

Title	磁気バブル論理ならびに記憶方式に関する研究
Author(s)	松田, 潤
Citation	大阪大学, 1976, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/2029
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

磁気バブル論理ならびに 記憶方式に関する研究

1976年1月

松 田 潤

内容梗概

本論文は、著者が大阪大学大学院工学研究科(電子工学専攻)の学生として尾崎研究室において行なった研究のうち、磁気バブル論理ならびに記憶方式に関する研究をまとめたものである。

第1章緒論では、磁気バブル素子とその現状を紹介し、本研究の目的ならびに本研究の諸成果について概説している。

第2章では、磁気バブル応用の最大の目標である大容量バブルメモリのアクセス時間を記憶方式の観点から考察している。まず、従来の代表的な記憶方式を紹介し、それぞれの長短について述べている。つぎに、磁気バブル素子の特性を考慮して、従来の方式よりも低価格で短いアクセス時間を達成できるセルループ方式という新しい記憶方式とそのアクセスアルゴリズムを示している。とくに、網状セルループ方式はダイナミックメモリの思想をとり入れた方式であり、この方式に応用した最小アクセス時間を実現するダイナミックメモリの構造を2.3.2節で理論的に求めている。

第3章では、磁気バブル素子の論理機能について考察している。まず、磁気バブル同志の反発作用によって実現される論理演算を表現するために各種の二入力保存形論理セルを導入し、つぎに、これらのセルを基本論理セルとし、ほかに定数1に対応するバブル発生器などを含めて磁気バブル論理系を定義し、そのモデルによる論理関数の実現を、バブルの供給源であるバブル発生器の個数に着目して考察している。そして、或る種の論理演算は任意の論理関数を実現できる万能性を有し、しかも、論理関数の変数の数によらず、一定数のバブル発生器を用いて実現できることなど、磁気バブル論理系の性質を明らかに

している。また、磁気バブル論理系による論理関数の標準的な実現法についても考察している。

第4章では、磁気バブル素子の記憶機能以外の諸特性をも考慮した磁気バブル応用装置の構成例をいくつか提案し、磁気バブル応用の新しい方向を示している。まず、4.2節では、磁気バブル素子の二次元性を考慮して、二次元シフトレジスタを利用した二次元情報処理装置の構成とその処理方式を示している。処理方式は、二次元アレイ情報の処理を行処理と列処理の繰返しにより行なうもので、置換の分解アルゴリズムを拡張した、二次元処理を行処理と列処理とに分解するアルゴリズムについて論じている。つぎに、4.3節では、磁気バブル素子の記憶機能、論理機能、二次元性を考慮し、二次元シフトレジスタを利用した並列処理装置の構成法を提案している。この種の応用装置は今後、最も有望であり、ここではその一例としてソーティングメモリの構成を示している。最後に、4.4節では、磁気バブル素子の遠隔制御性、一様制御性を考慮し、シフト機能と論理機能とを結合したループ結合形機能メモリについて論じている。ループ結合形機能メモリは磁気バブルチップ上に計算システム等を構成する際に有効であり、ここではその一例として加算メモリの構成を示している。

第5章結論では、本研究で得られた結果と、今後に残された問題についてまとめてある。

関連発表論文

- (1) 松田, 樹下, “磁気バブルをモデルとした論理関数の実現について”, 信学会電子計算機研究会資料, EC 72-44 (1973-01).
- (2) 松田, 樹下, “磁気バブル論理系の解析”, 信学論(D), 57-D, 1, pp. 39-45 (1974-01).
- (3) 松田, 樹下, “磁気バブルを用いたスイッチングデバイス”, 信学会電子計算機研究会資料, EC 74-5 (1974-05).
- (4) 松田, 樹下, “磁気バブルを用いた図形処理デバイス”, 昭和49年度信学会全国大会, p. 270 (1974-07).
- (5) 松田, 樹下, “スイッチング機能をもつ二次元バブルメモリ”, 信学論(D), 58-D, 3, pp. 143-149 (1975-03).
- (6) 松田, 樹下, “アクセス時間最小のダイナミックメモリの構造とそのバブルメモリへの応用”, 信学会電子計算機研究会資料, EC 75-16 (1975-06).

磁気バブル論理ならびに記憶方式に関する研究

目 次

第1章 緒論	1
第2章 磁気バブルメモリの方式	5
2.1 緒言	5
2.2 記憶方式	6
2.2.1 単ループ方式	6
2.2.2 複ループ方式	7
2.2.3 メジャー/マイナーループ方式	8
2.3 セルループ方式	9
2.3.1 梯子形セルループ方式	9
2.3.2 網状セルループ方式	14
2.4 結言	21
第3章 磁気バブル論理系の解析	22
3.1 緒言	22
3.2 磁気バブル論理系のモデル	22
3.3 磁気バブル論理系による論理関数の実現	23
3.4 結言	34
第4章 磁気バブル応用装置	35
4.1 緒言	35
4.2 スイッチング機能をもつ二次元バブルメモリ	35
4.2.1 二次元変換アレイ	36
4.2.2 装置の方式と回路	37
4.2.3 二次元変換アレイの行・列分解	41
4.3 ソーティングメモリ	47
4.4 ループ結合形機能メモリ	50
4.5 結言	52
第5章 結論	53
謝辞	55
参考文献	56

第1章 緒論

1967年にベル研究所のA. H. Bobeckらにより発明された可動磁区素子—磁気バブル素子—は、(1)バブルドメインがエネルギー的に安定に存在し得ること、(2)従来の磁区利用素子にない高速、高密度が達成できること、(3)バブル—バブル相互作用によって新しい機能をもつデバイスの構成が可能であること等、十分な実用性と新しい可能性をもつ素子として多くの研究者、技術者の注目を集め、材料・制御技術・応用の各方面で精力的な研究開発が行なわれている。

では、磁気バブルとは如何なるものであるのか。つぎに、応用面から見た磁気バブルの特性を簡単に紹介する。

磁気バブルとは、図1.1に示すような縞状磁区構造をもつ或る特殊な強磁性体薄膜に垂直バイアス磁界を印加したときに生じる小さな円柱状磁区のことであり、これが媒体内を移動する様子がちょうど水中の泡のように見えるところから、その名で呼ばれている。

磁気バブルの性質のうち、情報処理分野への応用上、重要なものをつぎに要約する。[1]~[9]

(i) バブルは適当な垂直バイアス磁界のもとで安定に存在し、何らかの原因で変形しても自己成形作用により直ちに一定の形状にもどる。

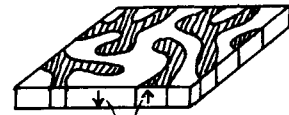
(ii) 或る強さ以上の垂直磁界の印加によりバブルの発生あるいは消滅が起こる。

(iii) 媒体面内に傾斜磁界があると、バブルはその傾斜に沿って移動する。

(iv) バブル同士は互いに反発しあい、二個のバブルは直径の4倍以内には接近して存在しえない。

(v) バブルは二つに分離することができ、分離後は自己成形作用によりどちらも同じバブルになる。

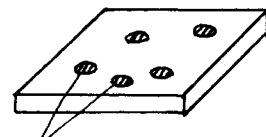
磁気バブルのこれらの性質によって、磁気バブル素子はつぎに示すような情報処理素子として興味深い機能をもつ。



磁化方向

縞状磁区構造

H: 垂直バイアス磁界



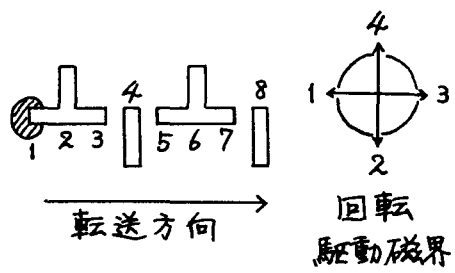
円柱状磁区(磁気バブル)

図1.1 強磁性体薄膜の磁区構造

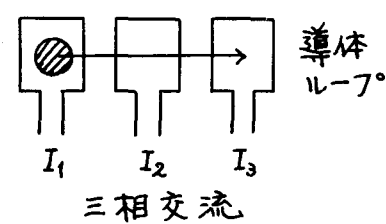
(i) 記憶機能. バブルの有無を1, 0に対応させて、媒体に情報を記憶することができる。しかも、自己成形作用により雑音に強い。

(ii) 発生・消去の任意性. バブルの発生および消去を制御することができる。これは、媒体への情報の書き込みを可能にするものである。また、後述するように磁気バブル論理回路で論理関数*を実現するときに必要な定数1を発生する回路としても重要である。

(iii) シフト機能. 媒体に記憶した情報を媒体内でシフトすることができる。シフトは、たとえば、図1.2に示すバブル転送回路により行なわれる。(a)は媒体表面に蒸着したTバーパーマロイ磁極パターンに、媒体面内の回転磁界により順番に発生する磁極によるバブルの吸引、反発を繰返してバブルを転送する方式で、磁極法と呼ばれる。(b)は媒体表面に蒸着したコイル状の導体に三相交流あるいは三相パルス電流を流し、そのとき発生する誘導磁界によりバブルを吸引、反発して転送する方式で、導体法と呼ばれる。磁極法では、チップ上のすべてのバブルを同一の回転磁界により駆動するため、一様制御、遠隔制御が可能である。一方、導体法では、局所的なバブルの駆動が可能であり、後述するゲート回路などに利用される。



(a) 磁極法



(b) 導体法

図1.2 バブル転送回路

(iv) 論理機能. バブル同志の反発作用を利用した論理演算が可能である。

(v) 複写機能. 情報を媒体内で複写することができる。

磁気バブル素子では、これらの機能を一つの固体素子に実現でき、従来のメモリデバイスには見られない特殊な動作を行なうデバイスを単一の固体素子で構成することができる。さらに、磁気バブル素子は、製造行程が簡単で低価格化が行なえる、バブル駆動に必要な電力が同程度の半導体部品に比べて $1/100 \sim 1/200$ であり、実用化に好都合な特徴をも備えている。

