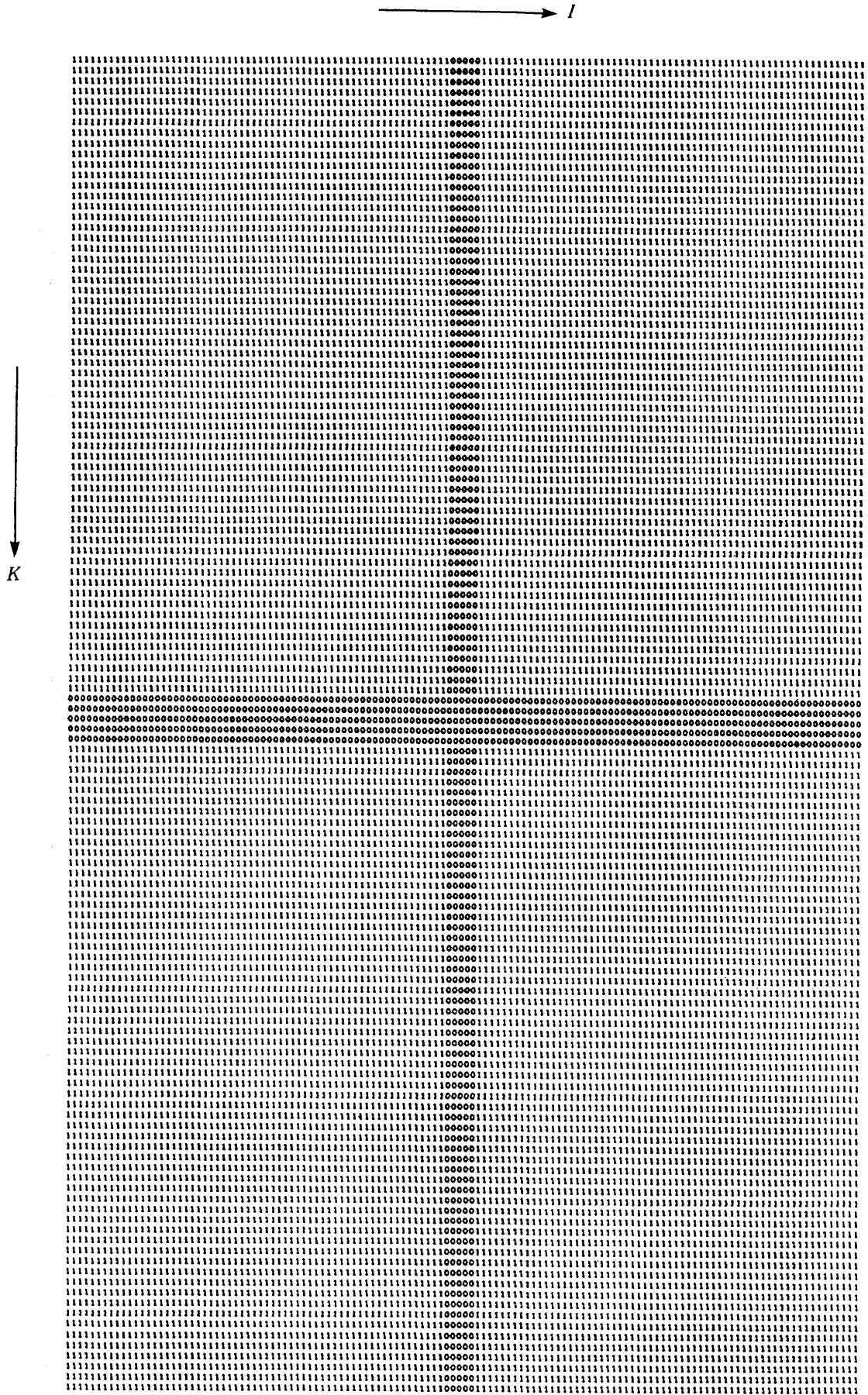


図28(2) 内部格子点の演算マスク (J=64)

