第３章

ＳＩＭＤコンピュータ

３．１ＳＩＭＤ方式の原理

3.1.1原理

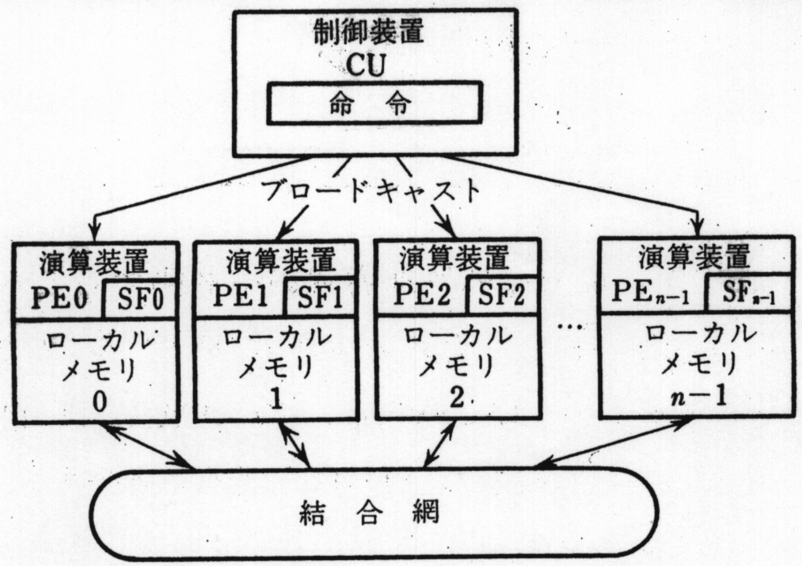
制御装置（CU）から発せられた命令：

同一構造を有する多数の演算装置（PE）に

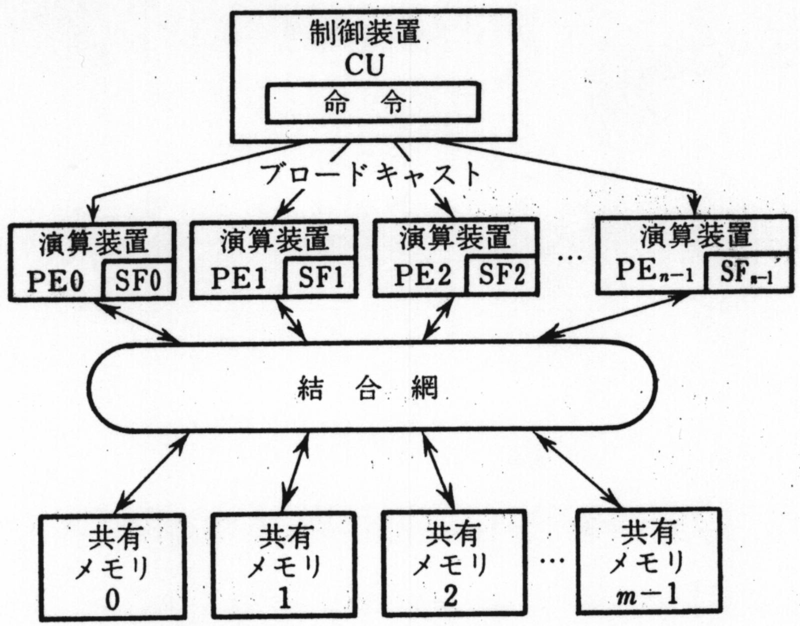
同時にブロードキャスト

メモリ共有型とメモリ非共有型

演算実行の抑止



（a） メモリ非共有



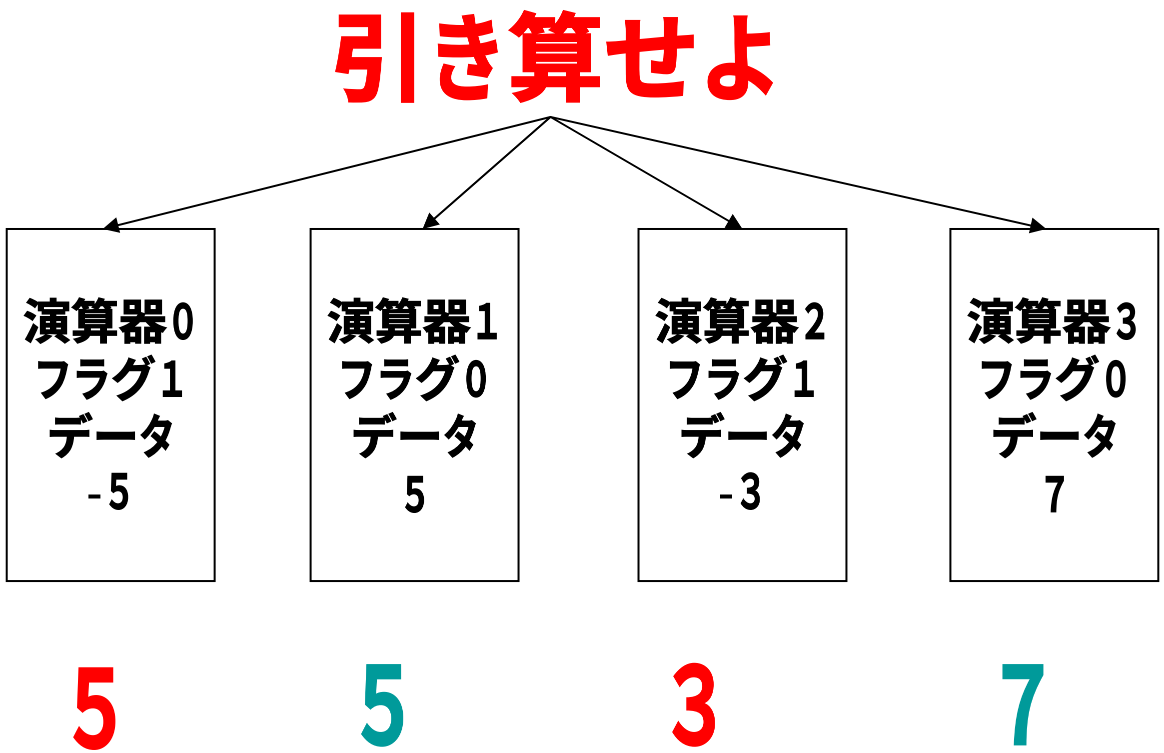
（b） メモリ共有

SIMD演算の例：絶対値の計算

正のデータを持っているもの　抑止フラグ0

負のデータを持っているもの　抑止フラグ1

0ー(データ)の一斉実行



3.1.2特徴

①演算装置の簡略化

②細粒度並列処理指向

③定型処理指向

　分岐少ない方がよい

④命令同期型プログラミング

　全PE終了後次命令発行

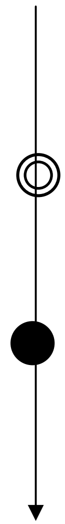
　プログラミングの制約

　　標準的なプログラミング

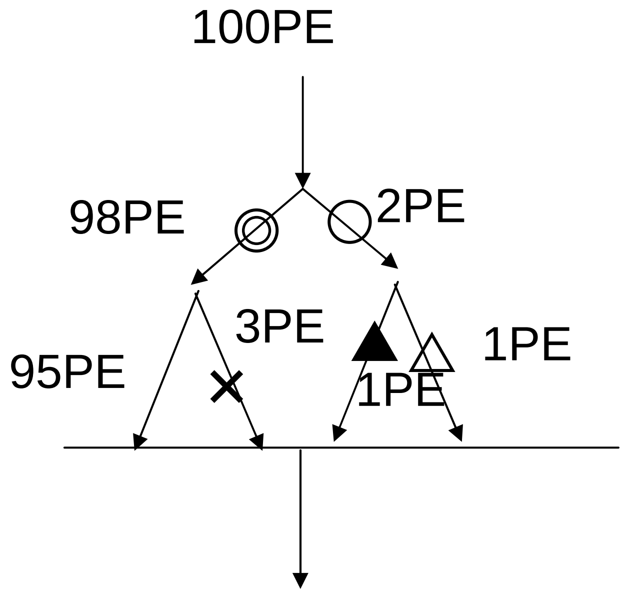
　　生産性が向上？

　コンパイラによる最適化

　　全PEが同時終了するよう，無資源競合



分岐のない
場合2演算



分岐のある
場合6演算

3.1.3　SIMD方式の柔構造化

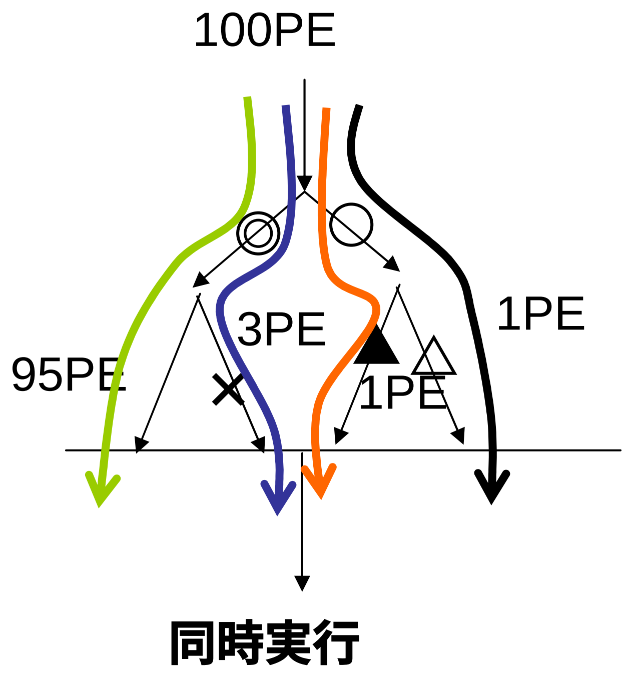
①通信の非同期化

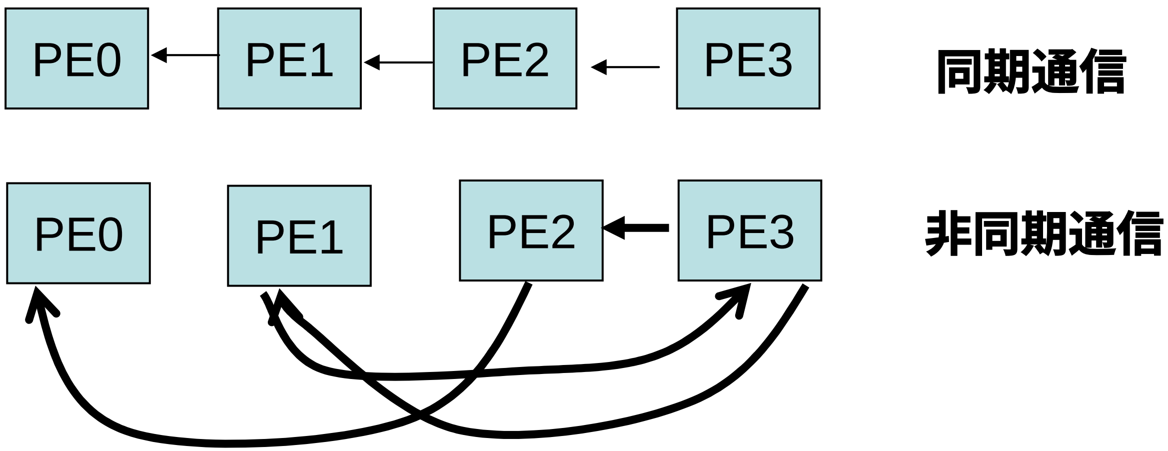
②SPMD化:Single　Program　Multiple　Data

　分岐点から合流点までをPEで非同期実行

③MIMD化

　　ＳＩＭＤは歴史的な役割を終えたのか？





３．２ＩＬＬＩＡＣ　ＩＶ

ILLIACIV：イリノイ大学で設計

　　　　　バロース社が開発

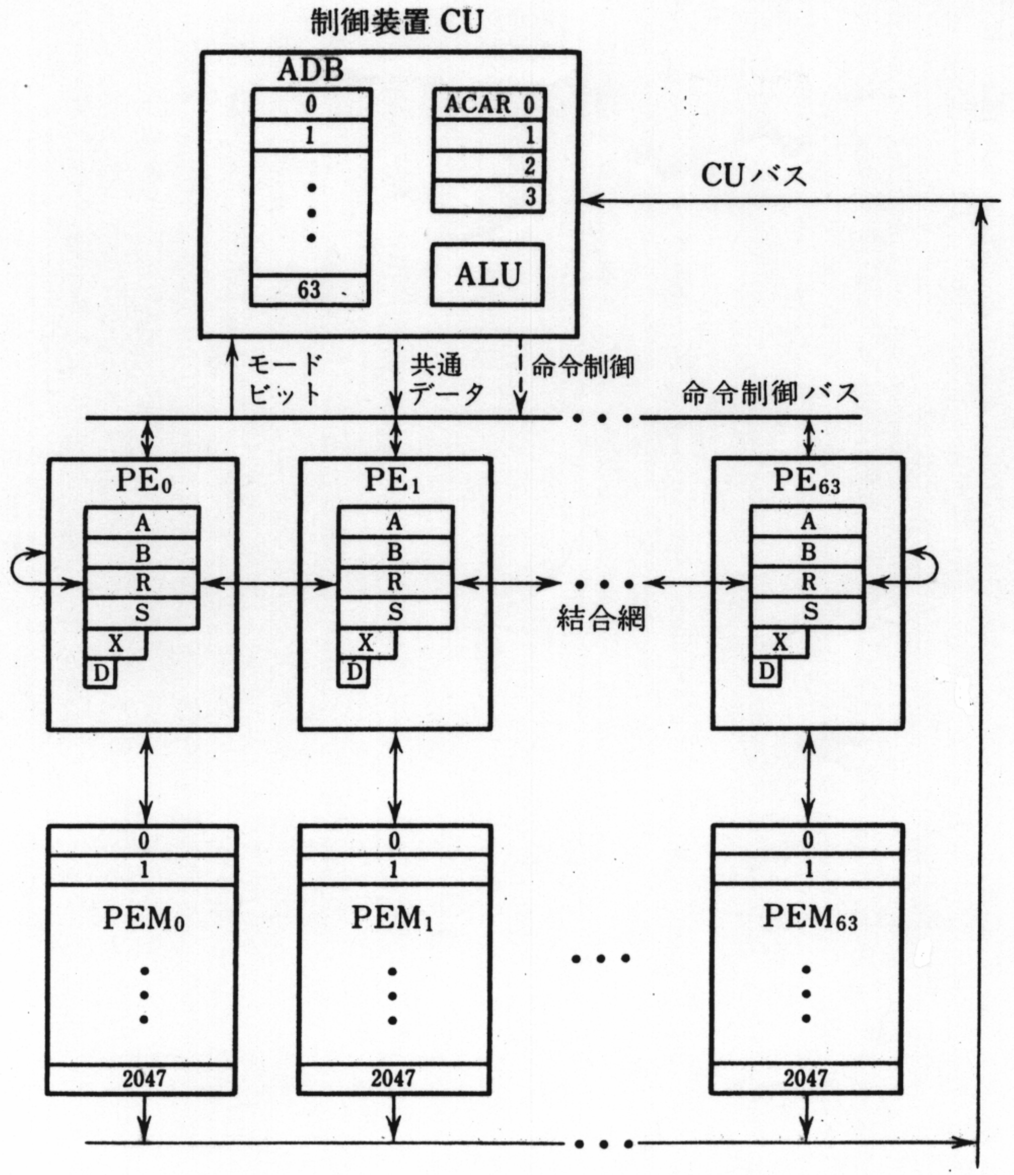
歴史的な並列コンピュータ

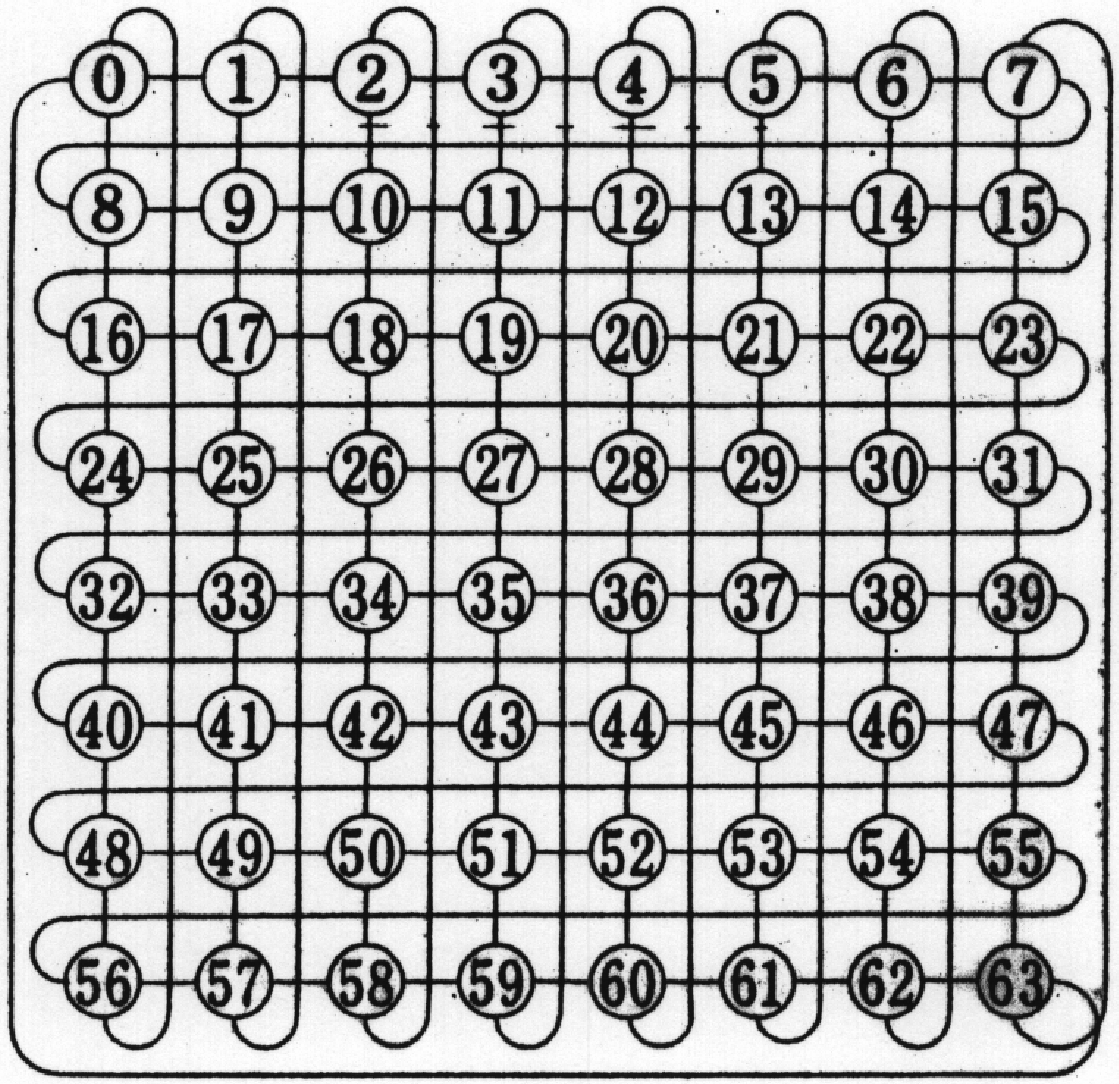
元々256台のPEの予定

64台の演算装置（PE）

疑似トーラス

大学紛争に巻き込まれた


制御装置 CU
ADB
0
1
63
ACAR 0
1
2
3
ALU
CU バス
モード
ビット
共通
データ
命令制御
命令制御バス
PE0
PE1
PE63
A
B
R
S
X
D
結合網
PEM0
PEM1
PEM63
0
1
2047


0 1 2 3 4 5 6 7
8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31
32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47
48 49 50 51 52 53 54 55
56 57 58 59 60 61 62 63

３．３　ＢＳＰ

（１）BSPの基本方式

・16台の同一構造をした演算装置PE
（浮動小数点加算・乗算：320nsec、除算：1.28μsec）

・メモリ共有方式
　　１６を超える最小素数１７台のメモリバンク

・クロスバスイッチ網

計画された2号機
　　512台のPE, 521台のメモリバンク

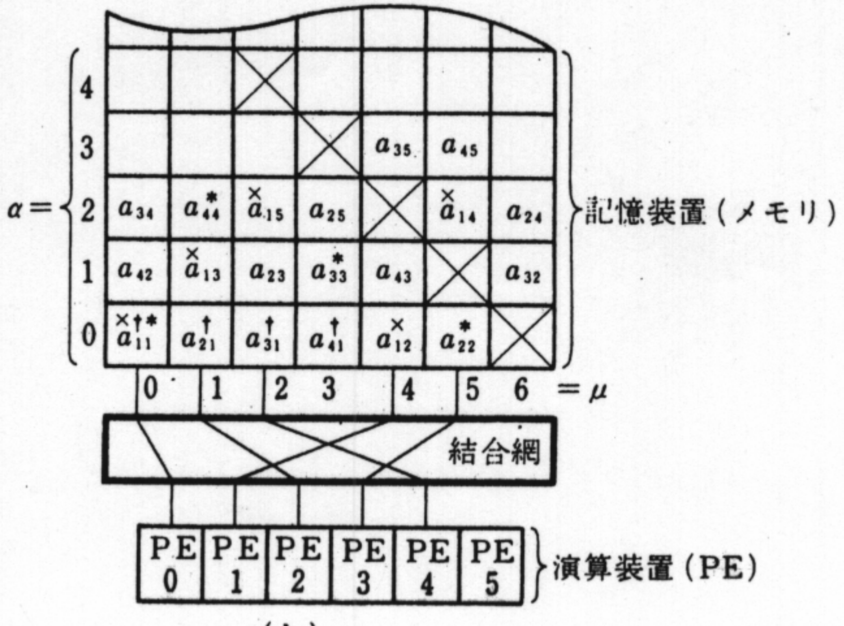
行列（4行5列）

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}

要素

	a_{11}	a_{21}	a_{31}	a_{41}	a_{12}	a_{22}	a_{32}	a_{42}	a_{13}	a_{23}	a_{33}	a_{43}	a_{14}	a_{24}	a_{34}	a_{44}	a_{15}	a_{25}	a_{35}	a_{45}
$a=$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$\mu=$	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5
$\alpha=$	0	0	0	0	0	0	1	1	1	1	1	1	2	2	2	2	2	2	3	3

（a）



（b）

（２）BSPのメモリ構成

Ｉ行Ｊ列の２次元配列A(i,j)

Aの要素$a_{i,j}$に与える１次元アドレス：

$$a=j*I+i+Base \qquad (1)$$

PE台数Pを越える最小の素数M

メモリバンク数

メモリアクセス

バンクアドレスμ(i、j)、

バンク内アドレスα(i,j)

$$\mu(i,j)=(j*I+i+Base) \bmod M \qquad (2)$$

$$\alpha(i、j)=\lfloor (j*I+i+\mathrm{Base})/P \rfloor \qquad (3)$$

線形Pベクトル

$$V(a,b,c,e)=\{A(i,j):i=aX+b, j=cX+e,\ o\leqq X<P\}$$

(4)