現在、バブル材料はバブル径が $1 \mu\text{m}$ 、バブルの移動度が 2000 cm/sec.Oe 、密度が $10^6 \sim 10^7 / \text{cm}^2$ 程度のものが開発されている。また、バブル制御技術に

* たとえば、非保存的な論理関数。

関しては、1MHzの動作速度をもつ転送、発生、消去、分割、ゲート、検出の各バブル回路が実現されつつある。^{[11]~[16]} これら基礎技術の開発と並行して応用研究も活発に行なわれており、とくにバブルメモリは実用化の段階に達している。^[21]

バブルメモリは、情報の不揮発性、高密度、超小形軽量、低消費電力、低価格など大容量メモリに適した特性を有し、しかも、アクセス機構が純電子的であるため動作速度、信頼性の点でも、機械的なアクセス機構による磁気ディスク、ドラムより優れており、将来における高速補助メモリあるいは高速大容量ファイルメモリとして期待されている。その際、バブルメモリが達成すべきアクセス時間の目標値は数msにおかれており、その開発が進められている。ところで、バブルメモリの回路構成はシフトレジスタ構成がとられる。これは、バブル転送回路が簡単に実現でき、しかも、その動作マージンが大きいことと、バブル検出回路が他のバブル回路に比べて高価であり、しかも、検出の際にバブルを或る大きさ以上に拡大しなければならぬため広い面積を占めること等の理由による。記憶密度を高めるためバブル径はますます小さくなる傾向にあり、検出回路が1ビット当りの価格に及ぼす影響は今後さらに大きくなることが予想される。このようにシフトレジスタにより構成されるバブルメモリのアクセス時間を決定する要因として、バブルの移動度とバブル転送回路の動作速度があげられる。アクセス時間の短いバブルメモリを構成するためには移動度の大きい材料と動作速度の大きいバブル制御技術の開発が必要なことは言うまでもない。しかし、現在の磁極法では1MHzの動作速度が限界であると言われている。この動作速度で数msのアクセス時間を達成するためには、数キロビット以下のシフトにより検出回路まで到達するように回路を構成しなければならない。これを磁気ディスク、ドラム方式、すなわち、単純なループシフトレジスタと検出回路により構成すると、数キロビットに対して一つの検出回路が必要となる。ところで、磁気バブル素子では転送路上に導体ループを設けてゲート動作を行なうことが可能である。そこで、転送路にバイパスを設けて、その分岐点にゲートを置き、或るビットにアクセス要求が出されたときに検出回路までバイパスを利用した近道を通っていけるようにすることができる。このときのバイパスのつけ方で決まる回路構造が記憶方式である。このように、磁気バブル素子ではゲート動作を利用してチップ上にいろいろな記憶方式を実現することができ、記憶方式を工夫することでより多くのビットに対して一つの検出回路を置くだけで目標のアクセス時間を達成することができ、検出回路の数を減らすことは1ビット当りの価格を下げることを意味している。代表的な記憶方式として、メジャー/マイナーループ方式がよく知られている。^[17]

磁気バブル応用の最大の目標は、現在までのところ、大容量メモリにおかれ

ているようであるが、記憶機能以外の磁気バブル素子の特性を生かした応用装置の提案もいくつかなされている。^{[46]~[56]} とくに、論理機能を利用した機能メモリ（情報処理機能をもつメモリ）の構成は興味深い。

磁気バブル素子で機能メモリを構成する際、ロジック・イン・メモリとして重要な役割を果たす磁気バブル論理に関する研究は応用面からも必要であり、磁気バブル論理回路の構成例が文献[15]、[31]、[36]~[43]などに数多く示されている。一方、磁気バブル素子の論理機能は数学的な立場からも論じられており、その性質がいくつか明らかにされている。磁気バブル論理の数学的な解析は、R. L. Graham^[33]により集合のモデルを用いて初めて行なわれた。これに続いて、A. D. Friedman と P. R. Menon^[34]は R. L. Graham と同様のモデルを用いて彼の理論を拡張している。また、これとは別に R. M. Sandfort と E. R. Burke^[31]は磁気バブルの保存性に着目し、バブル同志の反発によって4種類の論理演算が可能であることを示している。

本論文では、以上に述べた磁気バブル素子の現状をふまえて、研究結果をつぎの三つのテーマにわけてまとめている。

まず、第2章では、バブルメモリのアクセス時間を記憶方式の観点から考察し、ダイナミックメモリの思想を応用したセルループ方式という新しい記憶方式とそのアクセスアルゴリズムについて論じている。この方式で数msのアクセス時間を達成する場合、およそ、 10^5 ビットに対して検出回路を一つ設ければよい。チップの記憶容量（ $10^5 \sim 10^6$ ビット）を考えると、この数字はチップに設置する検出回路数が数個でよいことを意味しており、検出回路が1ビット当りの価格に及ぼす影響は非常に小さいと言える。

つぎに、第3章では、磁気バブル論理系の解析を行なっている。R. M. Sandfort と E. R. Burke による4種類の二入力保存形論理セルを基本論理セルとし、それによる論理関数の実現を、バブル供給源であるバブル発生器^[32]の個数に着目して考察し、完全性などの磁気バブル論理系の性質を明らかにしている。また、磁気バブル論理系による論理関数の標準的な実現法についても考察している。

最後に、第4章では、記憶装置以外の磁気バブル応用装置について論じ、磁気バブル応用の新しい方向を示している。磁気バブル素子の論理機能、2次元性、遠隔制御性を考慮した装置の構成とその処理方式をいくつか提案している。とくに、2次元シフトレジスタを利用した並列処理装置は今後、最も有望な磁気バブル応用装置であると思われる。

第2章 磁気バブルメモリの方式

2.1 緒言

一般に、メモリデバイスはマトリクス方式とセンサ方式の二つに分類されるが、バブルデバイスはそのいずれにも分類することができない。これは、メモリ機能とゲート動作の両者を一体の固体デバイスに実現できること、および、外部磁界によりバブルを駆動できることが特殊なメモリ構成を可能にしていることによる。

マトリクス方式は、多数のメモリ素子の配列と、番地選択、読出し、書込み制御を行なう周辺回路とから構成される。配列を構成するすべてのメモリ素子に対する布線と、番地選択回路の数に対応した回路接続が周辺回路系との間に必要であるため、比較的容量の小さいメモリに適している。コアメモリ、ワイヤメモリ、ICメモリなどがこれに属する。

センサ方式は、記憶媒体と情報の読出し、書込みを行なうセンサ、および、センサ・媒体間の相互位置制御を行なう機構とから構成される。情報信号回路はセンサに接続されるだけでよく、媒体への布線は不要である。また、媒体のみを増加することにより記憶容量を容易に増すことができ、媒体の着脱も可能である。ビット当たりの価格が安く大容量メモリに適しているが、センサ・媒体間の相互位置制御手段が機械的運動によるものが多く、動作速度と信頼性に問題がある。ディスク、ドラムなどがこれに属する。

これに対して、バブルデバイスでは媒体自身は静止したままで媒体内での磁区の移動による情報の移動のみが行なわれるという点で、機械的手段による磁気ディスク、ドラムなどとは本質的に異なっている。バブルを媒体面に設けられた転送路上を移動させるのに転送路に電氣的回路接続は不要であり、電気回路接続が必要なのはセンサや、ゲート動作を行なう特定の箇所だけである。しかも、これらの回路は転送路と同様に媒体面に蒸着される。このように、バブルメモリは純電子的アクセスによるため、ディスク、ドラムなどに比べて動作速度、信頼性の点で優っており、将来における高速補助メモリ、あるいは、高速大容量ファイルメモリとして期待されている。

バブルデバイスは素子の特性から、シフトレジスタを主体として構成するのが最も適している。シフトレジスタ構成によるバブルメモリのアクセス時間を短くするためには、(1)移動度の大きいバブル材料、(2)高速で、かつ、信頼性の高いバブル転送技術、(3)アクセスの際のシフト回数が少ない記憶方式、などの研究開発が必要である。この章では、記憶方式の観点からバブルメモリ

のアクセス時間について考察し、従来の方式よりも短いアクセス時間を実現する、セルループ方式という記憶方式を提案する。

2.2 記憶方式

この節では、これまでに知られているいくつかの記憶方式について概観し、そのアクセス時間および回路構成の特徴を述べる。

2.2.1 単ループ方式

単ループ方式は図2.1に示すように、1個の巡回シフトレジスタ(ループ)と、ループ上に設置された読出し・書込み回路(R/W回路)とから構成される。ループの長さはLであり、そこに長さrのレコードをr個連続して記憶している。アクセスは、レコードをR/W回路までシフトして行なう。この方式はチップ上のすべてのビットが同一の回路を共用しているため、構造が簡単で、かつ、ビット当たりの価格が安い。

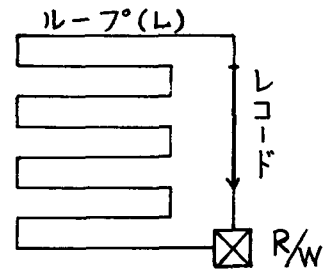


図2.1 単ループ方式

単ループ方式のアクセス時間は、レコードの先頭ビットをR/W回路までシフトするのに要する時間と、レコードの先頭ビットから最終ビットまでをビット直列にアクセスするのに要する時間の和で表わされ、