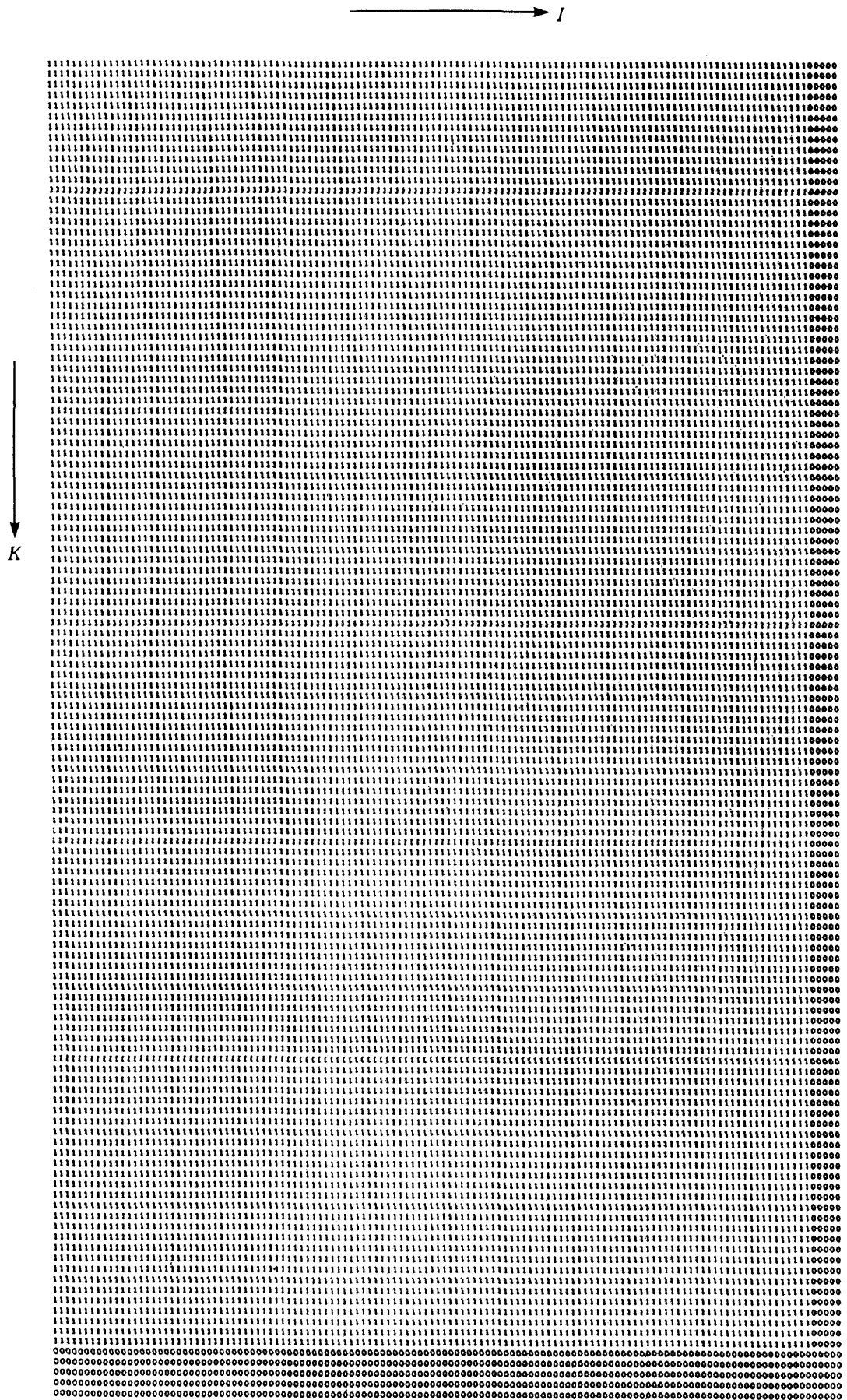


図28(3) 内部格子点の演算マスク (J=126)

```

;C*****C
;C
;      DO 2100 K = 3 , 125
;      DO 2100 J = 3 , 125
;      PT(2,J,K) = PT(3,J,K) - B(2,J,K)
;      PT(126,J,K) = PT(125,J,K) + B(126,J,K)
;2100   CONTINUE
;C
;C*****C
A2101   LA      3,0,ECNO,MONO,EMPT,0,C000,C000,INDEX1
;      SA      3,0,ECNO,MONO,EMPT,0,C000,C000,K00002
;
;A2100
;
;      MI
;      LA      0,0,ECNO,MKST,MINS,0,C000,C000,MASK3
;      FLA     0,3,ECNO,MONO,EMPT,0,C001,C000,PT
;      FSA     0,3,ECNO,MONO,EMPT,0,C000,C000,B
;      FTA     0,3,ECOF,MONO,EMPT,0,C000,C000,PT
;C
;      LA      3,0,ECNO,MONO,EMPT,0,C000,C000,INDEX1
;      AA      3,0,ECNO,MONO,EMPT,0,C000,C000,K00002
;
;      MI
;      LA      0,0,ECNO,MKST,MINS,0,C000,C000,MASK4
;
;      FLA     0,3,ECNO,MONO,EMPT,0,M001,C000,PT
;      FAA     0,3,ECNO,MONO,EMPT,0,C000,C000,B
;      FTA     0,3,ECOF,MONO,EMPT,0,C000,C000,PT
;C
;      HP
;C
;C*****C
    
```

図29 Skew マトリックス使用時の左右面の圧力設定

増大することを示している。

全体を観察するとポアソン方程式の解法部分では逐次処理に比べ並列処理では 20659 倍に、また時刻を dt 進める計算ではポアソン方程式の解法部分を 1 回通過するとして 21802 倍になっている。

以上のことから Skew マトリックスの効果は顕著であり、処理速度はポアソン方程式の解法部分で約 1.7 倍、時刻を dt すすめる部分で約 1.4 倍に向上する。

7. 性能の定式化

ここでは、格子点の数を n^3 、演算器の数を n^2 とし並列計算機の性能を定式化することを試みる。

3次元問題では並列化できる演算の比重が並列化できない部分に比べはるかに大きく、アムダールの法則による性能の‘飽和’はほぼ無視できる。表3のポアソン方程式の計算において並列化できないのは(3)の68クロックおよび(2)~(2)でループを設定する136クロック、113クロック、134

```

;C*****C
;C
;C POISSON SOLVER
;C
;      DO 900 K = 3 , 125
;      DO 900 J = 3 , 125
;      DO 900 I = 3 , 125
;      DELTA(I,J,K) = CONST * ((PT(I+2,J,K) + PT(I-2,J,K)