a=c=1,b=e=0**：行列**A**の対角要素**

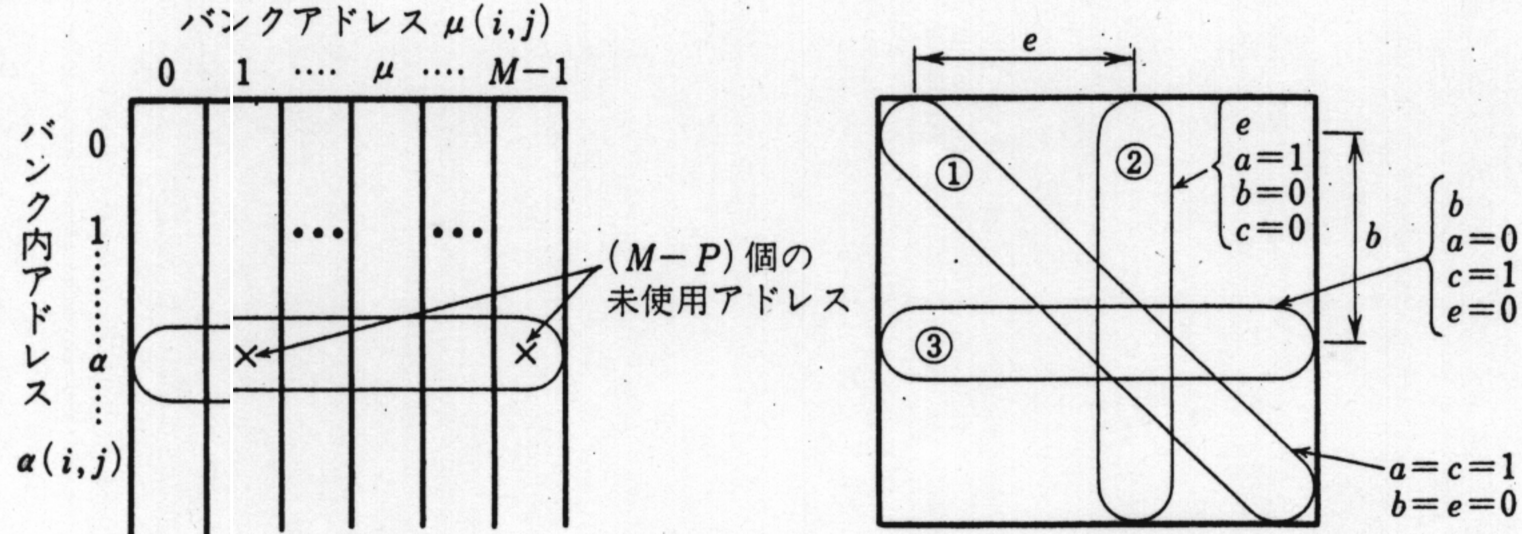
a=1,b=c=0**：列要素**

a=0、c=1,e=0**：行要素**

X**：線形**P**ベクトルの要素番号**

$$\mu(X)=(dX+B)\bmod M \qquad (5)$$

$$\alpha(X)=\lfloor (dX+B)/P \rfloor \qquad (6)$$



（a） 2次元アドレス

（b） 線形 P-ベクトル

ただし、$d=a+cI, B=b+eI+Base$ （７）

「定理」dとMが互いに素である時(Mは素数であるので、dがMの倍数でない時)、線形Pﾍﾞｸﾄﾙの全ての要素をP個のPEでﾊﾞﾝｸ競合なしに同時にｱｸｾｽできる。

「証明」いま、線型Ｐベクトルのある要素X_1とX_2が同一バンクに割り付けられたとすると、

$$X1 \neq X2$$

$$|X_1 - X_2| < P \qquad (8)$$

$$\mu(X_1) = \mu(X_2)$$

が成り立つ。（５）式より、

$$(dX_1+B)=m_1M+\mu(X_1)$$

$$(dX_2+B)=m_2M+\mu(X_2) \qquad （９）$$

（８）、（９）より、

$$d(X_1-X_2)=(m_1-m_2)M \qquad （１０）$$

Mは素数でかつd、Mは互いに素であるので、

$$|X_1-X_2|=|(m_1-m_2)/d|M>M \qquad （１１）$$

となり、（８）の前提に矛盾する。

（３）メモリアクセスと演算

①線形Pベクトルのパ ラメータa,b,c,eの値を決定

②PE番号Xとそれに対応する要素が格納されているメ

モリバンクμを対応

結合網の設定

③バンク内アドレスαを全てのバンクμについて並列に求め、

P個のデータの並列アクセスを行い、PEに転送

式（５）より、

$$dX+B=mM+\mu \quad (12)$$

両辺にd'(ただし､dd'=1mod M)を乗じて、

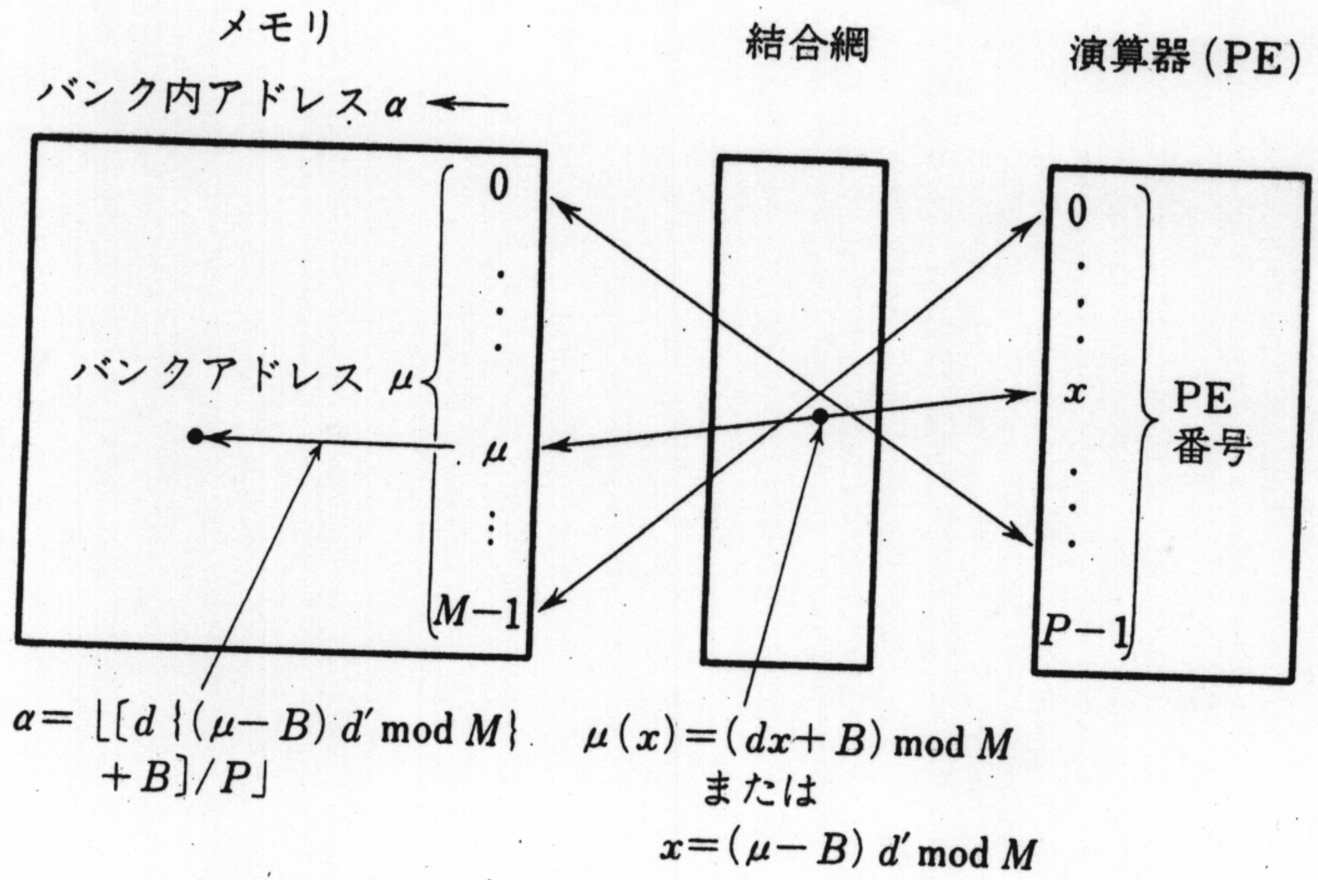
$$dd'X+d'B=md'M+d'\mu \quad (13)$$

これより、

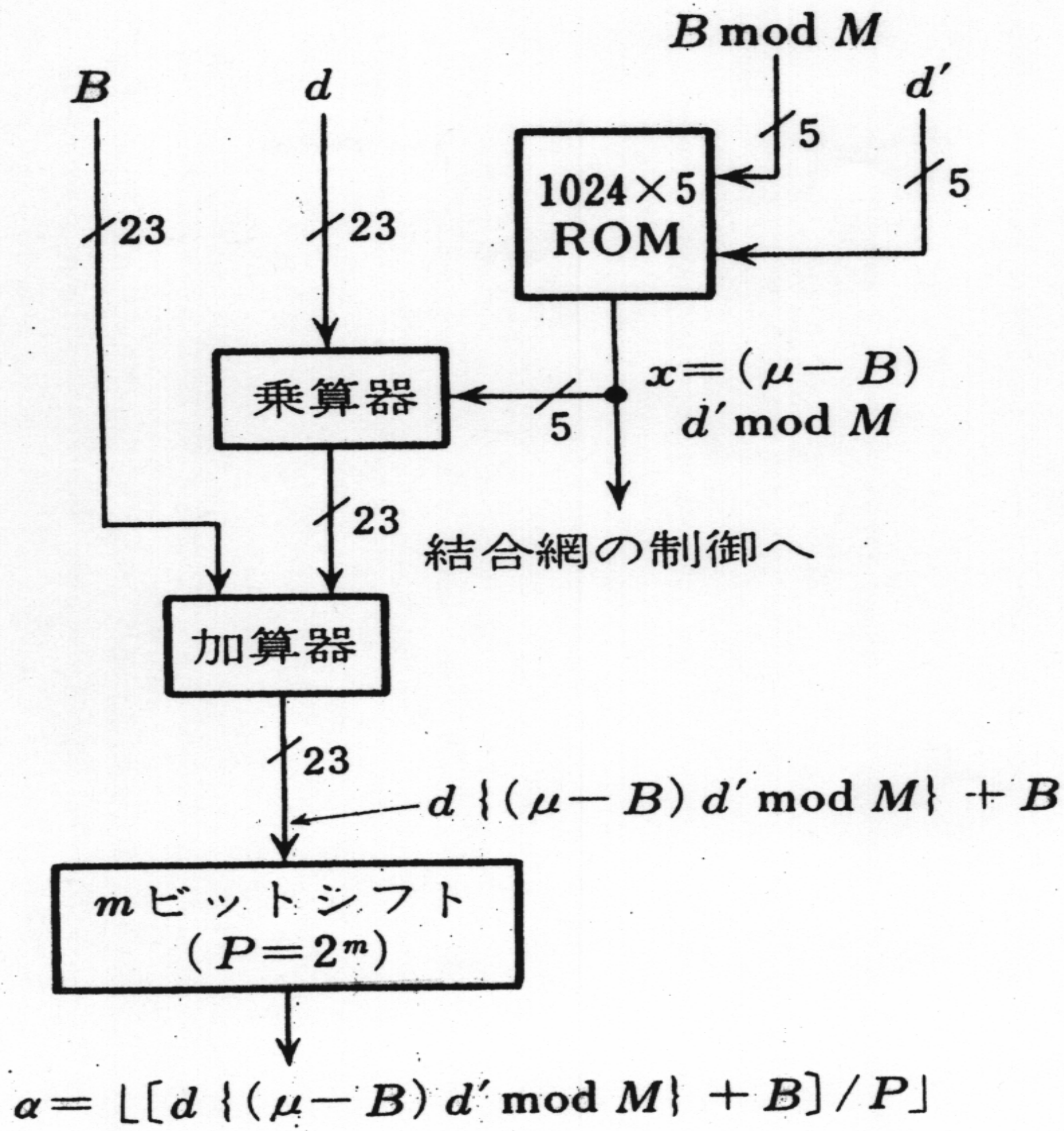
$$X = d'(\mu - B) \bmod M \tag{14}$$

（１４）を（６）に代入して、

$$\alpha(\mu) = \lfloor [d\{d'(\mu - B) \bmod M\} + B]/P \rfloor \tag{15}$$



メモリ
結合網
演算器 (PE)
バンク内アドレス α
0
バンクアドレス μ
μ
$M-1$
0
x
PE
番号
$P-1$
$\alpha = \lfloor [d\{(\mu - B)\, d' \bmod M\} + B]/P \rfloor$
$\mu(x) = (dx + B) \bmod M$
または
$x = (\mu - B)\, d' \bmod M$



$B \bmod M$
B
d
d'
5
5
23
23
1024×5
ROM
乗算器
5
$x=(\mu-B)$
$d' \bmod M$
23
結合網の制御へ
加算器
23
$d\{(\mu-B)\,d' \bmod M\}+B$
mビットシフト
$(P=2^m)$
$\alpha=\lfloor[d\{(\mu-B)\,d' \bmod M\}+B]/P\rfloor$

P=4, M=5の時、4x4行列を格納すると

$a_{11},a_{21},a_{31},a_{41},a_{12},a_{22},a_{32},a_{42},a_{13},a_{23},a_{33},a_{43},a_{14},a_{24},a_{34},a_{44}$

a	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
μ	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0
α	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3

対角要素： 競合

d＝5、M＝5！！

３．４　ＳＴＡＲＡＮ

（１）STARANの基本構成

大規模連想メモリ

（２）MDAの構成

MDA：通常のRAMを256個用いて構成

排他的論理和を多用した直交メモリ

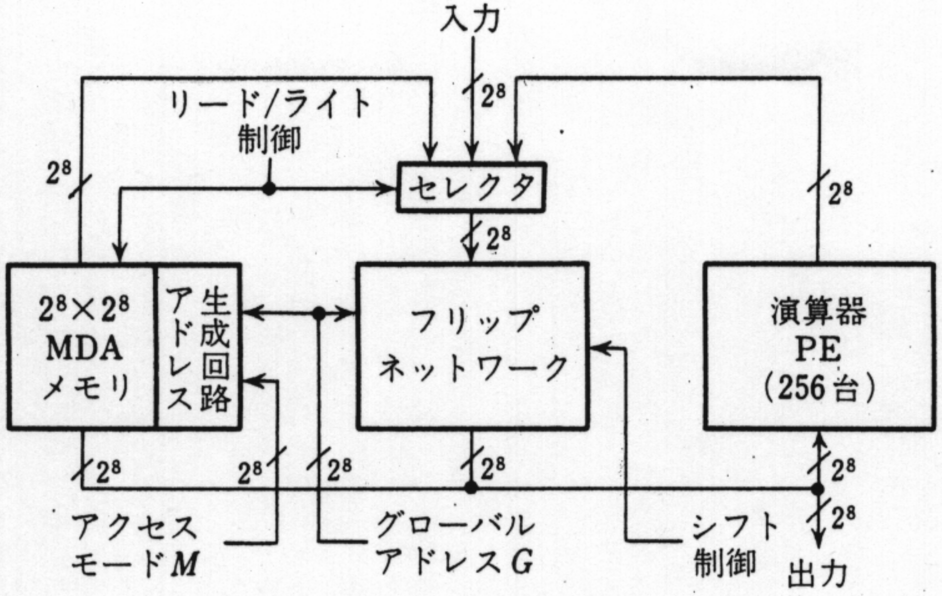
$A = B \oplus C \Rightarrow B \oplus A = B \oplus (B \oplus C) = (B \oplus B) \oplus C = C$

逆関数が容易に求められる

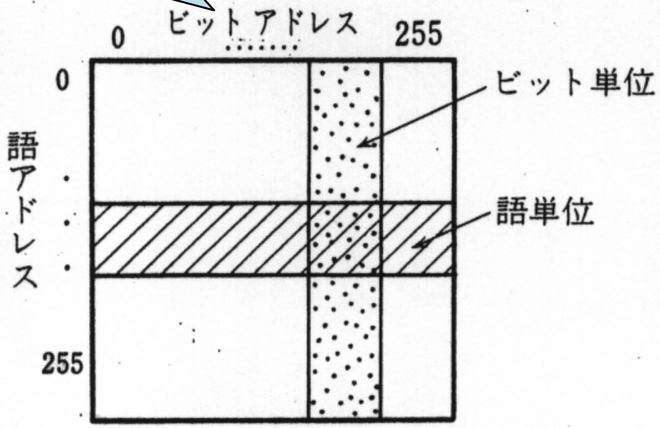
図3.9：8x8のMDA

$a_{B,W}$：語WのビットＢ

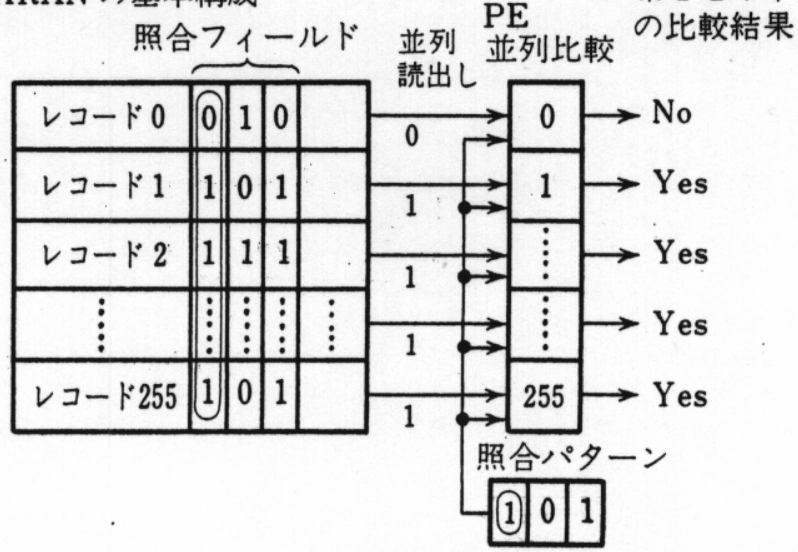
直交メモリ



(a) STARANの基本構成



(b) MDAの構成



(c) STARANの連想処理装置としての利用

$a_{B,W}$とメモリチップ番号Cの

チップ内アドレスBとの対応

ストレージルール：