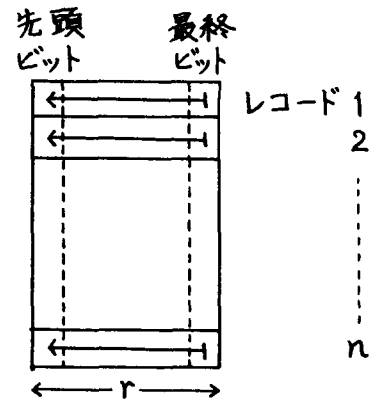


図2.2 メモリ空間

$$\begin{cases} t_E = \left(\frac{L}{2} + r\right) T_R \\ t_M = (L + r) T_R \end{cases} \quad \text{----- (2.1)}$$

となる。ただし、 t_E はレコードへのアクセス要求が生じた時点における先頭ビットのループ上での位置に関して平均したアクセス時間を表わし、 t_M は最悪の場合、すなわち、アクセス要求が生じたとき先頭ビットがR/W回路の直後に位置している場合のアクセス時間を表わす。また、 T_R は回転駆動磁界の周期であり、1ビットだけシフトするのに要する時間を表わしている。

ループの長さLは普通、 $n \cdot r$ に等しい。このとき、式(2.1)は、

$$\begin{cases} t_E = \frac{(n+2)r}{2} \cdot T_R \\ t_M = (n+1)r T_R \end{cases} \quad \text{----- (2.2)}$$

となり、単ループ方式はオーダ n のアクセス時間を要することになる。このアクセス時間は、後述する記憶方式に比べてかなり長い。

2.2.2 複ループ方式

複ループ方式は、レコードを複数個のループに分割して記憶し、複数個のR/W回路を用いて並列にアクセスすることにより、短いアクセス時間を実現したものである。図2.3に示すように、 m 個の巡回シフトレジスタ（ループ）から成り、各ループにはR/W回路が設けられている。各レコードは図2.4に示すように、 m 個の成分に分割され、第 i 成分を i 番目のループに記憶している。なお、 m には r の約数を選んでおり、各成分の長さは r/m である。

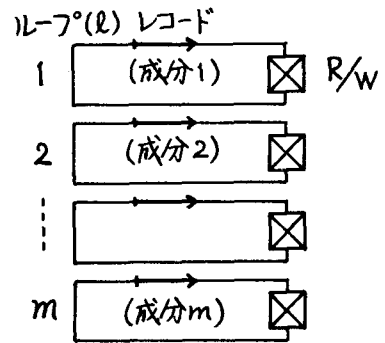


図2.3 複ループ方式

アクセスは m 個のR/W回路により直並列に行なう。すなわち、レコードの各成分の先頭ビットを各ループに設けられたR/W回路までシフトし、長さ r/m の各成分をビット直列にアクセスする。したがって、複ループ方式のアクセス時間は、

$$\begin{cases} t_E = \left(\frac{l}{2} + \frac{r}{m} \right) T_R \\ t_M = \left(l + \frac{r}{m} \right) T_R \end{cases} \quad \text{----- (2.3)}$$

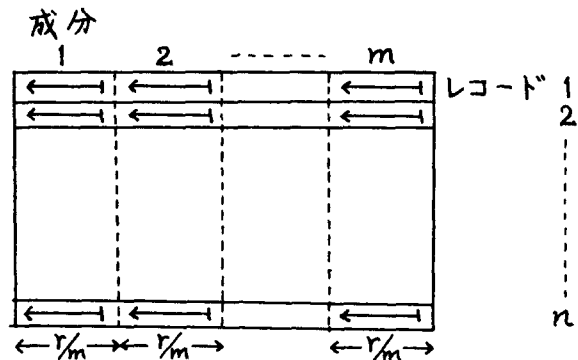


図2.4 メモリ空間

となる。ただし、 l は各ループの長さであり、普通は nr/m に等しい。このとき、式(2.3)は、

$$\begin{cases} t_E = \frac{(n+2)r}{2m} \cdot T_R \\ t_M = \left(n + \frac{r}{m} \right) T_R \end{cases} \quad \text{----- (2.4)}$$

$$t_M = \frac{(n+1)r}{m} \cdot T_R$$

となり、単ループ方式に比べて $1/m$ に改善されている。なお、 $1 < m \leq r$ であり、オーダー n のアクセス時間が可能である。

一方、回路構成から見ると、シフトレジスタ回路よりも高価な R/W 回路を m 個必要とするため、ビット当たりの価格は高い。

2.2.3 メジャー/マイナーループ方式

メジャー/マイナーループ方式 (M/m ループ方式) は、複ループ方式における m 個の R/W 回路をループで連結し、これを単一の R/W 回路で置きかえて共用できるようにしたもので、ビット当たりの価格が安い。

図 2.5 に示すように、メジャーループと呼ばれる巡回シフトレジスタと、マイナーループと呼ばれる m 個の巡回シフトレジスタとから構成される。長さ l_m のメジャーループには R/W 回路が設けられている。

マイナーループは長さ l_m で、トランスファゲートを通してメジャーループに結合されている。レコードはマイナーループに記憶されており、記憶形式は複ループ方式の場合と同様である。

アクセスはつぎの手順により行なわれる。

まず、レコードの各成分の先頭ビットを、マイナーループ上をトランスファゲートまでシフトする。これに要する時間は、

$$\begin{cases} t_{E1} = \frac{l_m}{2} \cdot T_R \\ t_{M1} = l_m T_R \end{cases} \quad \text{----- (2.5)}$$

である。つぎに、レコードの各成分をトランスファゲートを通してマイナーループからメジャーループに移し、メジャーループ上にレコードを組立てる。組立てたレコードはメジャーループ上を R/W 回路までシフトし、そこでアクセスを行なう。最後に、レコードの各成分を再びトランスファゲートを通してメジャーループからマイナーループ上のもとの位置に戻しておく。これに要する時間は、

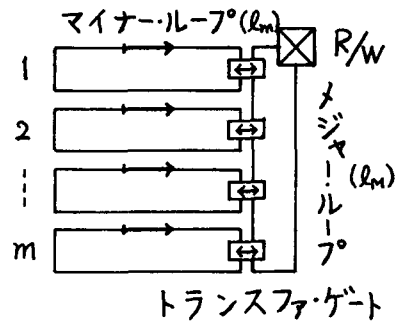


図 2.5 M/m ループ方式

$$t_2 = \left\{ l.c.m.(l_M, l_m) + \frac{r}{m} \right\} T_R \quad \text{----- (2.6)}$$

である。ただし、 $l.c.m.(l_M, l_m)$ は l_M と l_m の最小公倍数である。

M/mループ方式のアクセス時間は、式(2.5)と式(2.6)との和で表わされ、

$$t_E = \left\{ \frac{l_m}{2} + l.c.m.(l_M, l_m) + \frac{r}{m} \right\} T_R \quad \text{----- (2.7)}$$

$$t_M = \left\{ l_m + l.c.m.(l_M, l_m) + \frac{r}{m} \right\} T_R$$

となる。

l_M, l_m は普通、 $r \leq l_M < 2r$ 、 $l_m = nr/m$ であり、チップの容量が大きい場合には一般に $n \gg r$ であるため、 $l_m \gg l_M$ となる。 $l.c.m.(l_M, l_m)$ の値を小さくするために $l_M = l_m/k$ (l_m の約数) とすれば、式(2.7)は、

$$\begin{cases} t_E = \left(\frac{l_m}{2} + l_m + \frac{r}{m} \right) T_R \\ t_M = \left(l_m + l_m + \frac{r}{m} \right) T_R \end{cases} \quad \text{----- (2.8)}$$

と書きかえられる。式(2.8)と式(2.3)とを比べると、M/mループ方式においてR/W回路を共用したことによるアクセス時間の増加(式(2.8)の第2項)は $l_m T_R = (nr/m) T_R$ であることがわかる。この増加はチップの容量が大きくなったとき問題となる。

2.3 セルループ方式

この節では、セルループ方式という新しい記憶方式を提案する。セルループ方式はR/W回路数を増さずに、前述のM/mループ方式におけるR/W回路の共用に起因するアクセス時間の増加を少なくすることを目的としている。さらに、メモリリクエストのアドレス発生状況によりアクセス時間の実質的な短縮が可能である。また、セルループ間の結合度を強めることによりアクセス時間を大幅に短縮することも可能である。

2.3.1 梯子形セルループ方式

梯子形セルループ方式は最も簡単な構造をもつセルループ方式で、図2.6に示すように、セルループ(あるいは、単にセル)と呼ばれる長 m 個の巡回シ

フトレジスタと、メジャーロープと呼ばれる巡回シフトレジスタとから構成される。セルロープは長さ l_c で、 l 個ずつがスイッチゲートを通して梯子状に連結されている。長さ l_m のメジャーロープには R/W 回路が設けられている。また、窓セルと呼ばれる特定のセルロープがトランスファゲートを通してメジャーロープと直接に結合されている。

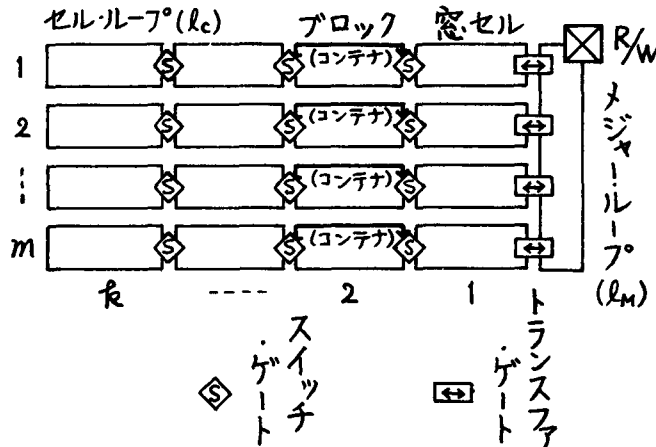


図 2.6 梯子形セルロープ方式

すべてのスイッチゲートは1本の導体に流す電流により一様に制御しており、スイッチゲートのオン・オフにより二通りのバブル転送路の切換えを行なっている。スイッチゲートがオフのときには、バブルはそれぞれのセルロープ上を転送され、情報はセル内に保持される。これをセルロープモード (Cモード) という。一方、スイッチゲートがオンになると、バブルは l 個のセルロープを1本につなぎあわせてできる転送路上を転送される。この転送路は長さ $l_m = l \cdot l_c$ で、M/mループ方式の場合に対応させて、マイナーロープと呼ぶ。そして、この転送モードをマイナーロープモード (mモード) という。