;      1      + PT(I,J+2,K) + PT(I,J-2,K) + PT(I,J,K+2)
;      2      + PT(I,J,K-2)) / 6.0 - PT(I,J,K)
;      3      - A(I,J,K) * 2.0 * H * H / 3.0)
;900   CONTINUE
;C
;C*****C
;
;A901   L      0,0,KOS001
;      T      0,0,J
;      LA     3,0,ECNO,MONO,EMPT,0,C000,C000,K00002
;      LA     0,0,ECNO,MONO,EMPT,0,C000,C000,MASK3D
;      TA     0,0,ECNO,MONO,EMPT,0,C000,C000,MASKWK
;
;A900  → L      0,0,J
;      IC     0,EMPT
;      T      0,0,J
;      MI
;      LA     0,0,ECNO,MKST,MINS,0,M001,M001,MASKWK
;      TA     0,0,ECNO,MONO,EMPT,0,C000,C000,MASKWK
;      ICA    3,ECNO,MONO,EMPT
;      L      1,0,KOS125
;      SR     0,1,MINS
;      J      0,A911
;
;      FLA     0,3,ECNO,MONO,EMPT,0,C002,C000,PT
;      FAA     0,3,ECNO,MONO,EMPT,0,M002,C000,PT
;      FAA     0,3,ECNO,MONO,EMPT,0,C002,C002,PT+2
;      FAA     0,3,ECNO,MONO,EMPT,0,M002,M002,PT-2
;      FAA     0,3,ECNO,MONO,EMPT,0,C000,C002,PT
;      FAA     0,3,ECNO,MONO,EMPT,0,C000,M002,PT
;      FDA     0,0,ECNO,MONO,EMPT,0,C000,C000,KF0006
;      FSA     0,3,ECNO,MONO,EMPT,0,C000,C000,PT
;      FLA     1,3,ECNO,MONO,EMPT,0,C000,C000,A
;      FMA     1,0,ECNO,MONO,EMPT,0,C000,C000,KF0002
;      FMA     1,0,ECNO,MONO,EMPT,0,C000,C000,H
;      FMA     1,0,ECNO,MONO,EMPT,0,C000,C000,H
;      FDA     1,0,ECNO,MONO,EMPT,0,C000,C000,KF0003
;      FSRA    0,1,ECNO,MONO,EMPT
;      FMA     0,0,ECNO,MONO,EMPT,0,C000,C000,CONST
;      FTA     0,0,ECOF,MONO,EMPT,0,C000,C000,DELTA
;
;      J      0,A900
;C
;C*****C
    
```