$$C=B \oplus W \tag{16}$$

$$W=B \oplus C \tag{17}$$

$\oplus$：排他的論理和

メモリチップCに対するチップ内アドレス生成規則

アドレッシングルール：

$$B=G \oplus (MC) \tag{18}$$

M：アクセスモード

G：グローバルアドレス

メモリチップC	0	1	2	3	4	5	6	7
0	a_{00}	a_{11}	a_{22}	a_{33}	a_{44}	a_{56}	a_{66}	a_{77}
1	a_{01}	a_{10}	a_{23}	a_{32}	a_{45}	a_{54}	a_{67}	a_{76}
2	a_{02}	a_{13}	a_{20}	a_{31}	a_{46}	a_{57}	a_{64}	a_{75}
3	a_{03}	a_{12}	a_{21}	a_{30}	a_{47}	a_{56}	a_{65}	a_{74}
4	a_{04}	a_{15}	a_{26}	a_{37}	a_{40}	a_{51}	a_{62}	a_{73}
5	a_{05}	a_{14}	a_{27}	a_{36}	a_{41}	a_{50}	a_{63}	a_{72}
6	a_{06}	a_{17}	a_{24}	a_{35}	a_{42}	a_{53}	a_{60}	a_{71}
7	a_{07}	a_{16}	a_{25}	a_{34}	a_{43}	a_{52}	a_{61}	a_{70}

チップ内アドレスB

$a_{B,W}$＝語WのビットB

式（１７）、（１８）より、

$$W=G\oplus(\sim MC) \qquad （１９）$$

メモリチップ番号CとPE番号Pの対応関係

スクランブルルール：　$C=G\oplus P$ （２０）

アンスクランブルルール：　$P=G\oplus C$ （２１）

アクセスルール：　$W=(MG)\oplus(\sim MP)$ （２２）

$B=(\sim MG)\oplus(MP)$

$M=(1,1,...,1)$、Gとして語アドレス

語GのビットPがPE番号Pでアクセス：

語単位アクセス

証明

$$A \oplus B = A\overline{B} + \overline{A}B$$

$$\left.\begin{aligned}\overline{A + B} &= \overline{A}\,\overline{B} \\ \overline{AB} &= \overline{A} + \overline{B}\end{aligned}\right\} \text{ドモルガンの双対定理}$$

$$W = (G \oplus MC) \oplus C = G \oplus (MC \oplus C) =$$

$$G \oplus (MC\overline{C} + \overline{MC}C) = G \oplus ((\overline{M} + \overline{C})C)$$

$$= G \oplus \overline{M}C$$

$$W = G \oplus (\overline{M}(G \oplus P)) = G\overline{\overline{M}(G \oplus P)} + \overline{G}\,\overline{M}(G \oplus P)$$

$$= G(M + \overline{G \oplus P}) + \overline{G}(\overline{M}(G\overline{P} + \overline{G}P))$$

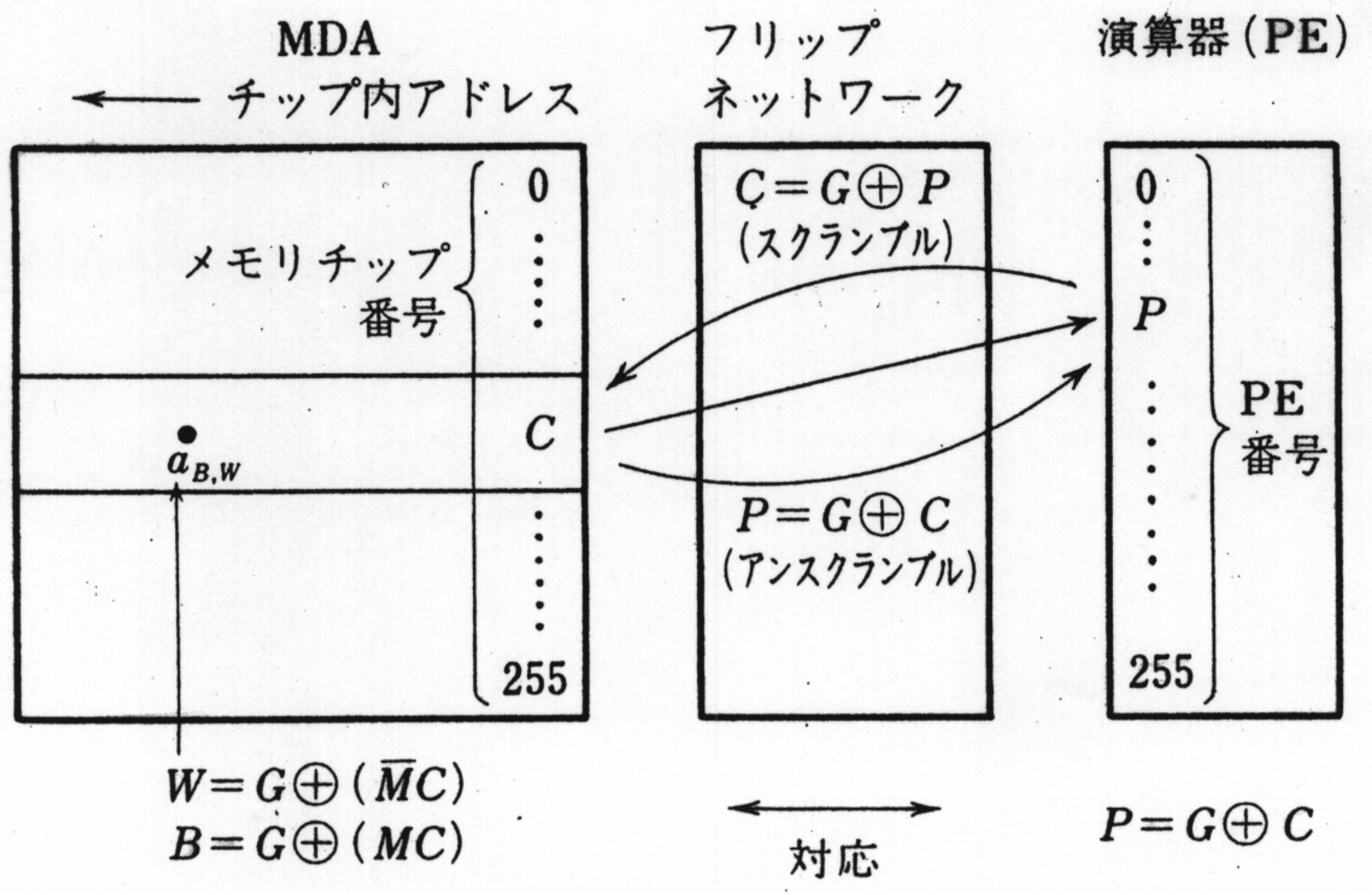
$$= GM + G(\overline{\overline{G}P + G\overline{P}}) + \overline{G}\,\overline{M}P$$

$$= GM + G(G + \overline{P})(\overline{G} + P) + \overline{G}\,\overline{M}P = GM + GP + \overline{G}\,\overline{M}P$$

$$= GM + GMP + G\overline{M}P + \overline{G}\,\overline{M}P = GM + G\overline{M}P + \overline{G}\,\overline{M}P$$

$$W = (MG \oplus \overline{M}P) = MG(M + \overline{P}) + (\overline{M} + \overline{G})(\overline{M}P)$$

$$= GM + \overline{M}P + \overline{G}\,\overline{M}P = GM + G\overline{M}P + \overline{G}\,\overline{M}P + \overline{G}\,\overline{M}P$$



MDA
← チップ内アドレス
メモリチップ
番号
0
C
255
$a_{B,W}$
$W=G\oplus(\overline{M}C)$
$B=G\oplus(MC)$
フリップ
ネットワーク
$C=G\oplus P$
(スクランブル)
$P=G\oplus C$
(アンスクランブル)
対応
演算器(PE)
0
P
255
PE
番号
$P=G\oplus C$