情報の記憶形式は図 2.7 に示すように、全レコードを $n/2l$ 個のレコードから成る $2l$ 個のブロックに分割し、さらに、各レコードを図 2.4 と同様に m 個の成分に分割したときにできる長さ $n/2l \cdot m$ の情報系列を各セルに2個ずつ記憶している。この単位系列をコンテナという。なお、mモードのもとでは、記憶形式はM/mループ方式の場合と同じになる。すなわち、 i 番目のマイナーロープ上にある各コンテナは、或るブロックに含まれるレコードの第 i 成分を連結したものである。

アクセスはつぎの手順により行なわれる。

まず、レコードが入っているコンテナをCモードのもとでセルロープの上半部

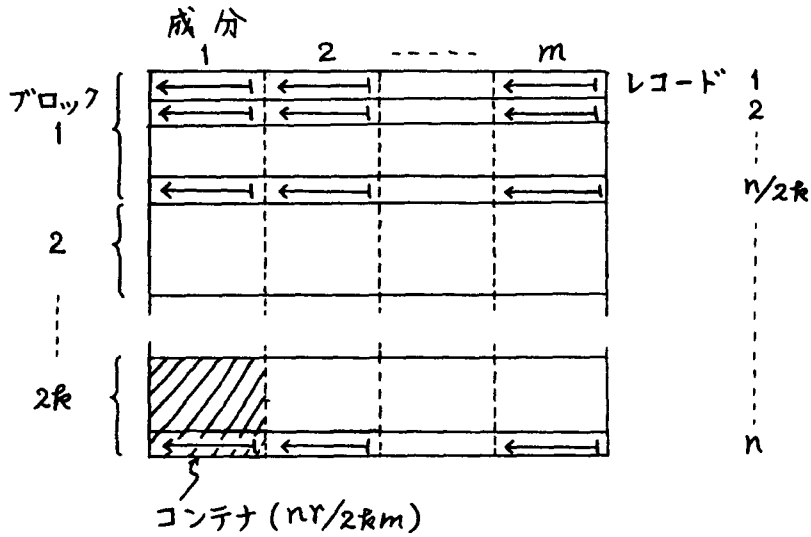


図2.7 メモリ空間

にシフトする。これに要する時間は、

$$\begin{cases} t_{E1} = \frac{l_c}{2} T_R \\ t_{M1} = l_c T_R \end{cases} \quad \text{----- (2.9)}$$

である。つぎに、このコンテナを m モードのもとで窓セルの上半部、すなわち、トランスファゲートの直前までシフトする。これに要する時間は、

$$\begin{cases} t_{E2} = \frac{(k-1)l_c}{4} \cdot T_R = \frac{l_m - l_c}{4} \cdot T_R \\ t_{M2} = \frac{(k-1)l_c}{2} \cdot T_R = \frac{l_m - l_c}{2} \cdot T_R \end{cases} \quad \text{----- (2.10)}$$

である。レコードが入っているコンテナを窓セルまでシフトした後は、メジャー・ループと窓セルとの間で、 M/m ループ方式においてメジャー・ループとマイナー・ループとの間で行なったのと同様の操作を施すことによりレコードへのアクセスが完了する。この最後の操作に要する時間は、

$$\begin{cases} t_{E3} = \left\{ \frac{l_c}{4} + l.c.m.(l_m, l_c) + \frac{r}{m} \right\} T_R \\ t_{M3} = \left\{ \frac{l_c}{2} + l.c.m.(l_m, l_c) + \frac{r}{m} \right\} T_R \end{cases} \quad \text{----- (2.11)}$$

である。

梯子形セルループ方式のアクセス時間は、三つの式((2.9)、(2.10)、および(2.11))の和で表わされ、

$$\begin{cases} t_E = \left\{ \frac{l_m + 2l_c}{4} + l.c.m.(l_M, l_c) + \frac{r}{m} \right\} T_R \\ t_M = \left\{ \frac{l_m + 2l_c}{2} + l.c.m.(l_M, l_c) + \frac{r}{m} \right\} T_R \end{cases} \quad \text{----- (2.12)}$$

となる。 $l.c.m.(l_M, l_c)$ の値を小さくするために、 l_c を l_M に等しくすれば、式(2.12)は、

$$\begin{cases} t_E = \left(\frac{l_m + 2l_M}{4} + l_M + \frac{r}{m} \right) T_R \\ t_M = \left(\frac{l_m + 2l_M}{2} + l_M + \frac{r}{m} \right) T_R \end{cases} \quad \text{----- (2.13)}$$

と書きかえられる。式(2.13)と式(2.8)(M/mループ方式のアクセス時間)とを比べると、R/W回路の共用に起因するアクセス時間の増加(第2項)は $l_M/l_m (= 1/r)$ に改善されていることがわかる。マイナーループの数 m を $r/2$ にしたときには、 l_M/l_m の値は約 r/n になる。チップの容量が大きいときには、一般に $n \gg r$ であり、容量が大きくなるほど、より改善されることがわかる。また、レコードの各成分の先頭ビットをトランスファゲートまでシフトするのに要する時間(第1項)も $1/2 + l_M/l_m (\approx 1/2)$ に改善されている。これは、レコードを含むコンテナを窓セルまでシフトするときに、スイッチゲートを横切る近道を利用していることによる。結局、梯子形セルループ方式はM/mループ方式よりも、

$$\Delta t_E = \left(\frac{5}{4} l_m - \frac{3}{2} l_M \right) T_R \quad \text{----- (2.14)}$$

$$\Delta t_M = \left(\frac{3}{2} l_m - 2 l_M \right) T_R$$

だけ短いアクセス時間を実現している。 $m = r/2$ 、 $l_M = 2r$ の場合には、 Δt_E 、 Δt_M の値はそれぞれ、

$$\Delta t_E = \left(\frac{5}{2} n - 3r \right) T_R \quad \text{----- (2.15)}$$

$$\Delta t_M = (3n - 4r) T_R$$

である。

つぎに、アクセスアルゴリズムの前半部(レコードを含むコンテナを窓セルまでシフトする部分)を式で示す。

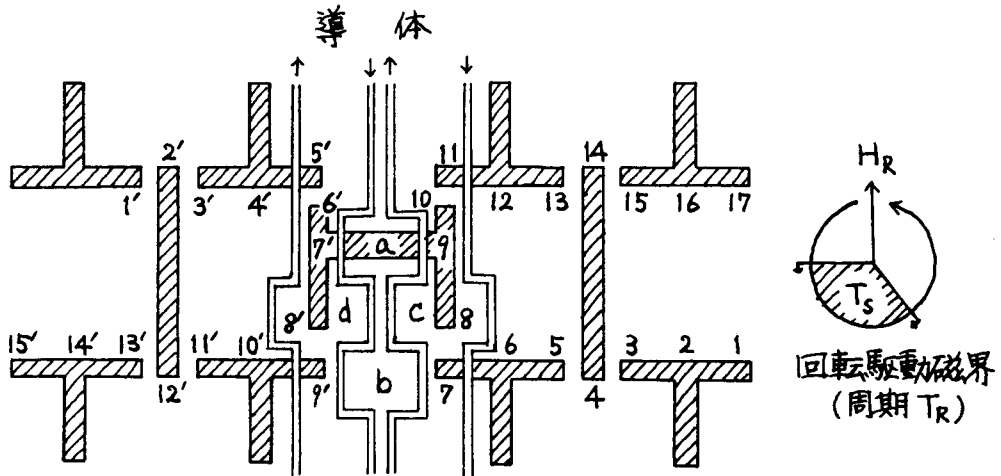
```

procedure TOWINDOW(CC, BN, TC, SC, ODR)
begin
  while CC ≠ BN do
    begin
      shift ;
      if TC = 0 (mod  $l_c/2$ ) then
        if SC = 0 then
          begin
            ODR ←  $\overline{\text{ODR}}$  ;
            if ODR = 0 then CC ← CC + 1 (mod 2k)
            else CC ← CC - 1 (mod 2k) ;
            D ← BN - CC (mod 2k) ;
            if (D < k and ODR = 0) or (D > k and ODR = 1)
            then SC ← 1
          end
        else
          if ODR = 0 then CC ← CC + 1 (mod 2k)
          else CC ← CC - 1 (mod 2k)
        end ;
      SC ← 0
    end
  end
end

```

TCはタイミングカウンタで、駆動磁界の回転数を $\text{mod } l_c/2$ で計数しており、ゲート（スイッチゲート、およびトランスファゲート）直前のビットのコンテナ内における位置を示している。CCはコンテナカウンタで、トランスファゲート直前のビットが所属するコンテナの番号を示している。ODRはマイナーループ上のコンテナの並び方を示すフリップフロップで、ODR=0は順方向に並んでいることを示し、ODR=1は逆方向に並んでいることを示す。SCはスイッチゲートのオン・オフを制御して転送モードの切換えを行なうフリップフロップで、SC=0でスイッチゲートはオフとなり、SC=1でオンとなる。すなわち、SC=0はCモードであることを示し、SC=1はMモードであることを示す。また、BNにはアクセスしたいレコードが所属するブロックの番号が入っている。アルゴリズム中、shiftはバブルを1ビットだけシフトし、TCを1だけ増す操作を表わしている。