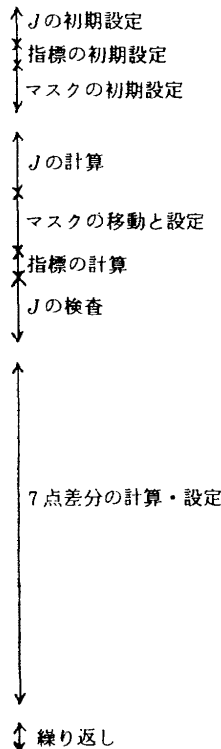


図30 Skew マトリックス使用時のプログラム例

クロックの合計 451 クロックでその比率は並列化した後でも約 0.41% である。格子点が増加するとその比重はさらに低下するため格子点の演算に関係する部分にだけ注目する。

ここで外側格子点・境界面の格子点を両端にとると内部の格子点数は $(n-4)^3$ である。また、計算の対象となる外側格子点数は $6(n-4)^2$ 、境界面の格子点数も同じく $6(n-4)^2$ である。

逐次処理では図 31 のような 3 重ループの処理時間は

$$((a(n-4)+b)(n-4)+c)(n-4)+d$$

で表すことができる。ここで a, b, c, d は各々 1 格子点の計算・ I ループの設定・ J ループの設定・ K ループ設定の処理時間であり計算機の構成と手続きによって決まる固定値である。例として表 3 の逐次処理における (7) V^* の処理時間は

$$((2985(n-4)+199)(n-4)+199)(n-4)+109$$

で表され、ここでは

$$n-4=123$$

の場合である。また図 32 の 2 重ループの処理時間は

$$(e(n-4)+f)(n-4)+g$$

で表すことができ、おなじく表 3 の逐次処理における (9) の処理時間は

$$(473(n-4)+210)(n-4)+167$$

で表すことができる。 n が 100 以上のときには第 2 項は 2 桁小さいことから第 1 項だけに注目することになると 3 重ループの処理時間は

$$a(n-4)^3$$

2 重ループの処理時間は

$$e(n-4)^2$$

で近似できる。

ここで逐次処理で左右面上の格子点の演算処理時間を α_1 、前後面上の格子点の演算処理時間を β_1 、上下面上の格子点の演算処理時間を γ_1 、左右面の外側格子点の演算のそれを δ_1 、前後面の外側格子点のそれを ε_1 、上下面の外側格子点のそれを ζ_1 、内部格子点のそれを η_1 とするとその処理時間は

$$\alpha_1(n-4)^2 + \beta_1(n-4)^2 + \gamma_1(n-4)^2 + \delta_1(n-4)^2 + \varepsilon_1(n-4)^2 + \zeta_1(n-4)^2 + \eta_1(n-4)^3$$

で近似できる。次に通常のマトリックスによるデータ構造で並列処理におけるそれぞれを $\alpha_2, \beta_2, \gamma_2, \delta_2, \varepsilon_2, \zeta_2, \eta_2$ とすると左右前後面とその外側格子点では $n-4$ 台の演算器に、また上下境界面・その外側格子点および内部格子点については $(n-4)^2$ 台の演算器によるとしてその処理時間は

$$\begin{aligned} & (\alpha_2(n-4)^2 + \beta_2(n-4)^2 + \delta_2(n-4)^2 + \varepsilon_2(n-4)^2) \\ & / (n-4) + (\gamma_2(n-4)^2 + \zeta_2(n-4)^2 + \eta_2(n-4)^3) \\ & / (n-4)^2 = (\alpha_2 + \beta_2 + \delta_2 + \varepsilon_2 + \eta_2)(n-4) + \gamma_2 + \zeta_2 \end{aligned}$$