M=(0,0,..,0)、Gとしてビットアドレス

語PのビットGがPE番号Pでアクセス：

ビット単位アクセス

（３）フリップネットワークの構成

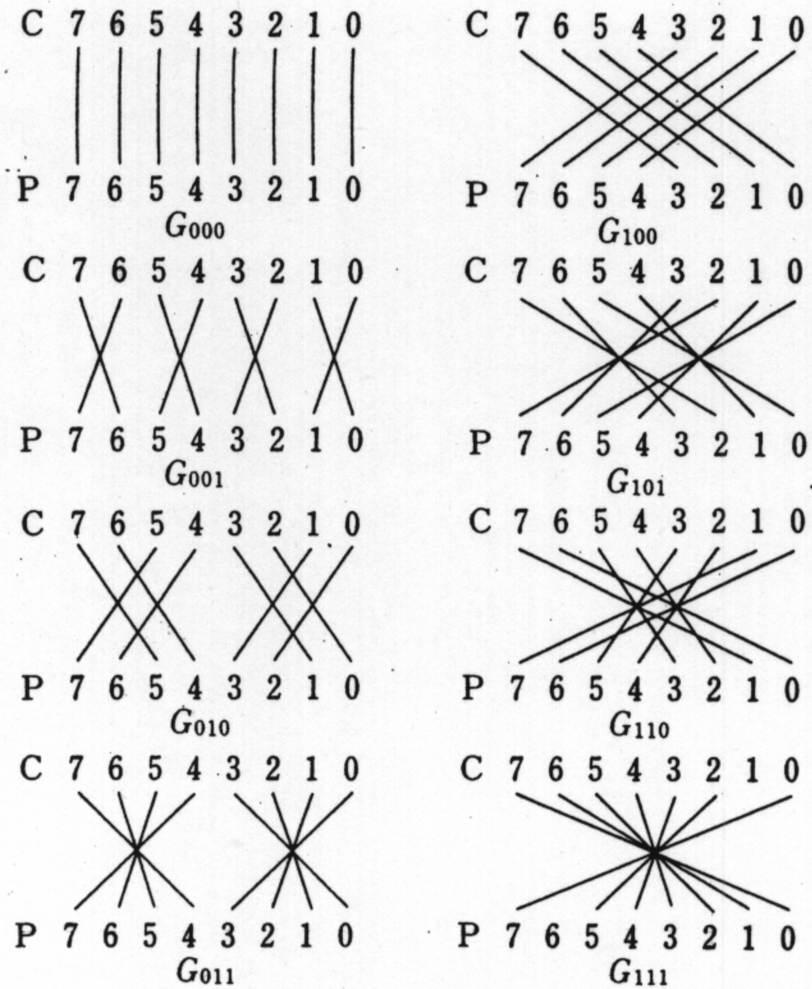
写像：P=G⊕C

G:G_{a2a1a0}

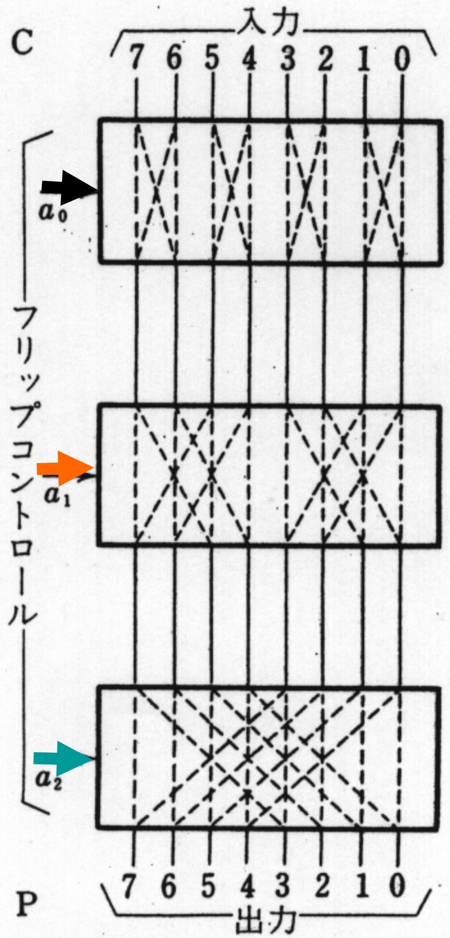
$$G_{a2a1a0}=a_2G_{100}\oplus a_1G_{010}\oplus a_0G_{001}$$

$$P=G\oplus C= a_2G_{100}\oplus a_1G_{010}\oplus a_0G_{001}\oplus C$$

間接２進ｎーキューブ



（a） フリップネットワークによる入出力間の論理的結合



（b） 多段結合網によるフリップネットワークの実現

３．５　ＧＦ－１１

IBM社のT.J.Watson研究所

576台のPE、

Memphis網

（１）各PE：

４台の浮動小数点演算器、

１台の固定小数点演算器、

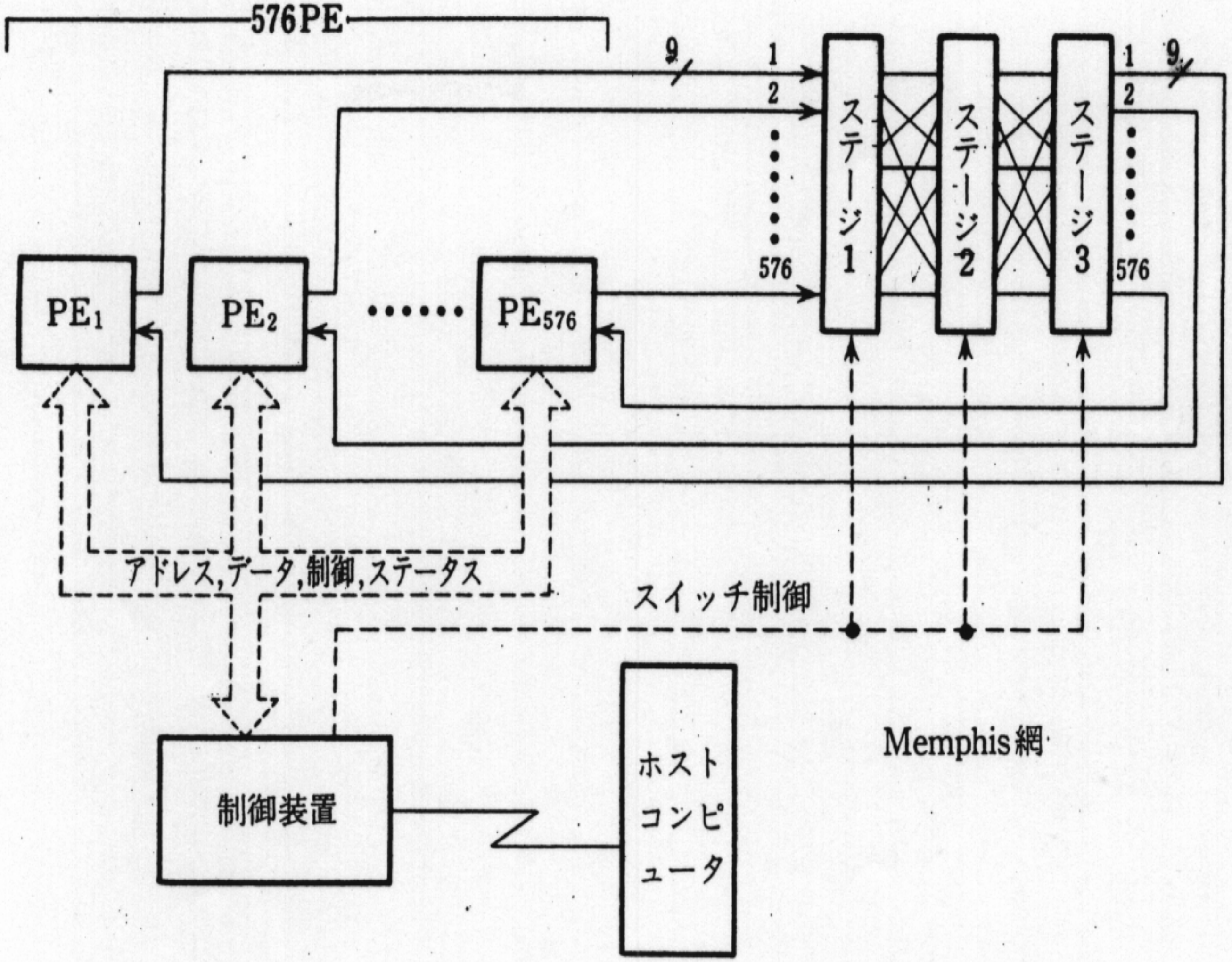
256語のレジスタファイル

180ビット長の命令によって同時に制御（VLIW）

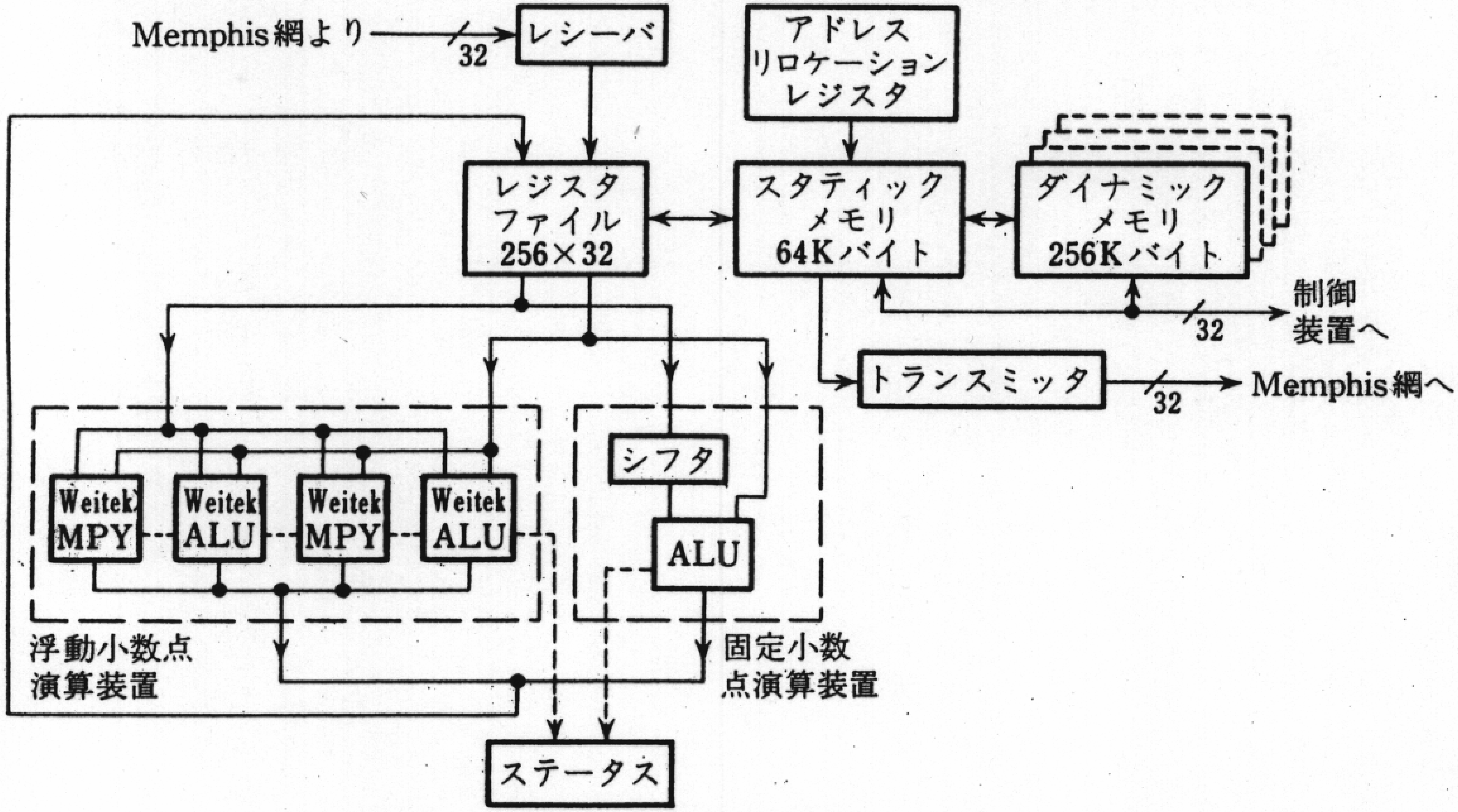
（２）Memphis網

３ステージClos網（n=m）

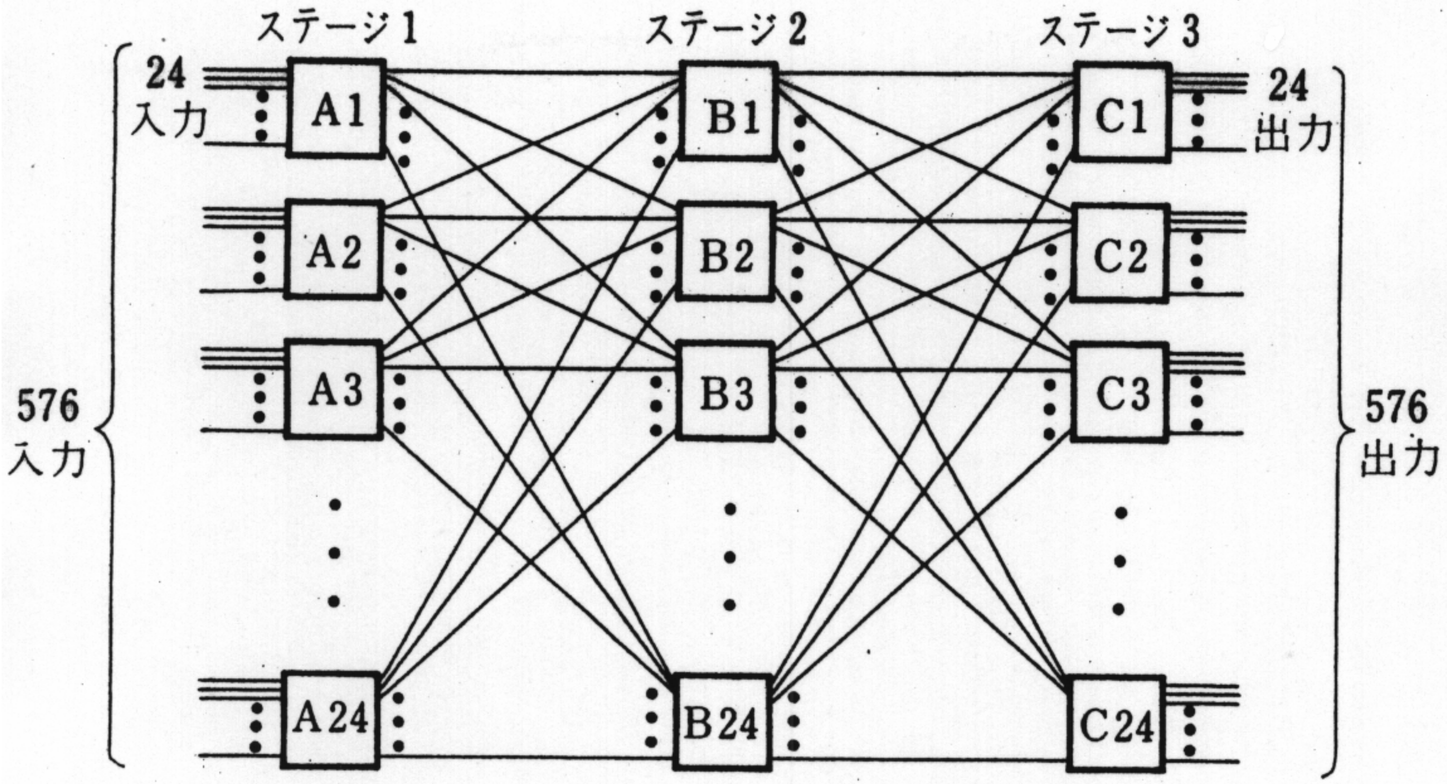
２４ｘ２４のクロスバ網を利用


576PE
9
1
2
576
ステージ1
ステージ2
ステージ3
1
2
576
9
PE1
PE2
······
PE576
アドレス,データ,制御,ステータス
スイッチ制御
制御装置
ホストコンピュータ
Memphis網

（a） GF-11の構成


Memphis網より
32
レシーバ
アドレス
リロケーション
レジスタ
レジスタ
ファイル
256×32
スタティック
メモリ
64Kバイト
ダイナミック
メモリ
256Kバイト
制御
装置へ
32
トランスミッタ
32
Memphis網へ
Weitek
MPY
Weitek
ALU
Weitek
MPY
Weitek
ALU
シフタ
ALU
浮動小数点
演算装置
固定小数
点演算装置
ステータス