最後に、スイッチゲートのバブル回路例を図2.8に示す。バブル転送路の切換えは導体に流す電流の有無により行なわれる。導体に電流を流さないときには、セルループ n 、 $n+1$ 上のバブルはTバーパーマロイパターン¹⁾の転送路に沿って、それぞれ $1 \rightarrow 2 \rightarrow \dots \rightarrow 17$ 、 $1' \rightarrow 2' \rightarrow \dots \rightarrow 15'$ と転送され、バブルはそれぞれのセルループから外へは出ない。これがCモードであり、コンテナは各セルに保持される。一方、導体にパルス幅 T_s 、周期 T_R のパルス電流を回転駆動磁界と同期させて流すと、電流が流れている期間、ループ電流によって a 、 b に吸引磁界が、 c 、 d に反発磁界が誘導される。その結果、転送路の切換えが起こり、バブルはスイッチゲートを横切って、 $1 \rightarrow 2 \rightarrow \dots \rightarrow 7 \rightarrow b \rightarrow 9'$



セル・ループ $i+1$ * スイッチ・ゲート * セル・ループ i

図2.8 スイッチゲートのバブル回路例

→15'、1'→2'→…→7'→a→9→…→17 と転送され、マイナーループ上を
 通って他のセルに移ることになる。これがmモードで、コンテナはマイナーループ上を窓セルまで移動することが可能である。

2.3.2 網状セルループ方式

梯子形セルループ方式はセルループをスイッチゲートにより一列に連結しており、コンテナを窓セルまでシフトするのに、連結されたセルループの長さ
 に比例した時間を要する。つぎに述べる網状セルループ方式はダイナミックメモリの構造を
 応用してセルループを結合することにより、 $\log_2 L$ に比例した時間でコンテナを窓セルまでシフト
 ことができ、アクセス時間はさらに短縮される。

まず最初に、アクセス時間最小のダイナミックメモリの構造を求める。
 一般に、データを記憶する長個のセルから成り、窓セルと呼ばれる特定のセルにのみ外部からの
 アクセスが可能で、メモリ変換と呼ばれる操作により長個のセルに記憶されているデータの配置を
 メモリ内部で変えることができるメモリをダイナミックメモリという。ダイナミックメモリにお
 いて窓セル以外のセルに記憶されているデータにアクセスしたいときには、メモリ変換によりその
 データを窓セルに移してアクセスすることになる。ダイナミックメモリの例とし

て磁気ディスク、ドラムなどをあげることができる。この場合には、読出し・書込みヘッドの下が窓セルで、媒体の回転による巡回シフトがメモリ変換である。このように、ただ1種類のメモリ変換ではアクセス時間は一般にオーダーである。ところが、適当な2種類のメモリ変換を用意すれば、オーダー $\log_2 k$ のアクセス時間を実現することができる。一般に、適当な r 種類のメモリ変換を用意してオーダー $\log_r k$ のアクセス時間を実現できることが知られている。

H. S. Stone [29] は容量が 2^S で、アクセス時間がたかだか S であるシャフルメモリと呼ばれるダイナミックメモリを提案している。また、A. V. Aho と J. D. Ullman [30] は容量が $k (= 2^S - 1)$ で、 b 個の連続したデータをたかだか $2.52 \log_2 k + b - 2$ の時間で連続アクセスできるダイナミックメモリを提案している。

ところで、2種類のメモリ変換をもつダイナミックメモリのアクセス時間に関して、

$$\begin{cases} t_E \geq \lfloor \log_2 k \rfloor - \frac{1}{k} (2^{\lfloor \log_2 k \rfloor + 1} - \lfloor \log_2 k \rfloor - 2) \\ t_M \geq \lceil \log_2 (k+1) \rceil - 1 \end{cases} \quad \text{----- (2.16)}$$

が成り立つ。ここでは、式(2.16)の右辺に示したアクセス時間の下限を実現する、 $k (= 2^S - 1)$ 個のセルから成るダイナミックメモリの構造を求める。容量が k のダイナミックメモリのセル集合を $C = \{c_1, c_2, \dots, c_k\}$ とするとき、メモリ変換は集合 C から C 自身への全単射 τ として定義される。セルを節点とし、 $\tau(c_i) = c_j$ の関係を節点 c_i から節点 c_j に向かう有向枝とする有向グラフ $G_\tau = [C, D, \psi_\tau]$ を考える。ただし、 $D = \{d_1, d_2, \dots, d_k\}$ は有向枝の集合で、 ψ_τ は、

$$\psi_\tau(d_i) = (c_i, c_j) \quad \Leftrightarrow \quad \tau(c_i) = c_j$$

を満足する。集合 D から直積 $C \times C$ への写像である。 τ が全単射であることから、グラフ G_τ の各節点の正負の線度 [61] はすべて1となる。

いま、2種類のメモリ変換を τ_1, τ_2 とし、 τ_1, τ_2 を表現する有向グラフをそれぞれ、 $G_1 = [C, D_1, \psi_1]$ 、 $G_2 = [C, D_2, \psi_2]$ とする。ただし、 $D_1 \cap D_2 = \emptyset$ であるとする。

二つのグラフ G_1, G_2 を合成して得られる有向グラフ、

$$G = G_1 \cup G_2 = [C, D_1 \cup D_2, \psi]$$

* $\lfloor x \rfloor$ は x を超えない最大の整数を表わし、 $\lceil x \rceil$ は x よりも小さくない最小の整数を表わす。

$$\psi(d) = \begin{cases} \psi_1(d) & ; d \in D_1 \text{ のとき} \\ \psi_2(d) & ; d \in D_2 \text{ のとき} \end{cases}$$

を考える。メモリの構造はこのグラフ G と 1 対 1 に対応している。グラフ G_1, G_2 が前述の性質をもつことから、グラフ G の各節点の正負の線度はすべて 2 となる。

したがって、アクセス時間（すなわち、そのセルから窓セルまでの距離）が 0 のセルは 1 個（窓セル自身）であり、アクセス時間が 1 のセルはたかだか 2 個、一般にアクセス時間が n ($0 \leq n \leq s-1$) のセルはたかだか 2^n 個である。各値のアクセス時間をもつセルがここで述べた上限個数だけ存在するとき、かつ、そのときに限り、メモリのアクセス時間は最小となる。すなわち、グラフ G が図 2.9 に示す完全 2 進木を部分グラフとして含むとき、かつ、そのときに限り、最小アクセス時間が実現される。ただし、図 2.9 ではセル C_1 を窓セルにしている。また、実線で示した枝はメモリ変換 σ に対応し、破線で示した枝はメモリ変換 τ に対応している。なお、セルには番地をつけて表わすことにするので、 C_1 以外のセルを図 2.9 のように配置しておいても一般性を失わない。

この 2 進木に、各節点から深さ $s-1$ の節点に向かう枝をつけ加えてグラフ G を完成し、アクセス時間最小のダイナミックメモリの構造を決めることがで

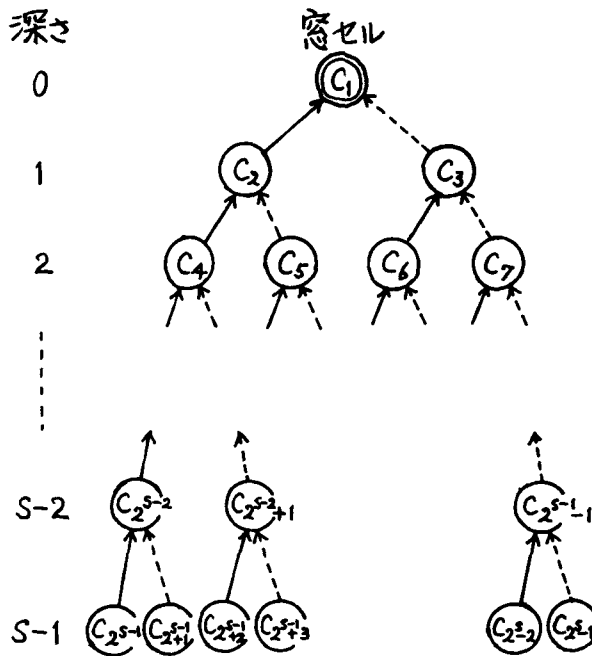


図 2.9 C_1 を根とする深さ $s-1$ の完全 2 進木

きるが、その際、つぎの条件を満足するアクセスアルゴリズムが存在するようなメモリ変換であることが重要である。

- (i) アクセスしたいデータを記憶しているセルを簡単に求めることができる。
- (ii) アクセスしたいデータを記憶しているセルがわかったときに、そのデータを窓セルまで持っていく最短経路を簡単に求めることができる。

セル、およびデータにはそれぞれ1から l までの番地、および識別番号をつけて表わす必要があるが、ここでは、セルの番地としてセル C_i に番地 i をつけることにする。このように決めると、メモリ変換 τ_1, τ_2 は図2.9の2進木に関して、

$$\begin{cases} \tau_1(i) = \frac{i}{2} & ; i = 2, 4, \dots, 2^s - 2 \\ \tau_2(i) = \frac{i-1}{2} & ; i = 1, 3, \dots, 2^s - 1 \end{cases} \quad \text{----- (2.17)}$$

を満足する。この関係をすべての i について、

$$\begin{cases} \tau_1(i) = 2^{-1}i \pmod{l} \\ \tau_2(i) = 2^{-1}(i-1) \pmod{l} \end{cases} ; 1 \leq i \leq l \quad \text{----- (2.18)}$$

と拡張する。ただし、

$$x \pmod{l} = \begin{cases} x \pmod{l} & ; x \pmod{l} \neq 0 \text{ のとき} \\ l & ; x \pmod{l} = 0 \text{ のとき} \end{cases}$$