DO 1000 K=3, n-2

DO 1000 J=3, n-2

DO 1000 I=3, n-2

P(I, J, K) =

⋮

1000 CONTINUE

図 31 3 重ループの例

DO 2000 K=3, n-2

DO 2000 I=3, n-2

P(I, 1, K) =

P(I, n, K) =

⋮

2000 CONTINUE

図 32 2 重ループの例

である。

データ構造を Skew マトリックスにすると境界面とその外側格子点 $(n-4)^2$ 個に演算器 $(n-4)^2$ 個が対応する。ここで $\alpha_3, \beta_3, \gamma_3, \delta_3, \varepsilon_3, \zeta_3, \eta_3$ を同じく計算機の構成と手続きによりきまる固定値とすると、その処理時間は

$$\begin{aligned} & (\alpha_3(n-4)^2 + \beta_3(n-4)^2 + \gamma_3(n-4)^2 + \delta_3(n-4)^2 \\ & + \varepsilon_3(n-4)^2 + \zeta_3(n-4)^2) / (n-4)^2 \\ & + \eta_3(n-4)^3 / (n-4)^2 = \alpha_3 + \beta_3 + \gamma_3 + \delta_3 + \varepsilon_3 \\ & + \zeta_3 + \eta_3(n-4) \end{aligned}$$

となる。

ここで演算器 1 台あたりの効率を

逐次処理時間 / (並列処理時間・演算器数)

とする。この値が 1 を越えると演算器台数以上の性能であることを表す。演算器数は n^2 台であり、通常のマトリックスを用いた場合には

$$\begin{aligned} & (\alpha_1(n-4)^2 + \beta_1(n-4)^2 + \gamma_1(n-4)^2 + \delta_1(n-4)^2 \\ & + \varepsilon_1(n-4)^2 + \zeta_1(n-4)^2 + \eta_1(n-4)^3) \\ & / ((\alpha_2 + \beta_2 + \delta_2 + \varepsilon_2 + \eta_2)(n-4) + \gamma_2 + \zeta_2) \cdot n^2 \\ & \rightarrow \eta_1 / (\alpha_2 + \beta_2 + \delta_2 + \varepsilon_2 + \eta_2) \quad (n \rightarrow \infty) \end{aligned}$$

となり、Skew マトリックスを用いた場合には

$$\begin{aligned} & (\alpha_1(n-4)^2 + \beta_1(n-4)^2 + \gamma_1(n-4)^2 + \delta_1(n-4)^2 \\ & + \varepsilon_1(n-4)^2 + \zeta_1(n-4)^2 + \eta_1(n-4)^3) \\ & / ((\alpha_3 + \beta_3 + \gamma_3 + \delta_3 + \varepsilon_3 + \zeta_3 + \eta_3(n-4)) \cdot n^2) \\ & \rightarrow \eta_1 / \eta_3 \quad (n \rightarrow \infty) \end{aligned}$$

となる。後者は大規模 3 次元問題では内部格子点での計算量により性能が決定されることを示している。表 2 および 3 のデータから $\alpha_i, \beta_i, \gamma_i, \delta_i, \varepsilon_i, \zeta_i, \eta_i$ ($i=1 \sim 3$) を求めるため表 4 および 5 を作成する。