（b） PEの構成



（c） Memphis網の構成

各ボックスは24×24のクロスバースイッチ
各入出力線は9ビット幅（8ビットデータ，1ビットパリティ）

３．６ＣＭ－１

従来のコンピュータシステム：メモリは受動的

IBMのJ.Backus：フォンノイマンボトルネック

ロジックインメモリの考え方

メモリの中に論理を持ち込み、

処理をメモリ側で分担する方式

（１）Connection Machine の構成

相互結合網：2進12キューブ

各ノードに１台のルータが結合

16台のセルが結合（セル総数は65536台）

（２）Connection Machine の基本演算操作

プログラミング言語Lisp

基本データ構造：Xector（ゼクタ）

Xector：

（インデックス-値）ペアからなる要素の集合

{sky-blue grass-green apple-red}

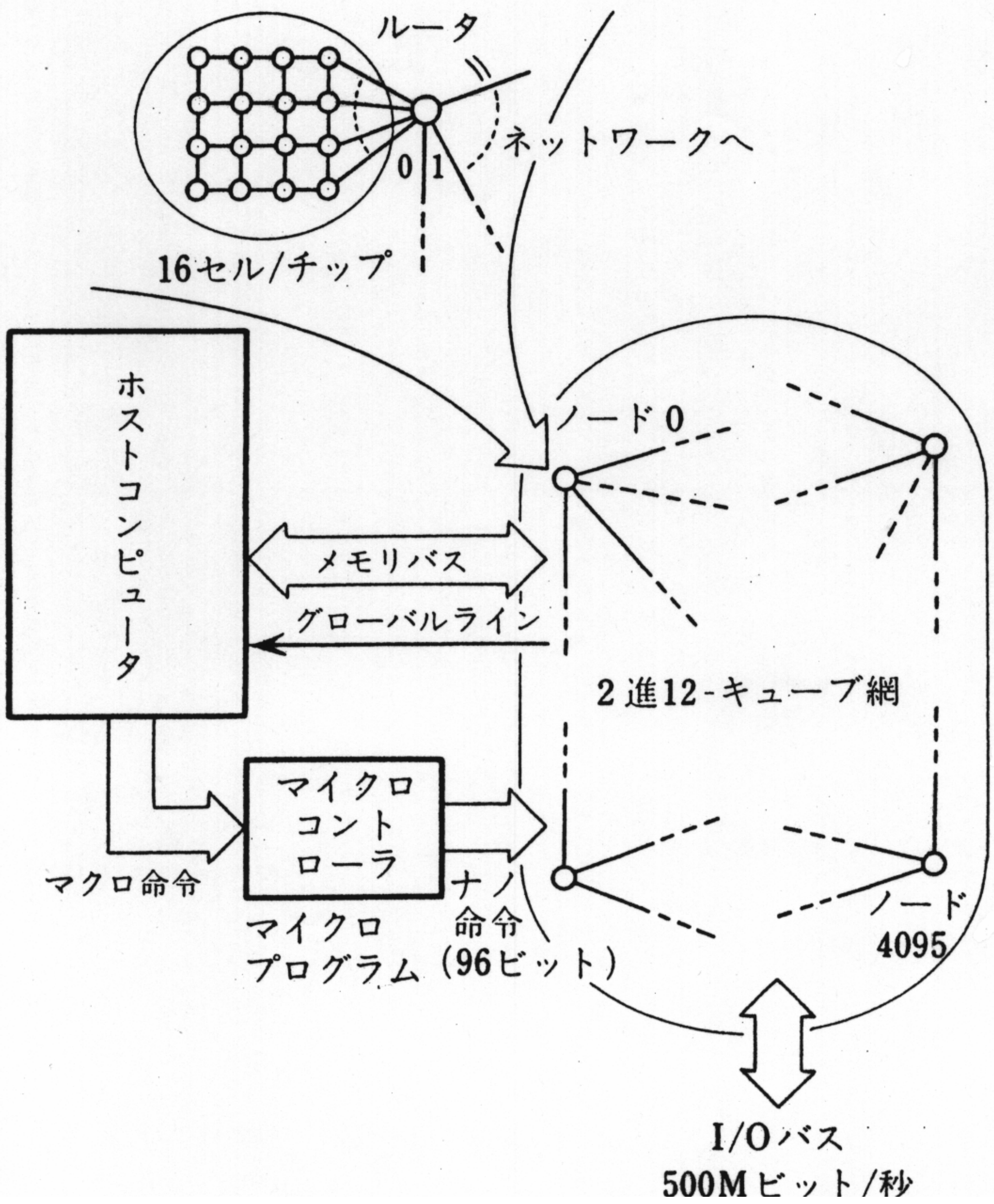
並列演算：α、β操作

α操作：各Xector内の同一インデックスを含む要素に独立に作用

(α+'{A-1 B-2}'{A-3 B-3})={A-4 B-5}

β操作：Xector要素の値に作用して、
一つの値を生成
(β+'{A-1 B-2 C-3})=6

α、β操作の組み合わせ
集合演算（積集合の計算など）、木構造演算（木構造要素の和）、バタフライ構造演算（バイトニックソータなど）、文字列演算、サーチ演算、配列演算（画像処理の空間フィルタなど）



ルータ
ネットワークへ
0 1
16セル/チップ
ホストコンピュータ
ノード0
メモリバス
グローバルライン
2進12-キューブ網
マイクロコントローラ
マクロ命令
マイクロプログラム
ナノ命令(96ビット)
ノード4095
I/Oバス
500Mビット/秒

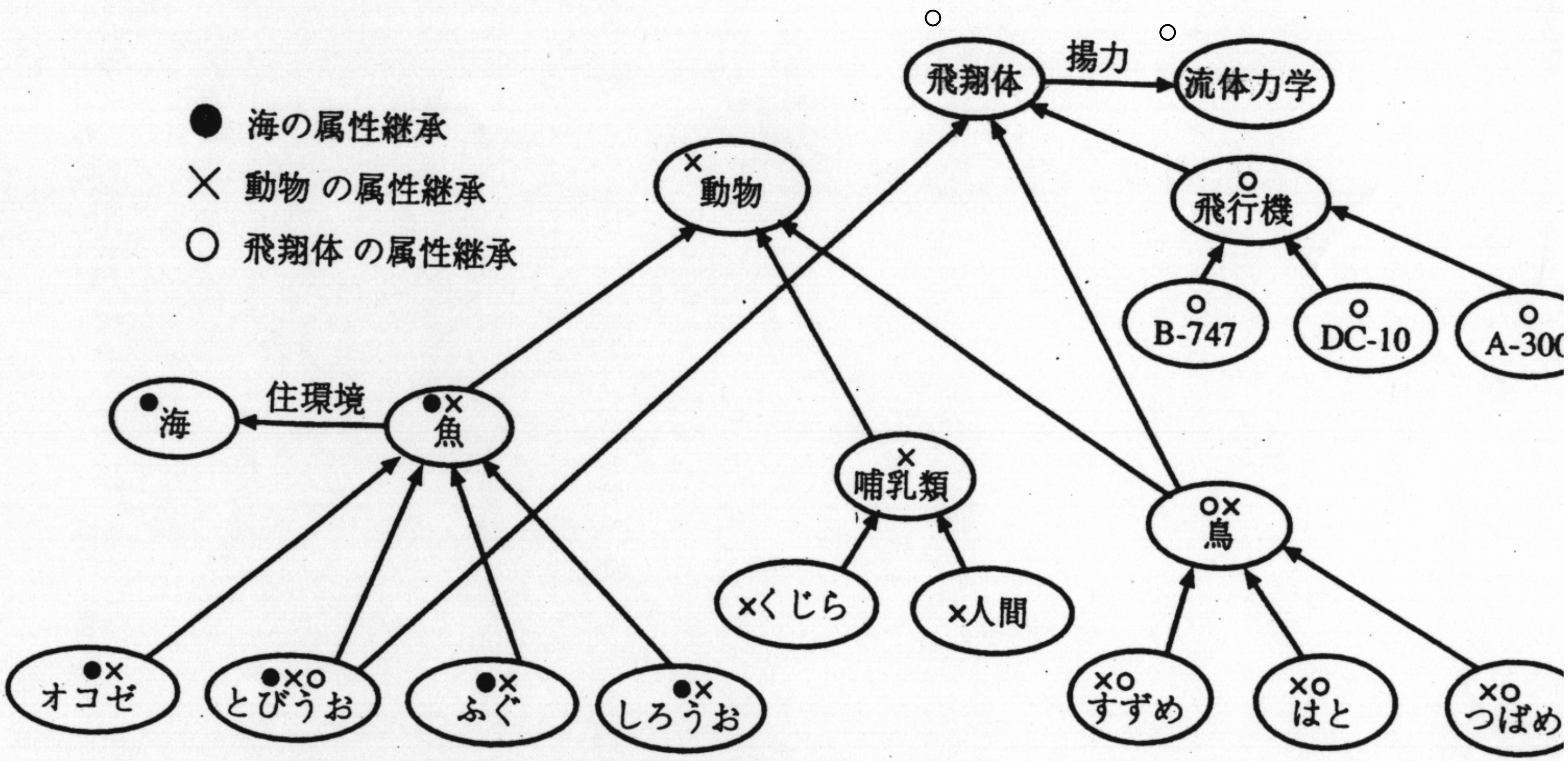


図 6. 43　意味ネットワーク

MPP

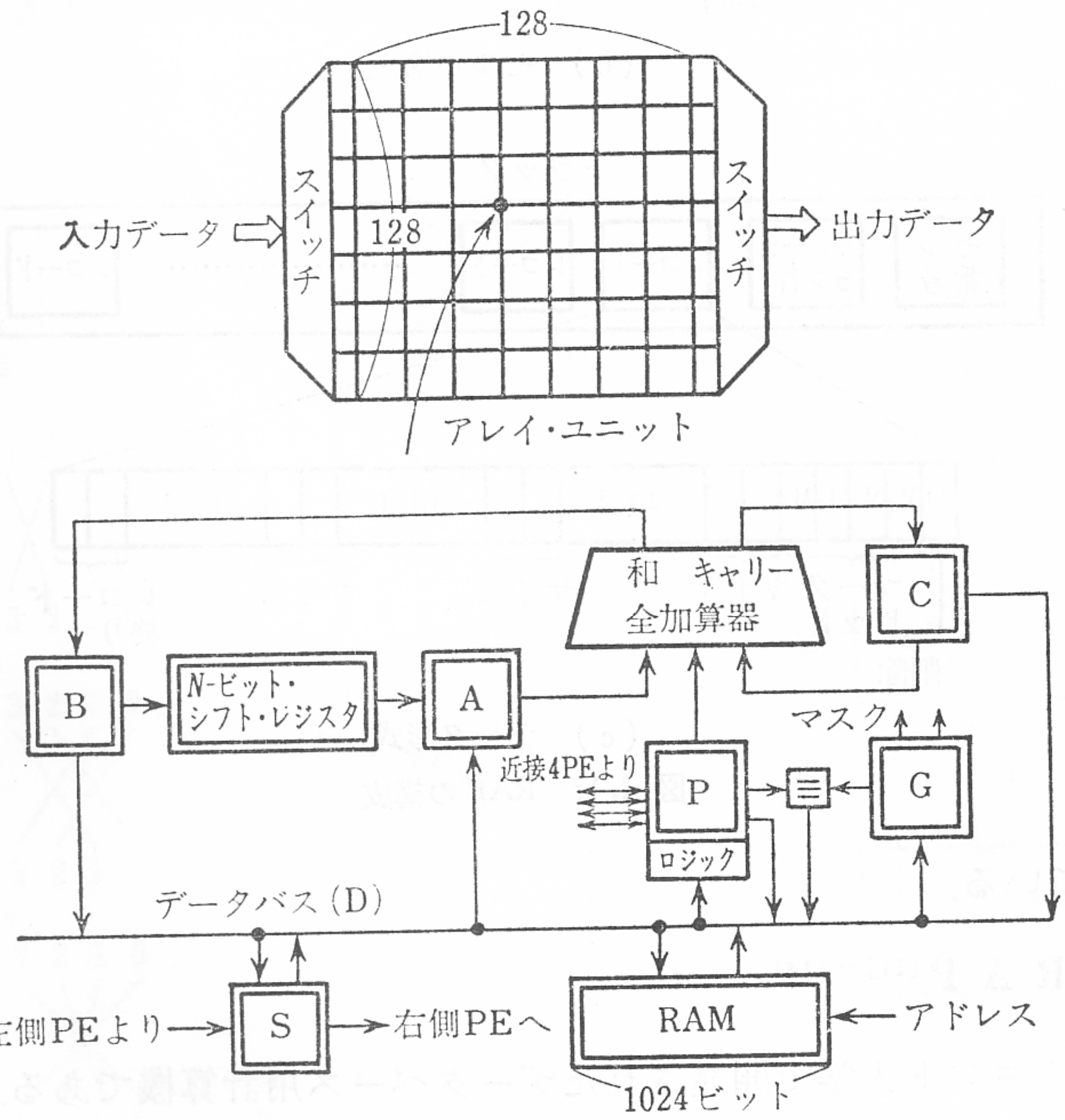


図 4.18 MPP の構成(K. E. Batcher: Design of a Massively Parallel Processor, *IEEE Trans.* C. Vol. 29, No. 9, 1980, pp. 836-840 による)

フィルタ処理：輪郭線抽出など

ラプラシアンオペレータ

$$\nabla^2 \varphi = \partial^2 \varphi / \partial x^2 + \partial^2 \varphi / \partial y^2$$

$$= \varphi(I+1, J) - \varphi(I, J) - (\varphi(I, J) - \varphi(I-1, J)) +$$
$$\varphi(I, J+1) - \varphi(I, J) - (\varphi(I, J) - \varphi(I, J-1))$$

$$= \varphi(I+1, J) + \varphi(I-1, J) + \varphi(I, J+1) + \varphi(I, J-1) - 4\varphi(I, J)$$

	1	
1	−4	1
	1	

(a) Original photograph

(b) Printout of the digital gray-level picture

(c) Binary picture

Figure 3-2 Picture input and line extraction. The dark horizontal line in the upper part is due to the burn in the CRT surface of the FSS used for digitization.