であり、 2^{-1} は $2^{-1} \cdot 2 = 1 \pmod{l}$ を満足する2の逆数で、 $2^{-1} \pmod{l} = 2^{s-1}$ である。式(2.17)から式(2.18)に拡張された部分に対応する枝を図2.9の2進木につけ加えると、各節点の正負の総度がすべて2の強連結グラフが得られ、ここで定義した τ_1, τ_2 が最小アクセス時間を実現できるメモリ変換の組であることがわかる。

つぎに、この τ_1, τ_2 を用いたアクセスアルゴリズムを構成する。

まず、前述の(ii)の処理を行なう部分は、右のように構成される。ここで、 A はアクセスしたいデータを記憶しているセルの番地を S 桁2進数で表わしたもので、それを、

$$A = a_{s-1} a_{s-2} \dots a_1 a_0$$

とする。 $lsd(A)$ は A の最下位のビット a_0 を表わす。また、 $right\ shift(A)$ は A を1ビットだけ右にシフトする操作である。すなわち、 $right\ shift(A)$ により A の内

```

Access Algorithm Part II
begin
  while A ≠ 1 do
    begin
      if lsd(A) = 0 then  $\tau_1$ 
      else  $\tau_2$ ;
      right shift(A)
    end;
  access to window
end

```

容は $0a_{s-1}a_{s-2}\dots a_1$ に変更される。

つぎに、(i) の処理を行なう部分について述べる。
 集合 $K = \{1, 2, \dots, k\}$ から K 自身への写像。

$$loc(i) = 2^p i + q \pmod{k} \quad ; \quad i \in K$$

を考える。 $k = 2^s - 1$ であるから、パラメータ p, q の任意の値に対して写像 loc は全単射となる。そこで、識別番号 i のついたデータを番地 $loc(i)$ のセルに記憶することにする。このようにすれば、メモリ変換 τ_1, τ_2 によりデータ i はそれぞれ、

$$\begin{aligned} \tau_1(loc(i)) &= 2^{-1} loc(i) \pmod{k} \\ &= 2^{p-1} i + 2^{-1} q \pmod{k} \end{aligned}$$

$$\begin{aligned} \tau_2(loc(i)) &= 2^{-1}(loc(i) - 1) \pmod{k} \\ &= 2^{p-1} i + 2^{-1}(q-1) \pmod{k} \end{aligned}$$

の番地のセルに転送される。したがって、メモリ変換を施すたびにパラメータ p の値を $p-1 \pmod{s}$ に変更し、 q の値をメモリ変換 τ_1 のときには $2^{-1}q \pmod{k}$ に、メモリ変換 τ_2 のときには $2^{-1}(q-1) \pmod{k}$ に変更しておくことにより、現在のデータの配置状態を把握しておくことができる。以上をまとめて、つぎに示すアクセスアルゴリズムが得られる。図 2.10 に容量が 7 ($= 2^3 - 1$) のダイナミックメモリの構造を示す。

```

procedure ACCESS(i,p,q)
begin
  A ← 2pi+q(mod'k) ;
  begin
    while A ≠ 1 do
      begin
        if lsd(A) = 0 then
          begin
            τ1 ;
            q ← 2-1q(mod'k)
          end
        else
          begin
            τ2 ;
            q ← 2-1(q-1)(mod'k)
          end ;
        p ← p-1(mod's) ;
        right shift(A)
      end ;
      access to window
    end
  end
end
    
```

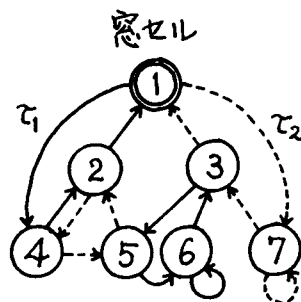


図 2.10 最小アクセス時間を実現するダイナミックメモリの構造

つぎに、ここで得られた最小アクセス時間を実現するダイナミックメモリの構造をセルループの結合に応用した網状セルループ方式を提案する。

網状セルループ方式は図2.11に示すように、メモリ変換を行なう S_1 ゲートと、メモリ変換を行なう S_2 ゲートにより各個のセルループを結合してマイナーループを構成している。記憶形式は各セルにコンテナを2個ずつ記憶している。アクセスはレコードが入っているコンテナを、前述のアクセスアルゴリズムによる順序で S_1 ゲート、 S_2 ゲートを横切って窓セルまでシフトして行なう。 S_1 ゲートと S_2 ゲートとはセルループ上に間隔 $l_c/2$ で配置されているため、セル i ($i \geq 2^2$)にあるコンテナを窓セルまでシフトするのに要する時間は、コンテナの先頭ビットをゲートまでシフトするのに要する時間、

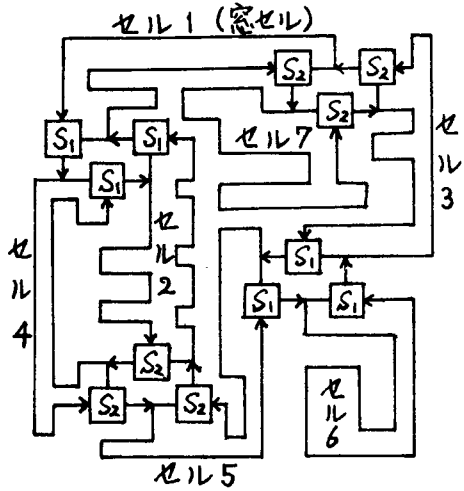


図2.11 網状セルループ方式におけるマイナーループの構造

$$\begin{cases} t_{E1} = \frac{l_c}{2} \cdot T_R \\ t_{M1} = l_c T_R \end{cases} \quad \text{----- (2.19)}$$

と、先頭ビットがゲートにそろってからゲートを開き、そのコンテナを窓セルまでシフトするのに要する時間 $t_2(i)$ との和で表わされる。ここで、 $t_2(i)$ は

$$t_2(i) = \begin{cases} \frac{l_c}{2} T_R + t_2(i) & ; \text{セル}i\text{のコンテナを窓セルに転送するメモリ変換系列が} \tau\tau \dots \text{であるとき} \\ l_c T_R + t_2(i) & ; \text{セル}i\text{のコンテナを窓セルに転送するメモリ変換系列が} \tau\tau \dots \text{であるとき} \end{cases} \quad \text{----- (2.20)}$$

である。ただし、 τ は τ_1 、または τ_2 であり、 τ_1 は τ_2 と異なるメモリ変換を表わす。セル1、2、3については、

$$t_2(1) = 0$$

$$t_2(2) = t_2(3) = \frac{l_c}{2} \cdot T_R \quad \text{----- (2.21)}$$

である。

コンテナを窓セルに転送するのに要する平均の時間は、

$$t_E = t_{E1} + \frac{1}{R} \sum_{i=1}^R t_2(i) \quad \text{----- (2.22)}$$

と表わされる。一方、深さ j (≥ 1) のセル i ($2^j \leq i \leq 2^{j+1} - 1$) と深さ $j+1$ のセル $2i$ 、 $2i+1$ について、

$$t_2(2i) + t_2(2i+1) = 2t_2(i) + \frac{3l_c}{2} \cdot T_R \quad \text{----- (2.23)}$$

が成立する。そこで、深さ j のセルについて、

$$T(j) = \sum_{i=2^j}^{2^{j+1}-1} t_2(i) \quad \text{----- (2.24)}$$

とおけば、

$$T(j+1) = 2T(j) + \frac{3l_c}{2} \cdot 2^j T_R \quad \text{----- (2.25)}$$

なる漸化式が成立する。なお、

$$T(1) = t_2(2) + t_2(3) = l_c T_R \quad \text{----- (2.26)}$$

である。この漸化式から、

$$T(j) = (3j-1) 2^{j-2} l_c T_R \quad \text{----- (2.27)}$$

が得られる。したがって、式 (2.22) は、

$$\begin{aligned} t_E &= \frac{l_c}{2} \cdot T_R + \frac{1}{R} \sum_{i=1}^R t_2(i) \\ &= \frac{l_c}{2} \cdot T_R + \frac{1}{R} \sum_{j=1}^{S-1} T(j) \\ &= \frac{l_c}{2} \left\{ 1 + \frac{(3S-7) 2^{S-1} + 4}{R} \right\} T_R \\ &= \frac{1}{4} \left\{ 3 \log_2(R+1) - 5 + \frac{3 \log_2(R+1) + 1}{R} \right\} l_c T_R \quad \text{---- (2.28)} \end{aligned}$$

となる。 R が大きいときには、

$$t_E \approx \frac{3 \log_2 R}{4} \cdot l_c T_R \quad \text{----- (2.29)}$$

である。

また、最も長く時間がかかるのは、コンテナがセル 2^{S-1} またはセル $2^S - 1$ に

あり、しかも、コンテナの先頭ビットがゲートの直後にあるときで、

$$t_M = t_{M1} + (S - 1.5) l_c T_R$$
$$= \{ \log_2 (R + 1) - 0.5 \} l_c T_R \quad \text{----- (2.30)}$$

だけの時間がかかる。Rが大きいときには、

$$t_M \approx \log_2 R \cdot l_c T_R \quad \text{----- (2.31)}$$

である。式(2.28)、(2.30)を、式(2.9)+式(2.10)と比べると、コンテナを窓セルまで転送するのに要する時間は梯子形セルループ方式ではオーダーRであったのに対し、網状セルループ方式ではオーダー $\log_2 R$ に改善されていることがわかる。例えば、R=7の場合には、梯子形セルループ方式では $t_E = 2 l_c T_R$ 、 $t_M = 4 l_c T_R$ であるのに対し、網状セルループ方式では、 $t_E = (19/14) l_c T_R$ 、 $t_M = 2.5 l_c T_R$ であり、およそ6~7割に改善されている。