これよりポアソン方程式の解法で通常のマトリックスで

表 4 ポアソン方程式の計算

計算の種類 \ 処理形態	逐次処理時間	並列処理時間	並列処理時間 (Skew マトリックス)
(14) 左右面の格子点	3,332,347 = 220.2 × 123 ² ∴ α ₁ = 220.2	26,435 = 224.9 × 123 ∴ α ₂ = 214.9	184 ∴ α ₃ = 184
(15) 上下面の格子点	3,332,347 = 220.2 × 123 ² ∴ γ ₁ = 220.2	107 ∴ γ ₂ = 107	138 ∴ γ ₃ = 138
(16) 前後面の格子点	3,332,347 = 220.2 × 123 ² ∴ β ₁ = 220.2	26,435 = 214.9 × 123 ∴ β ₂ = 214.9	184 ∴ β ₃ = 184
(17) 左右面の外部側の格子点	2,407,142 = 159.1 × 123 ² ∴ δ ₁ = 159.1	22,253 = 180.9 × 123 ∴ δ ₂ = 180.9	150 ∴ δ ₃ = 150
(18) 上下面の外部側の格子点	2,407,142 = 159.1 × 123 ² ∴ ζ ₁ = 159.1	69 ∴ ζ ₂ = 69	108 ∴ ζ ₃ = 108
(19) 前後面の外部側の格子点	2,407,142 = 159.1 × 123 ² ∴ ε ₁ = 159.1	22,253 = 180.9 × 123 ∴ ε ₂ = 180.9	105 ∴ ε ₃ = 105
(20) 内部格子点	1,668,606,365	55,978	62,658
(21)	245,341,321	15,119	19,854
(22)	345,465,013	25,595	26,702
	計 2,259,412,699 = 1214.1 × 123 ³ ∴ η ₁ = 1214.1	計 96,692 = 786.1 × 123 ∴ η ₂ = 786.1	計 109,214 = 887.9 × 123 ∴ η ₃ = 887.9

の効率は表 4 を用いて

$$\begin{aligned} \eta_1 / (\alpha_2 + \beta_2 + \delta_2 + \epsilon_2 + \eta_2) \\ = 1214.1 / (214.9 \times 2 + 180.9 \times 2 + 786.1) \\ = 0.7695 \end{aligned}$$

となり、Skew マトリックスでの効率は

$$\eta_1 / \eta_3 = 1214.1 / 887.9 = 1.3673$$

となる。

さらに時刻を $4t$ 進める計算で通常のマトリックスでの効率は表 5 を用いて

$$\begin{aligned} \eta_1 / (\alpha_2 + \beta_2 + \delta_2 + \epsilon_2 + \eta_2) \\ = 5454.5 / (559.8 \times 2 + 180.9 \times 2 + 3465.1) \\ = 1.1026 \end{aligned}$$

となり、Skew マトリックスでの効率は

$$\eta_1 / \eta_3 = 5454.5 / 3782.6 = 1.4419$$

であり Skew マトリックスの優位性が明確となる。

8. おわりに

集積回路の進歩は著しく、その時々々の回路技術に最適な計算機構成があるはずであり、一台の計算機による処理時間と複数台の計算機によるそれとの比は並列計算機を構築するうえで最も分かり易い指標である。

応用プログラムを分割し、シミュレータを用いて各部分ごとに逐次処理による演算時間と並列処理による演算時間を得て、それらを比較することにより、並列計算機の性能を定量的に示すことが可能となった。この結果は性能向上策を検討する時の基礎データとして極めて明快である。並列計算機による三次元流れ場の解析で Skew マトリックスの効果は顕著であり、演算器に見合う性能向上が達成できると言える。