SIMD計算機は歴史的な役割を終えたか？

汎用並列コンピュータ：YES

　　マルチプロセッサが席巻

専用コンピュータ

　　画像処理：　光の直接入力、演算ビット幅

　　ニューラルネット：多対多ブロードキャスト

機能メモリ

　　DRAMセルに演算器

MMXやSSEなどマルチメディア命令

$$f(x) = \frac{1}{1 + exp(-x + \theta)}$$

$$f'(x) = f(x)(1 - f(x))$$

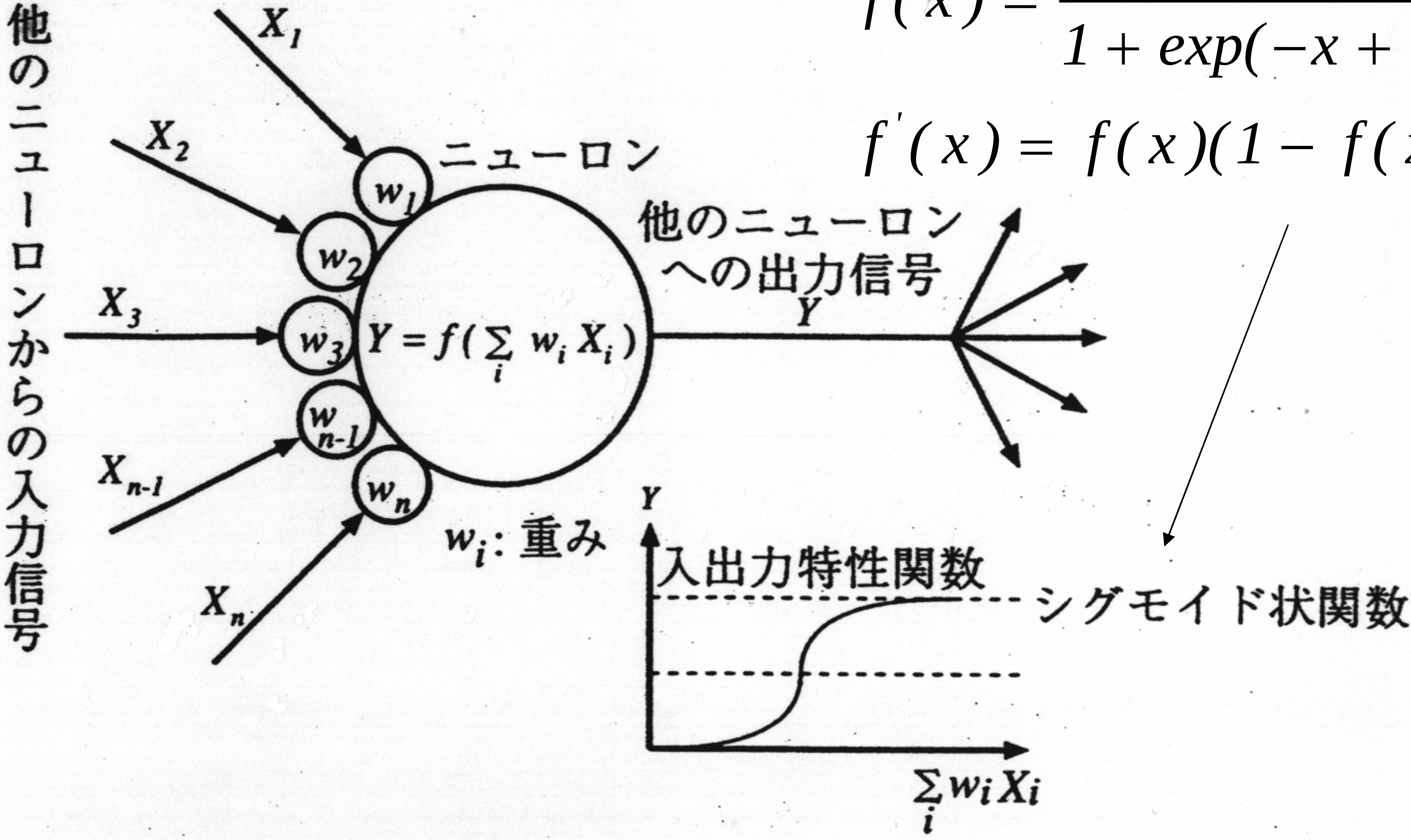


図 6. 47　ニューロンモデル

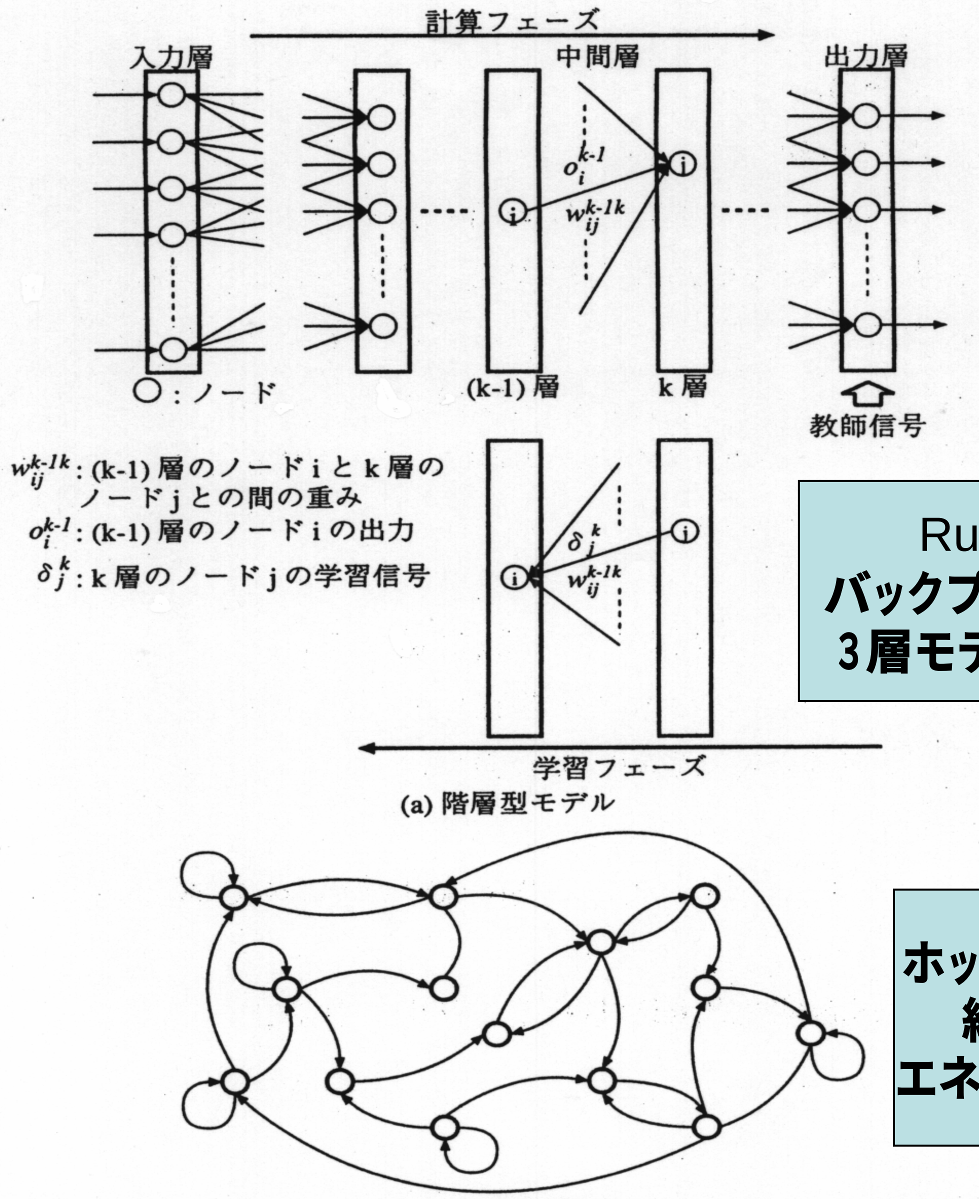


図 6. 48　ネットワークモデル

Rumelhartのバックプロパゲーション
3層モデル、学習機械

ホップフィールドモデル
組合わせ問題→
エネルギ関数の最小化

マルチメディア対応の機械命令

Intel社

①ＭＭＸ：画像処理，整数演算

６４ビットデータに対する

サブワード演算：

8bx8、16bx4など

飽和演算

カラー画像

R,G,B：各８ビット

表 5.4 Intel マルチメディア向き命令 (MMX)

Paddb など	8 つのバイトデータ(b), 4 つの 16 ビットデータ(w), 2 つの 32 ビットデータ(d)に対する並列加減算
Pcompeqb など	8 つのバイトデータ, 4 つの 16 ビットデータ, 2 つの 32 ビットデータに対する並列比較演算
Pmullw など	4 つの 16 ビットデータ(w)に対する並列乗算
Pmaddwd など	4 つの 16 ビットデータについて, 2 つの隣接 16 ビットデータの積の和
Psraw など	4 つの 16 ビットデータ, 2 つの 32 ビットデータ, 1 つの 64 ビットデータに対する論理/算術, 左右シフト演算
Punpcktbw など	パックされている 8 つのバイトデータ, 4 つの 16 ビットデータ, 2 つの 32 ビットデータをアンパックする.
Pactsswb など	16 ビットデータ, 32 ビットデータを 8 ビットデータ, 16 ビットデータにパック
Pand など	64 ビットデータの論理(and, or, xor などを含む)操作
Movd など	32 ビットデータまたは 64 ビットデータのレジスターメモリ, レジスターレジスタ間転送

フィルタ処理：輪郭線抽出など

ラプラシアンオペレータ

$$
\begin{aligned}
\nabla^2 \varphi &= \partial^2 \varphi / \partial x^2 + \partial^2 \varphi / \partial y^2 \\
&= \varphi(I+1, J) - \varphi(I, J) - \\
&\quad (\varphi(I, J) - \varphi(I-1, J)) + \\
&\quad \varphi(I, J+1) - \varphi(I, J) - \\
&\quad (\varphi(I, J) - \varphi(I, J-1)) \\
&= \varphi(I+1, J) + \varphi(I-1, J) \\
&\quad + \varphi(I, J+1) + \varphi(I, J-1) \\
&\quad - 4\varphi(I, J)
\end{aligned}
$$

	1	
1	−4	1
	1	

(a) Original photograph

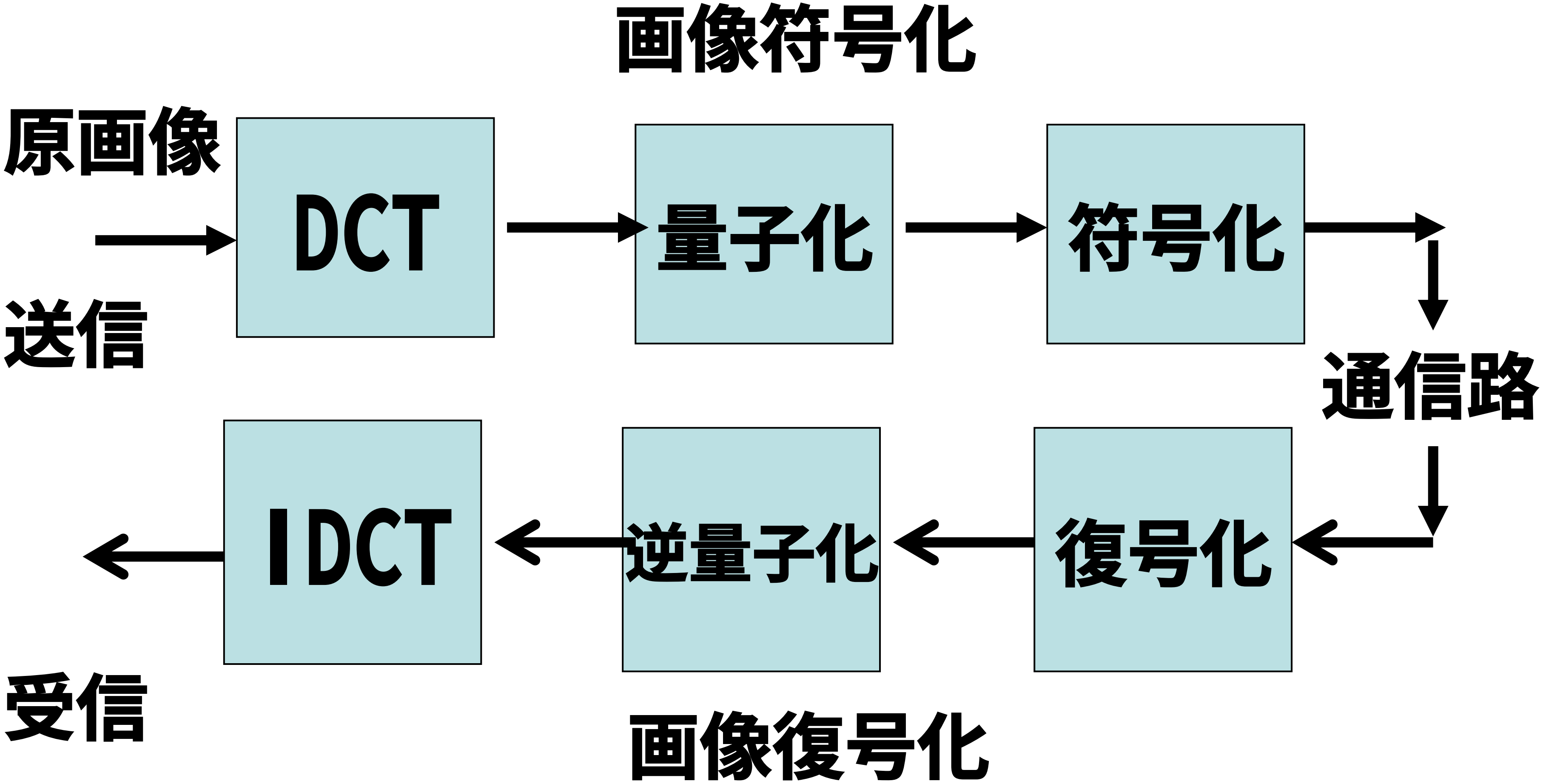
(b) Printout of the digital gray-level picture

(c) Binary picture

Figure 3-2 Picture input and line extraction. The dark horizontal line in the upper part is due to the burn in the CRT surface of the FSS used for digitization.

画像符号化･復号化

DCT：離散コサイン変換



DCT変換（１次元）の証明

安田、藤原監訳

情報圧縮技術、bit別冊、共立、１９９７

正規化 DCT

$$X^{c2}(k) = \sqrt{\frac{2}{N}}\, c_k \sum_{n=0}^{N-1} x(n) \cos\left[\frac{(2n+1)k\pi}{2N}\right], \qquad (5.11)$$

$$k = 0, 1, \ldots, N-1$$

$$\Updownarrow$$

正規化 IDCT

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} c_k X^{c2}(k) \cos\left[\frac{(2n+1)k\pi}{2N}\right],$$

$$n = 0, 1, \ldots, N-1$$

$$\cos\{(2n+1)k\pi/(2N)\}$$

	n=0	1	N/4-1	N/4	N/2-1	N/2	3/4N-1	3/4N	N-1
K=0	1	1	1	1	1	1	1	1	1

K=1　1/(2N) 3/(2N)・・・・・・・・・・・・・・・・・・・・・・1/2-1/(2N)　1/2+1/(2N)・・・・・・・・・・・・・・・・・・・・・・1-1/(2N)

K=2　1/N　3/N・・・(1/2-1/N)　(1/2＋1/N)・・・(1-1/N)　(1+1/N)・・・(3/2-1/N)　(3/2+1/N)・・・(2-1/N)

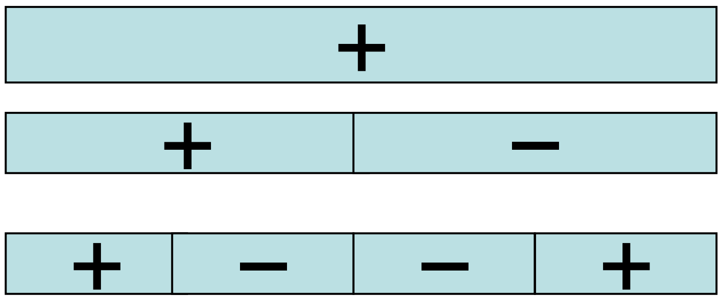
・・・・・・・・・・・・・・・・・

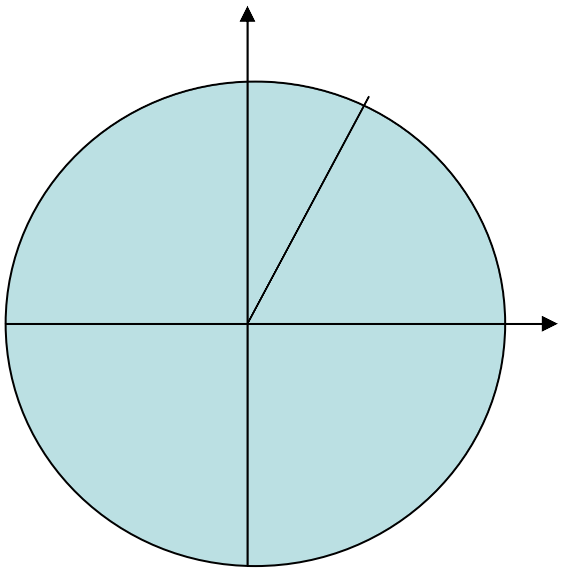
K=N-1

n=0　n=1　n=2　n=3

1/2-1/(2N)　3/2-3/(2N)　1/2-5/(2N)　3/2-7/(2N)