2.4 結言

この章では、記憶方式の観点からバブルメモリのアクセス時間について考察し、大容量バブルメモリに適したセルループ方式という新しい記憶方式の提案を行なった。セルループ方式の特徴をまとめるとつぎのようになる。

- (1) 高価なR/W回路を共用することでその数を減らし、ビット当たりの価格を低くしている。
- (2) 従来のM/mループ方式に比べて、R/W回路の共用に起因するアクセス時間の増加が少なくて済み、アクセス時間が大幅に短縮される。
- (3) 網状セルループ方式にみられるように、セルループの結合を密にすることによりアクセス時間をさらに短縮できる。

最後に、梯子形セルループ方式では窓セルにあるコンテナの番号に近い番号のコンテナ程、窓セルから近いセルに入っているため、現在アクセスされているレコードに近いレコードがつぎにアクセスされる確率が大きいときには、実質的にアクセス時間の短縮が行なわれる。

第3章 磁気バブル論理系の解析

3.1 緒言

磁気バブル素子は、バブル相互間の反発力や外部磁界とバブルとの相互作用などを利用して論理演算を行なうことができる。これは従来のメモリ素子にはなかった機能で、第4章で述べる機能メモリのように特殊な動作を行なうデバイスを実現することができる。その際、磁気バブル論理はロジック・イン・メモリとしての重要な役割をになっている。

磁気バブル論理は、R. L. Graham [33]、A. D. Friedman、P. R. Menon [34]により集合のモデルを用いて初めて解析された。磁気バブル論理回路は、その原理を考えると本質的にはバブルの転送路を切替える回路であると言える。転送路の切替えは媒体表面に設けた導体中を流れるループ電流による局部磁界とバブルとの相互作用や、他のバブルとの反発を利用して行なわれる。前者はバブル・電流論理と呼ばれ、第2章で述べたトランスファゲートやスイッチゲートなどのゲート動作を行なう箇所に用いられる。後者はバブル・バブル論理と呼ばれる。いずれの場合にも論理演算中にバブルが新しく発生したり、消滅したりすることがない。すなわち、保存性が磁気バブル論理の大きな特徴であると言える。この章では、磁気バブル論理の保存性を記述するのに適したモデルとして R. M. Sandfort と E. R. Burke [31]による二入力保存形論理セルを選び、これを用いて磁気バブル論理系の性質を明らかにする。また、論理関数の標準的な構成法についても考察する。

3.2 磁気バブル論理系のモデル

この節では、以下で用いる磁気バブル論理系のモデルを示す。ここで考察する磁気バブル論理はバブル相互間の反発を利用したバブル・バブル論理であり、バブルの有無をそれぞれ1、0に対応させた二値論理である。

反発した後、バブルが移動する位置は図3.1に示す4通りの場合が可能である。図3.1に示した4種類の二入力保存形論理セルを基本論理セルと呼ぶ。基本論理セルのほかに、保存的なバブル回路として図3.2に示す伝送と交叉を考える。これらは論理回路を構成する際に、或る変数に対応するバブルを別の場所で利用するときに必要な回路である。また、図3.3に示すバブル発生器(Gと書く)とバブル消去器は、非保存的な論理回路の構成に必要なバブル回路である。ここでは、伝送、交叉、バブル発生器、バブル消去器、および

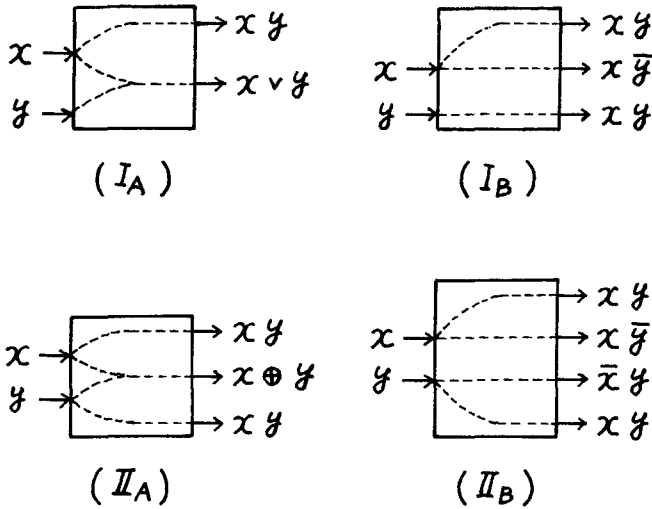


図3.1 基本論理セル

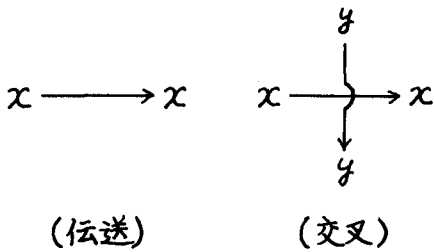


図3.2 伝送と交叉

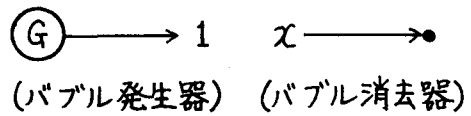


図3.3 バブル発生器と消去器

いくつかの種類の基本論理セルよりなる系を磁気バブル論理系のモデルとし、これを基本論理セルの集合で表わすことにする。なお、ここにあげたバブル回路は例えば回転駆動磁界とパーマロイ磁極パターンによる磁極法、あるいは、ハニカム形駆動方式による導体法などにより物理的に磁気バブル素子で実現することができる。

3.3 磁気バブル論理系による論理関数の実現

この節では、各論理系による論理関数の実現を、使用されるGの個数に着目して考察し、各論理系の性質を明らかにする。ただし、或る論理系による論理

関数の実現とは、その系を構成する基本論理セルと伝送、交叉、およびバブル発生器、バブル消去器からなる回路で、各入力変数をただ一度用いてその論理関数を実現することをいう。以下に述べる補題と定理が成立する。

〔補題 3.1〕 最小項 $m_i^{(n)}$ を含む n 変数関数 $f(x_1, x_2, \dots, x_n)$ ($n \geq 1$) (以下では $f(X)$ と書く) は G を含まない回路では実現されない。ただし、

$$m_i^{(n)} \triangleq x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}$$

$i = i_1 i_2 \dots i_n$ (n 桁 2 進表示), $x^1 \triangleq x$, $x^0 \triangleq \bar{x}$ である。

(証明) f が G を含まない回路で実現されると仮定する。 $f \geq m_i^{(n)}$ であるから、 $f(0, 0, \dots, 0) = 1$ である。これは回路にバブル入力がないときにもバブル出力があることを意味しており、 G を含まない回路の内部ではバブルは発生されないということと矛盾する。 (証明終)

〔補題 3.2〕 論理系 $\{I_A\}$ では、 G をいくつ用いても実現できない n 変数関数 ($n \geq 1$) が存在する。

(証明) 変数と定数と論理演算・(論理積)、 \vee (論理和) により構成される関数は単調関数である。すなわち、 $f(X)$ は、

$$a_1 \leq b_1, a_2 \leq b_2, \dots, a_n \leq b_n$$

$$\Leftrightarrow f(a_1, a_2, \dots, a_n) \leq f(b_1, b_2, \dots, b_n)$$

を満足する。

したがって、論理系 $\{I_A\}$ により実現される関数は単調関数である。一方、単調でない n 変数関数 ($n \geq 1$) が存在する。 (証明終)

〔補題 3.3〕 論理系 $\{I_B\}$ ($\{II_B\}$) により G を 1 個用いて、すべての n 変数最小項 ($n \geq 1$) を発生することができる。

(証明) 変数の数 n に関する帰納法により証明する。

論理系 $\{I_B\}$ の場合。

$n=1$ のとき、図 3.1 において変数 x を G に置き換えることにより出力 y が得られる。すなわち、補題が成立している。

$n=l$ のとき、補題が成立すると仮定する。

$n=l+1$ のとき、図 3.4 により補題の成立は明らかである。

論理系 $\{II_B\}$ の場合も同様にして証明される。(図 3.5 を参照) (証明終)

〔補題 3.4〕 論理系 $\{I_B\}$ ($\{II_B\}$) により G を 2 個用いて、任意の n 変数関数 ($n \geq 1$) を実現することができる。

(証明) 論理系 $\{I_B\}$ の場合。

任意の n 変数関数 $f(X)$ はいくつかの最小項の否定の積の形、

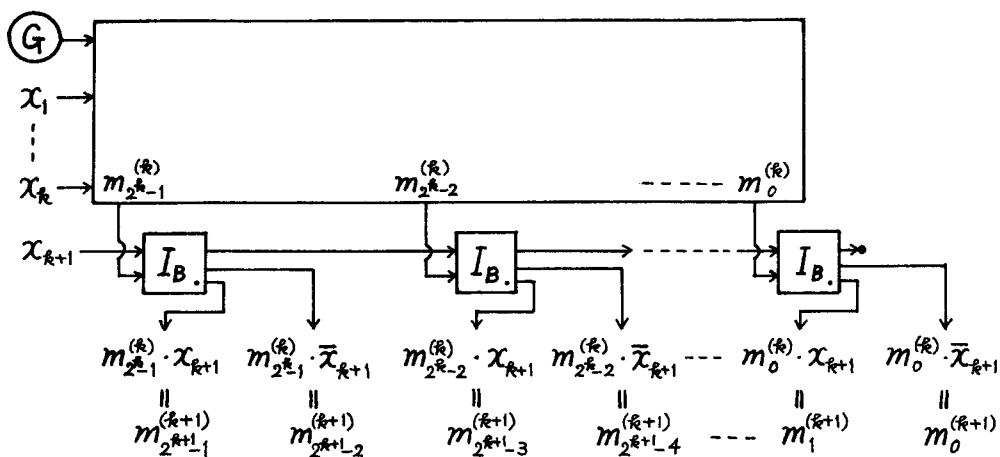


図3.4 論理系 $\{I_B\}$ による最小項の発生
(\cdot は論理積出力を表わす)