他方、入出力処理についてはさらに多くのプログラムの検討が必要であり、また並列計算機の構成、演算器、結合網、命令系および計算法等の見直しにより、より一層の性

表5 時刻を Δt 進める計算

計算の種類	逐次処理時間	並列処理時間	並列処理時間 (Skew マトリックス)
(9) (14) 左右面の格子点	7,182,014	42,425	314
	3,332,347	26,435	184
	計 10,514,361	計 68,860	計 498
	$=694.9 \times 123^2$ $\therefore \alpha_1 = 694.9$	$=559.8 \times 123$ $\therefore \alpha_2 = 559.8$	$\therefore \alpha_3 = 498$
(10) (15) 上下面の格子点	7,182,014	225	297
	3,332,347	107	138
	計 10,514,361	計 332	計 435
	$=694.9 \times 123^2$ $\therefore r_1 = 694.9$	$\therefore r_2 = 332$	$\therefore r_3 = 435$
(11) (16) 前後面の格子点	7,182,014	42,425	314
	3,332,347	26,435	184
	計 10,514,361	計 68,860	計 498
	$=694.9 \times 123^2$ $\therefore \beta_1 = 694.9$	$=559.8 \times 123$ $\therefore \beta_2 = 559.8$	$\therefore \beta_3 = 498$
(17) 左右面の外側の 格子点	2,407,142	22,253	150
	$=159.1 \times 123^2$ $\therefore \delta_1 = 159.1$	$=180.9 \times 123$ $\therefore \delta_2 = 180.9$	$\therefore \delta_3 = 150$
	(18) 上下面の外側の 格子点	2,407,142	69
	$=159.1 \times 123^2$ $\therefore \zeta_1 = 159.1$	$\therefore \zeta_2 = 69$	$\therefore \zeta_3 = 108$
(19) 前後面の外側の 格子点	2,407,142	22,253	150
	$=159.1 \times 123^2$ $\therefore \epsilon_1 = 159.1$	$=180.9 \times 123$ $\therefore \epsilon_2 = 180.9$	$\therefore \epsilon_3 = 150$
	(7) (8) (20) (21) (22) (23) (24) 内部格子点	5,557,723,252	223,135
629,890,243	34,553	40,272	
1,668,606,365	55,978	62,658	
245,341,321	15,119	19,854	
345,465,013	25,595	26,702	
1,322,389,960	53,395	59,102	
380,776,129	18,440	23,176	
計 10,150,192,283	計 426,215	計 465,269	
$=5,454.5 \times 123^3$ $\therefore \eta_1 = 5,454.5$	$=3,465.1 \times 123$ $\therefore \eta_2 = 3,465.1$	$=3,782.6 \times 123$ $\therefore \eta_3 = 3,782.6$	

能向上を図ることが今後の課題である。

今後は有限要素法等の各種応用プログラムを並列計算機に移植し命令追跡による模擬を実施することを計画している。

謝 辞

本研究を行うにあたり富士通(株)を始めとする多くの方々から有益な御意見・御協力をいただいた。また、流れ場

解析プログラムの作成にあたり福井大学の井上建二教授より助言を頂いたことを記し感謝の意を表する。

参 考 文 献

- 1) Kawaguti, M. & Jain, P. : "Numerical Study of a Viscous Fluid Flow Past a Circular Cylinder." J. Phys. Soc. Japan, 21 (1966), 2055-62
- 2) 原田, 吉田, 中村: "並列計算機の検証模擬", 航空

- 宇宙技術研究所報告, TR-1154, 1992年 5月
- 3) 原田, 吉田, 中村: “2次元CFDによる並列計算機のシミュレーション”, 第7回航空機計算空気力学シンポジウム論文集, pp.13~17, 航空宇宙技術研究所, SP-10, 1989年11月
 - 4) 原田: “並列計算機のシミュレーション”, 第6回航空機計算空気力学シンポジウム論文集, pp.49~54, 航空宇宙技術研究所, SP-9, 1988年12月
 - 5) 原田: “並列計算機のアセンブラ”, 航空宇宙技術研究所資料, TM-586, 1988年 6月
 - 6) 原田: “並列計算機のアーキテクチャシミュレータ”, 航空宇宙技術研究所資料, TM-583, 1988年 3月
 - 7) 棚橋他: “三次元立方体キャビティ内の流れ”, 日本機械学会第3回JSME-CFDワークショップ課題1資料集, 1990年 4月19, 20日
 - 8) 河合他: “GSMAC 有限要素法による立方体キャビティ内の非定常三次元流れの解析”, 日本機械学会論文集(B編) 55巻515号論文 No.88-1236A, pp.1922~1929, 1989年
 - 9) 星野: “PAX コンピュータ”, オーム社, pp.92~94, 昭和60年 3月15日

航空宇宙技術研究所報告1211号

平成5年8月発行

発行所 航空宇宙技術研究所
東京都調布市深大寺東町7丁目44番地1
電話三鷹(0422)47-5911(大代表)〒182
印刷所 株式会社三興印刷
東京都新宿区西早稲田2-1-18

Printed in Japan