Kが小さいほど低周波領域





基底ベクトル

$$c(k,n)=1/\sqrt{N} : k=0$$

$$c(k,n)=\sqrt{2/N}\cos\frac{(2n+1)k\pi}{2N} : k\neq 0$$

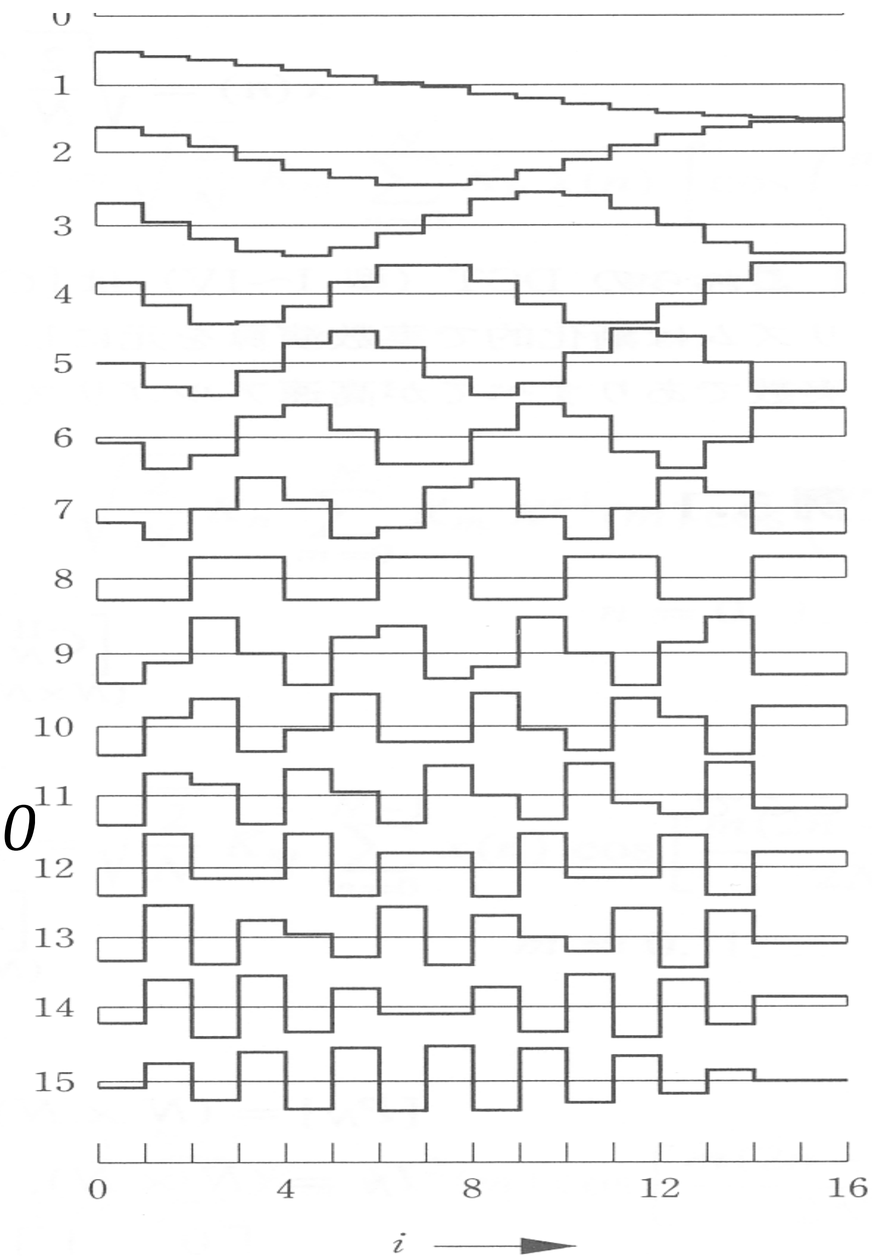


図 5.1　DCT-II の基底関数 $N = 16$

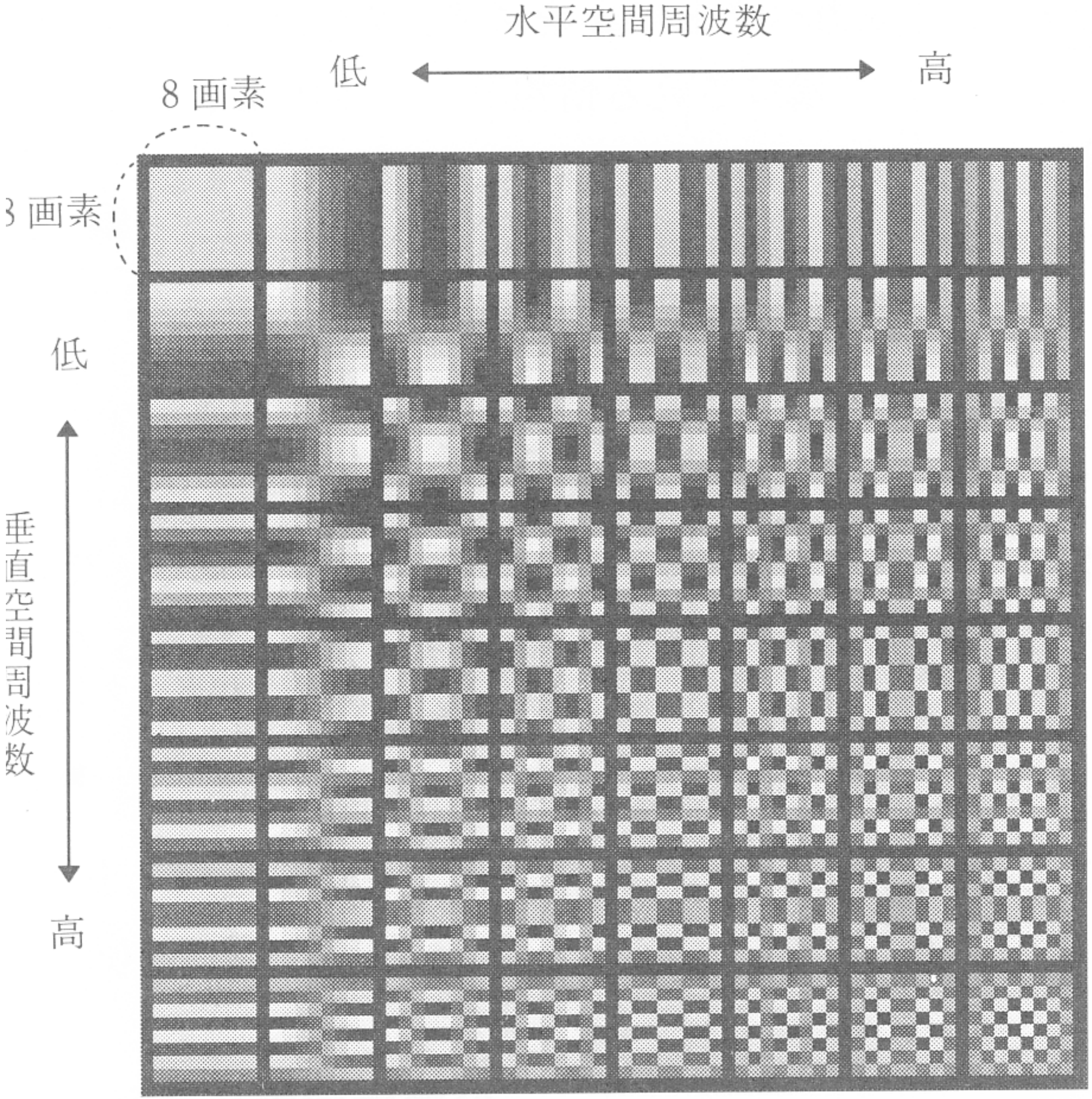


図 4.13　2 次元 DCT の基底画像パターン（8×8 の場合）

$$x_n = \frac{2}{N}\sum_{k=0}^{N-1} c_k^2 \sum_{m=0}^{N-1} x_m \cos\left[\frac{(2m+1)k\pi}{2N}\right]\cos\left[\frac{(2n+1)k\pi}{2N}\right]$$

$$= \frac{2}{N}\sum_{m=0}^{N-1} x_m \sum_{k=0}^{N-1} c_k^2 \cos\theta_m \cos\theta_n \tag{5.25}$$

ただし，

$$\theta_m = \frac{(2m+1)k\pi}{2N}, \quad \theta_n = \frac{(2n+1)k\pi}{2N}$$

$$\cos\theta_m\cos\theta_n = \frac{1}{2}\Big[\cos(\theta_m+\theta_n) + \cos(\theta_m-\theta_n)\Big]$$

$$= \frac{1}{2}\left[\frac{e^{j(\theta_m+\theta_n)} + e^{-j(\theta_m+\theta_n)}}{2}\right] + \frac{1}{2}\left[\frac{e^{j(\theta_m-\theta_n)} + e^{-j(\theta_m-\theta_n)}}{2}\right]$$

ここで，$j=\sqrt{-1}$ とおくと，式(5.25) は次のようになる．

$$x_n = \frac{1}{2N}\sum_{m=0}^{N-1} x_m \sum_{k=0}^{N-1} c_k^2 \left(\exp\left[j\frac{2k\pi}{2N}(m+n+1)\right] + \exp\left[-j\frac{2k\pi}{2N}(m+n+1)\right]\right.$$

$$\left. + \exp\left[j\frac{2k\pi}{2N}(m-n)\right] + \exp\left[-j\frac{2k\pi}{2N}(m-n)\right]\right) \tag{5.26}$$

$$c_k = \begin{cases} 1, & k \neq 0 \\ 1/\sqrt{2}, & k = 0 \end{cases}$$

$$= \frac{1}{2N} \sum_{m=0}^{N-1} x_m \sum_{k=0}^{N-1} c_k^2 \Big(W_N^{-(m+n+1)\frac{k}{2}} + W_N^{(m+n+1)\frac{k}{2}} \qquad (5.27)$$
$$+ W_N^{-(m-n)\frac{k}{2}} + W_N^{(m-n)\frac{k}{2}} \Big)$$

ただし，$W_N = \exp\left(\frac{-j2\pi}{N}\right)$＝ユニタリ値の N 乗根，である．

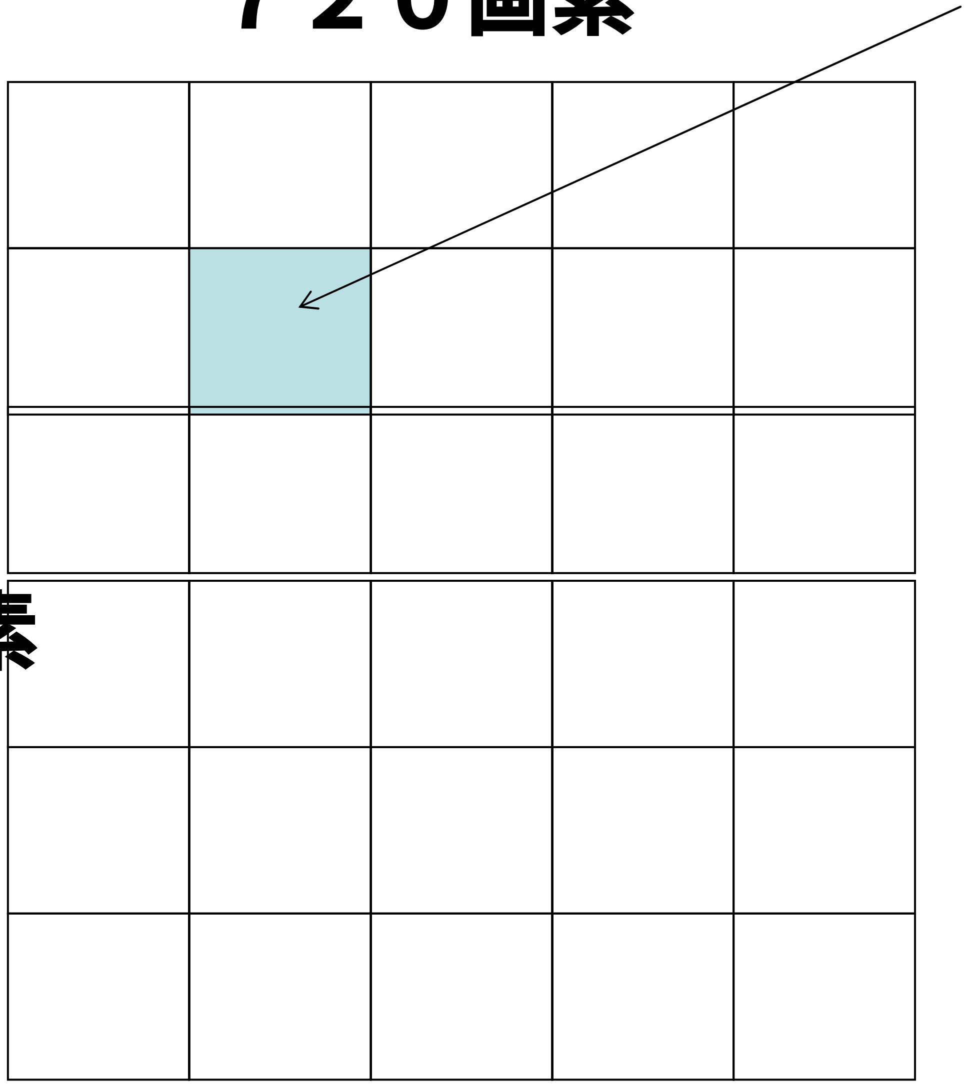
式(5.27) は以下のように変形出来る．

$$m = n \text{ and } k \neq 0 \quad \text{のとき} \left(\frac{1}{2N} x_n 2N\right) = x_n$$

$$m = n \text{ and } k = 0 \quad \text{のとき} \left(\frac{1}{2N} x_n \left(\frac{1}{\sqrt{2}}\right)^2 4N\right) = x_n$$

$$\text{ただし} \quad \sum_{k=0}^{N-1} W_N^{kl} = N\delta(l), \quad \text{ここで} \ \delta(l) = \begin{cases} 0, & l \neq 0 \\ 1, & l = 0 \end{cases}$$

７２０画素



ブロック

８ｘ８画素

１画像：

４８０画素

９０ｘ６０
ブロック

画面

正規化 2D-DCT

各ブロックに対して
DCT（N,M=8）

$$X_{u,v}^{c2} = c_u\, c_v \frac{2}{\sqrt{NM}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x_{n,m} \cos\left[\frac{(2n+1)u\pi}{2N}\right] \cos\left[\frac{(2m+1)v\pi}{2M}\right]$$

$$= \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} c_u \left[\sqrt{\frac{2}{M}} c_v \sum_{m=0}^{M-1} x_{n,m} \cos\frac{(2m+1)v\pi}{2M}\right] \cos\frac{(2n+1)u\pi}{2N}, \qquad (5.38)$$

$$u = 0, 1, \ldots, N-1, \quad v = 0, 1, \ldots, M-1, \qquad c_l = \begin{cases} 1/\sqrt{2}, & l = 0 \\ 1, & l \neq 0 \end{cases}$$

正規化 2D-IDCT

$$x_{n,m} = \frac{2}{\sqrt{NM}} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} c_u\, c_v X_{u,v}^{c2} \cos\left[\frac{(2n+1)u\pi}{2N}\right] \cos\left[\frac{(2m+1)v\pi}{2M}\right], \qquad (5.39)$$

$$n = 0, 1, \ldots, N-1, \qquad m = 0, 1, \ldots, M-1$$

演算量

各ブロック当たり：64x(63+64*4)=20480

総和

毎秒当たりのブロック数

5400*30=162000

X*cos*cos

演算量：3.2G演算／秒

cos：テーブル参照

各ブロックのDC係数Xuvに量子化

Xquv=Nearest Int（Xuv/Quv）

表8.1　輝度量子化マトリックス Q_{uv} [367]

v

u

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

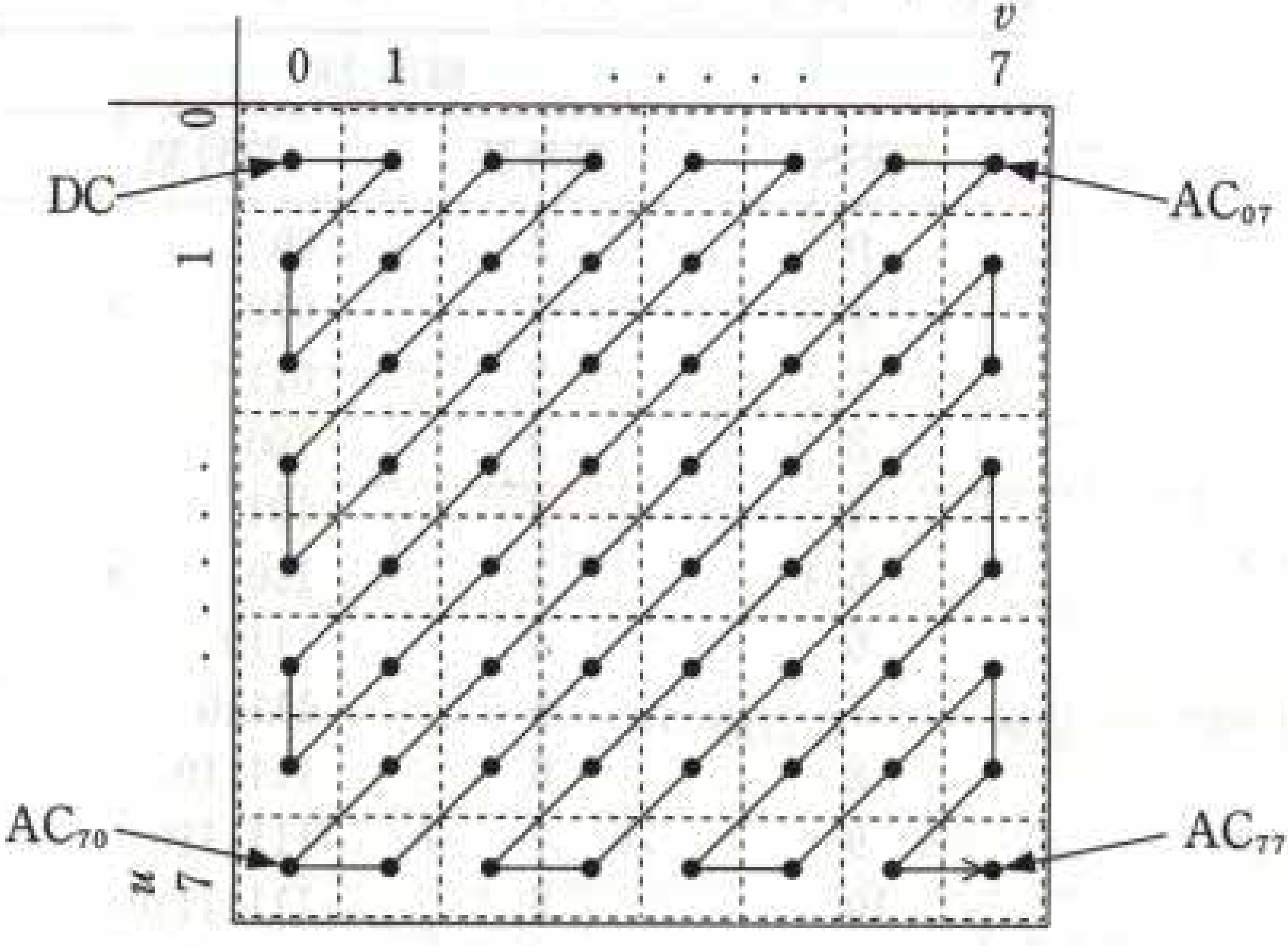
出典：© 1993 ITU-T.

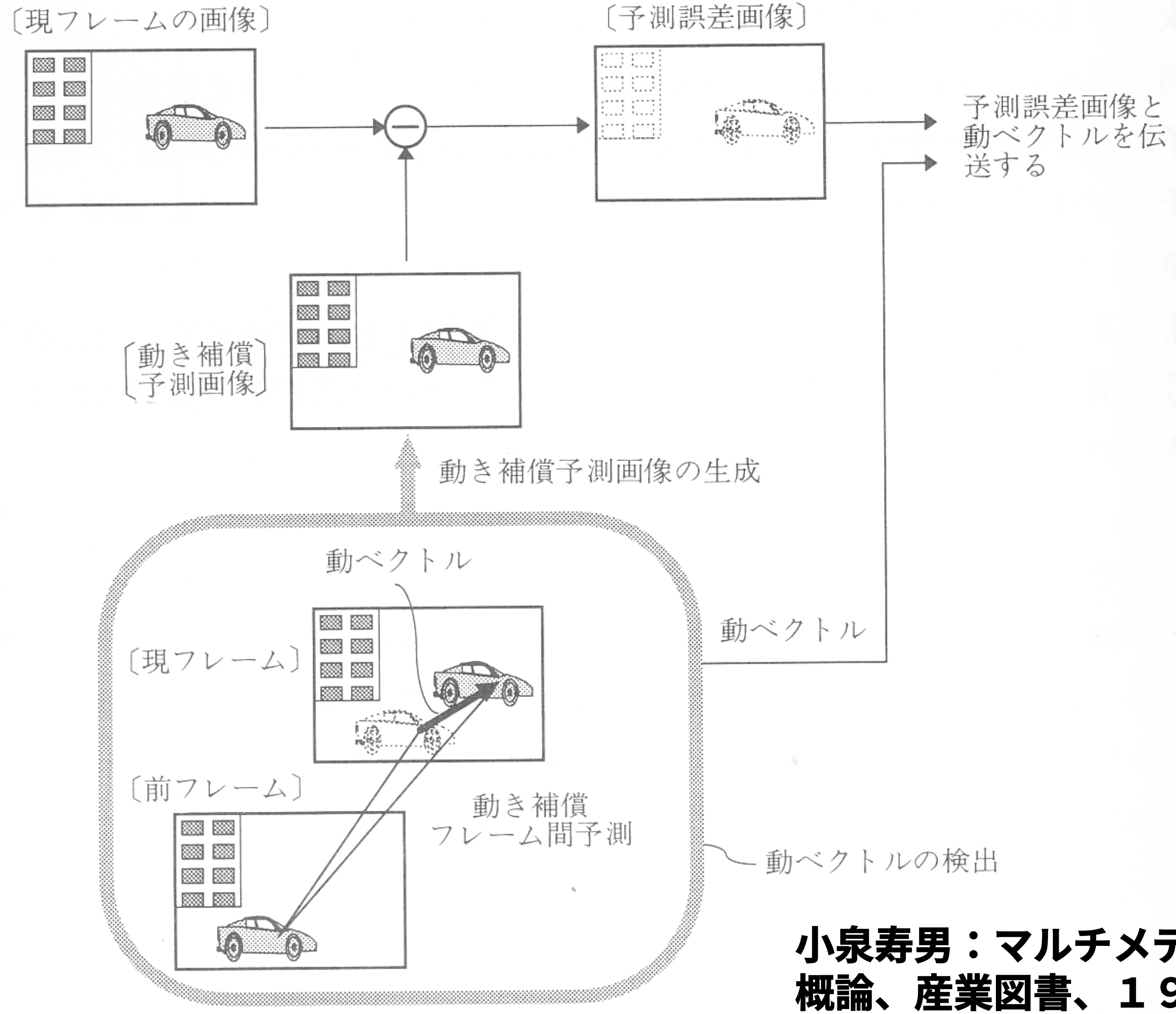
表8.2　色差量子化マトリックス Q_{uv} [367]

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

出典：© 1993 ITU-T.

各ブロックDC係数をジグザクスキャンし符号化（ハフマン符号、ランレングス符号）





小泉寿男：マルチメディア概論、産業図書、１９９７

図 4.10　動き補償フレーム間予測の原理

動き補償：フレーム間での圧縮

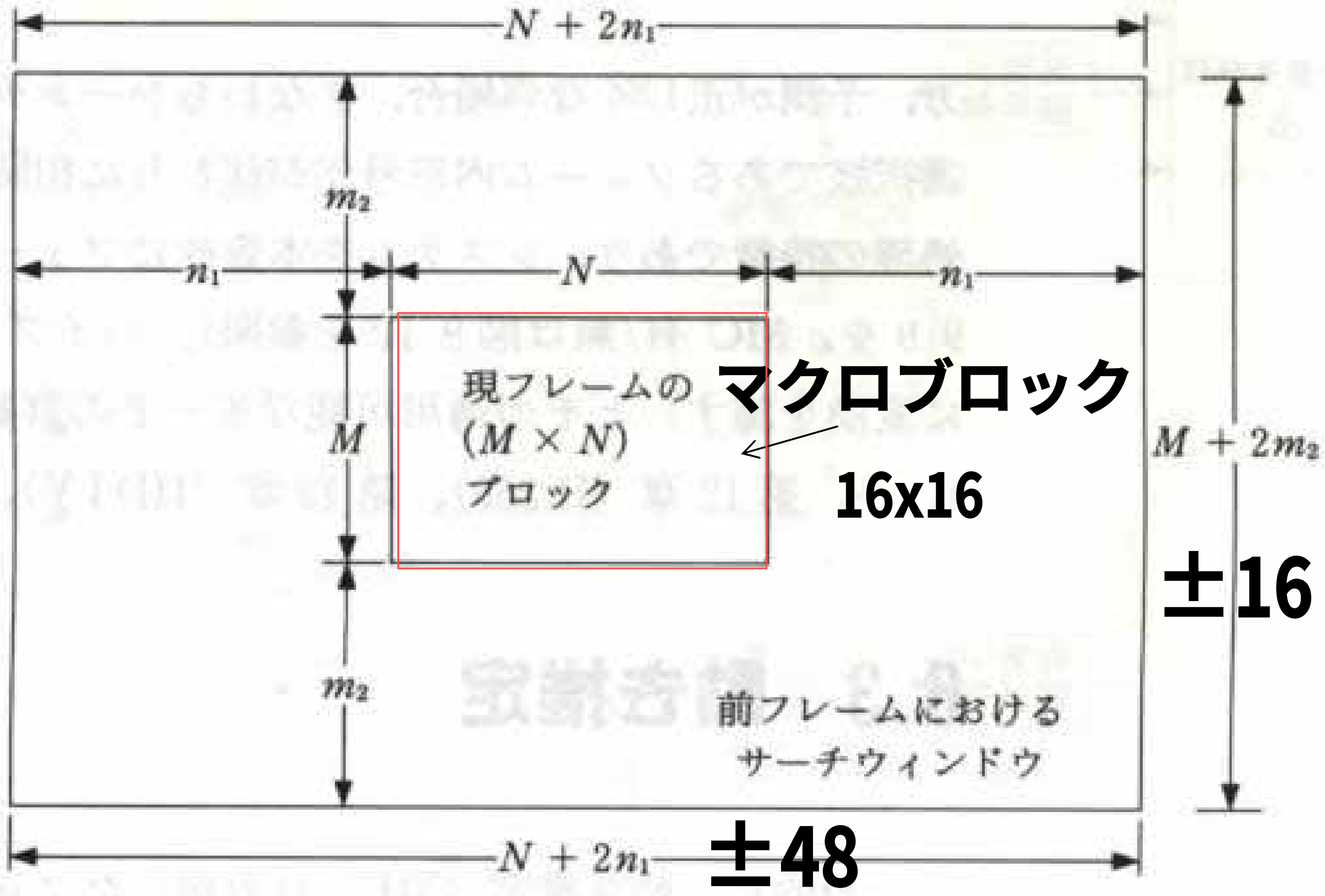


図 6.5　前フレームの $(M+2m_2)\times(N+2n_1)$ のサーチウィンドウを用いた，$(M \times N)$ ブロックの動き推定（全探索）．動きベクトル範囲はフレーム間距離あたり水平 $\pm n_1$ 画素，垂直 $\pm m_2$ ラインである．

動き補償での演算量

マクロブロック当たり：±48,16画素領域

16x16x97x33x3=2458368演算

マクロブロックサイズ　比較演算数

毎秒当たりのマクロブロック処理数：

45x30x30=40500

フレーム数/秒

演算量：1000億：100G演算

②SSE（Streaming SIMD Extension）：

１２８ビットデータに対する

単精度浮動小数点演算ｘ４

グラフィックス処理

座標変換

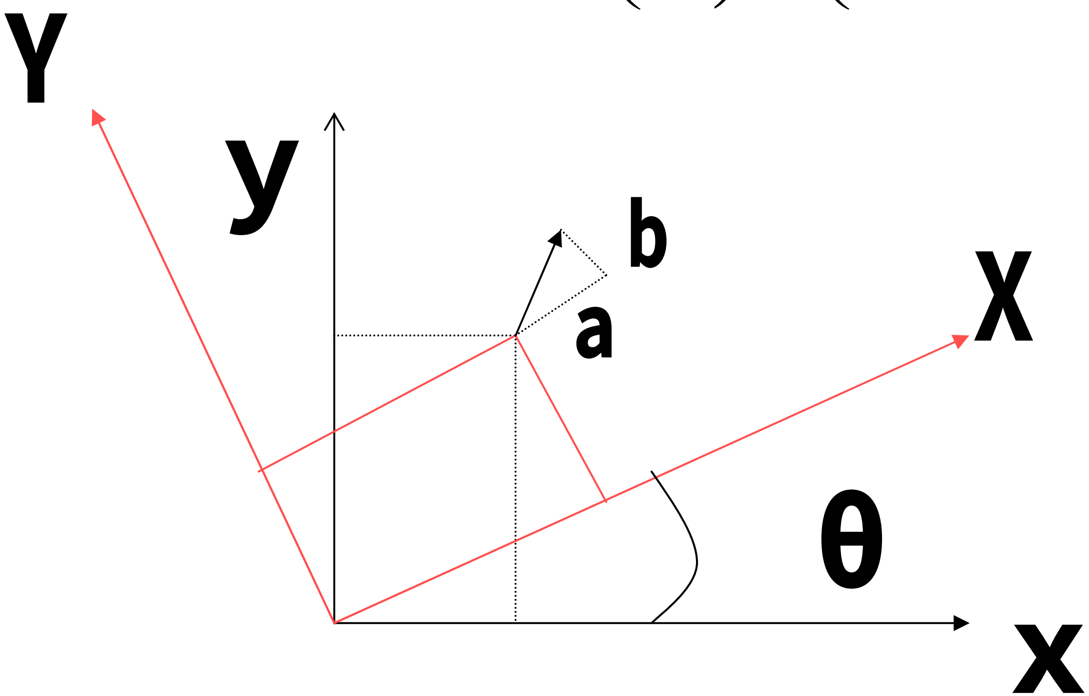
(X,Y,Z,1)=（x,y,z,1）A

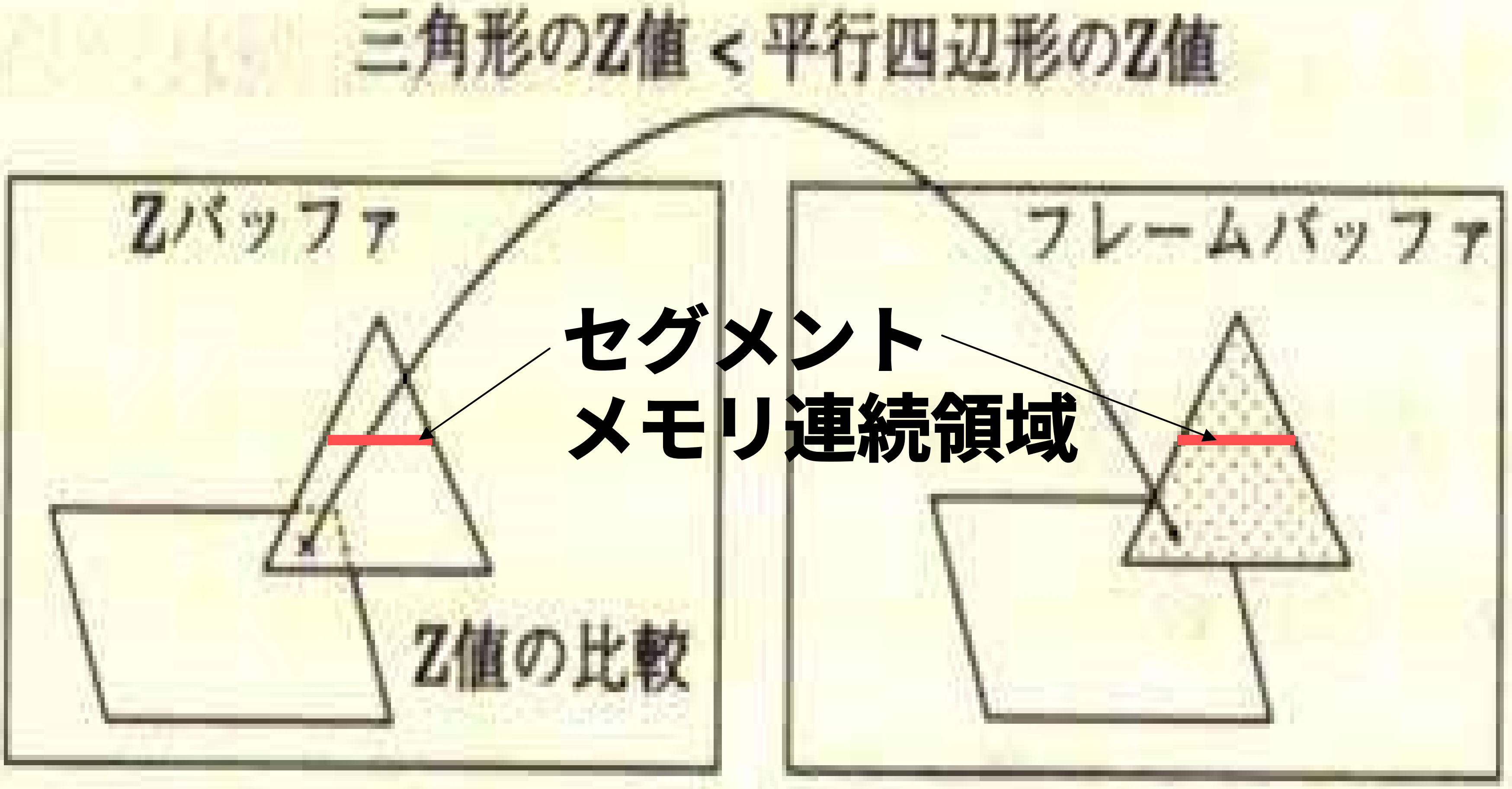
A:４ｘ４行列

Zバッファによる隠れ面処理

座標変換

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & a \\ -\sin\theta & \cos\theta & b \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$





隠れ面消去

図6.15 Zバッファ法