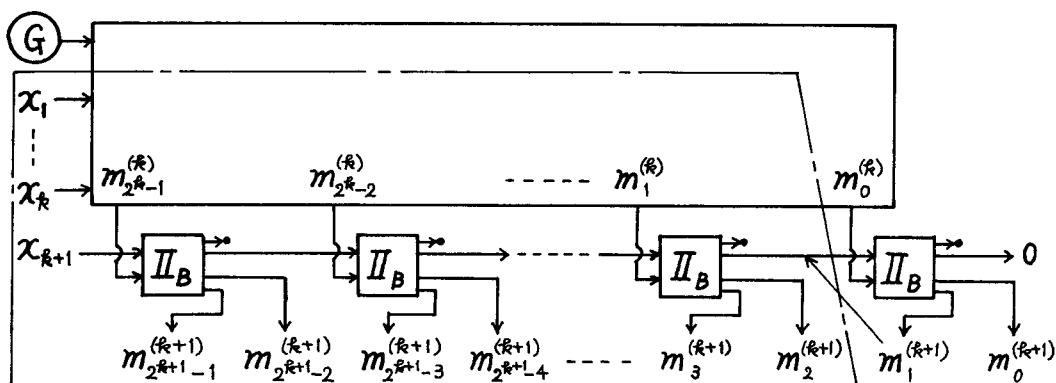


図3.5 論理系 $\{II_B\}$ による最小項の発生
(一点鎖線内は m_0 以外の最小項の発生)

$$f(x) = 1 \cdot \overline{m_i^{(n)}} \cdot \overline{m_j^{(n)}} \cdots \overline{m_r^{(n)}} \quad \text{----- (3.1)}$$

に表わすことができる。式(3.1)は、 f が図3.6のように実現されることを意味している。このことと補題3.3とから補題の成立がいえる。

論理系 $\{II_B\}$ の場合も同様にして証明される。

(証明終)

〔補題3.5〕 論理系 $\{II_A\}$ により G を1個用いて、すべての n 変数最小項($n \geq 1$)を発生することができる。とくに、 $m_0^{(n)}$ 以外の最小項は2個ず

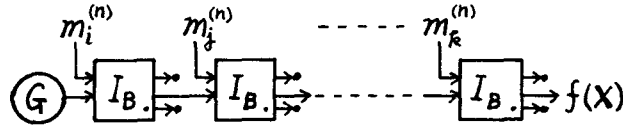


図 3.6 論理系 $\{I_B\}$ による $f(X)$ の実現

つ発生することができる。

(証明) 補題 3.3 の証明と同様に、帰納法により証明される。(図 3.7 を参照) (証明終)

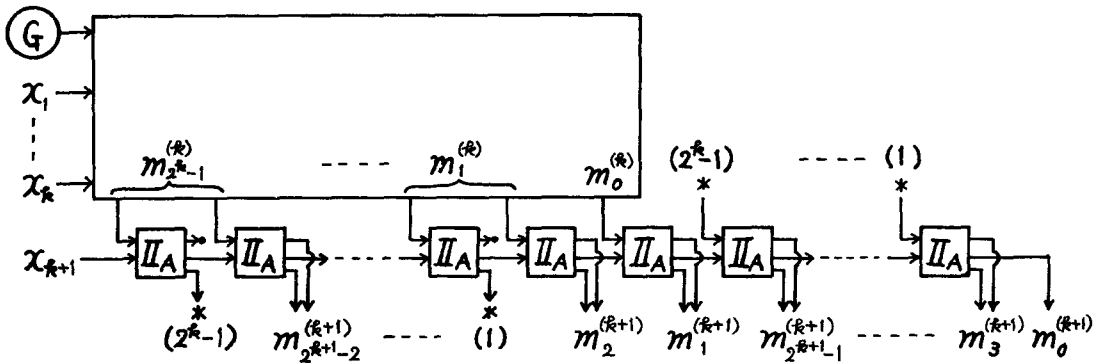


図 3.7 論理系 $\{II_A\}$ による最小項の発生
($\rightarrow*(i)$, $(i)*\rightarrow$ は接続を表わす)

[補題 3.6] 論理系 $\{II_A\}$ により G を 1 個用いて、任意の n 変数関数 ($n \geq 1$) を実現することができる。

(証明) 任意の n 変数関数 $f(X)$ はいくつかの最小項の排他的論理和の形。

$$f(X) = m_i^{(n)} \oplus m_j^{(n)} \oplus \dots \oplus m_k^{(n)} \quad \text{----- (3.2)}$$

に表わすことができる。

式 (3.2) は、 f が図 3.8 のように実現されることを意味している。このことと補題 3.5 とから、補題の成立がいえる。
(証明終)

以上で、 $\{I_A\}$ 以外の論理系は任意の関数を実現できるとい

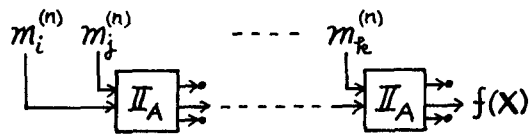


図 3.8 論理系 $\{II_A\}$ による $f(X)$ の実現

う意味で完全系であることがわかった。

つぎに、各論理系について、関数の種類とその関数を実現するために必要な G の個数との関係を考察する。

[定義 3.1] G_{IO} 関数
定数 0 と変数を 0 段 G_{IO} 関数という。

$$f(X) = x_i x_j g_1(X/i, j) \vee (x_i \vee x_j) g_2(X/i, j) \vee g_3(X/i, j)$$

----- (3.3)

と表わすことができ、かつ、

$$f'(X') \triangleq x'_i g'_1(X'/i, j) \vee x'_j g'_2(X'/i, j) \vee g'_3(X'/i, j)$$

が $(m-1)$ 段 G_{IO} 関数 ($m \geq 1$) であるような変数の組 x_i, x_j が存在するとき、 $f(X)$ を m 段 G_{IO} 関数という。ただし、 $g_r(X/i, j)$ ($r=1, 2, 3$) は $\{x_1, x_2, \dots, x_n\} - \{x_i, x_j\}$ よりなる $(n-2)$ 変数関数を表わし、 $g'_r(X'/i, j)$ は $g_r(X/i, j)$ の変数名をかえて得られる関数を表わす。0 以上の或る整数値 m に対して m 段 G_{IO} 関数であるとき、この関数を G_{IO} 関数と定義する。 (定義終)

[補題 3.7] $f(X)$ が論理系 $\{I_A\}$ により G を用いないで実現されるための必要十分条件は、 $f(X)$ が G_{IO} 関数であることである。

(証明) (十分) G_{IO} 関数の段数 m に関する帰納法により証明する。

0 段 G_{IO} 関数は定数 0 と変数であるから、 G を用いないで実現されることは明らかである。

l 段 G_{IO} 関数は G を用いないで実現されると仮定する。

$f(X)$ を $(l+1)$ 段 G_{IO} 関数とする。 $f(X)$ は式 (3.3) ように表わされ、かつ、

$$f'(X') \triangleq x'_i g'_1(X'/i, j) \vee x'_j g'_2(X'/i, j) \vee g'_3(X'/i, j)$$

が l 段 G_{IO} 関数であるから、

$f(X)$ は図 3.9 のように G を用いないで実現される。

(必要) $f(X)$ を実現している回路に含まれる基本論理セル I_A の個数に関する帰納法により証明する。

$f(X)$ が G も I_A も用いずに実現されるならば、 $f(X)$ は定数 0

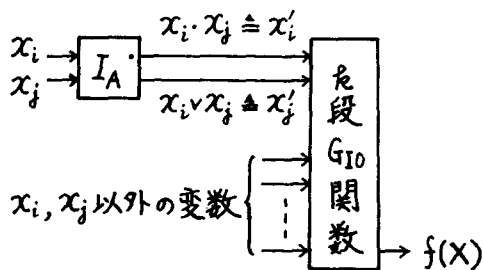


図 3.9 論理系 $\{I_A\}$ による G_{IO} 関数の実現

または、変数 x_p ($1 \leq p \leq n$) である。すなわち、 $f(X)$ は G_{IO} 関数である。
 l 個の I_A を含み、 G を含まない回路で実現される関数は G_{IO} 関数であると仮定する。

$f(X)$ を $(l+1)$ 個の I_A を含み、 G を含まない回路で実現される関数で、 0 でも x_p ($1 \leq p \leq n$) でもないとする。このとき、 $f(X)$ を式 (3.3) のように表わせる変数の組 x_i, x_j が存在する。なぜならば、 $f(X)$ を実現している回路において 2 入力とも変数であるような I_A が少なくとも 1 個存在し、かつ、各入力変数はただ 1 度しか用いられないので、この二つの変数 x_i, x_j に関しては $x_i \cdot x_j, x_i \vee x_j$ を単位として扱われることになる。いま、 $f(X)$ を実現している回路から、 x_i, x_j を入力とする I_A を取り除いて得られる l 個の I_A よりなる回路を考える (図 3.10 を参照)。この回路により実現される関数は、

$$f'(X') = x'_i g'_1(X'/i, j) \vee x'_j g'_2(X'/i, j) \vee g'_3(X'/i, j)$$

と表わされ、かつ、帰納法の仮定により G_{IO} 関数 (m 段) である。したがって、もとの関数 $f(X)$ は $(m+1)$ 段 G_{IO} 関数、すなわち、 G_{IO} 関数である。

(証明終)

ここで、 G (すなわち、定数 1) を I_A の入力とした場合を考える。図 3.1 において変数 x を G に置き換

えたとき出力は $x \cdot 1 = x, x \vee 1 = 1$ となり入力と同じである。すなわち、 G は I_A の入力としては冗長である。このことと補題 3.7 とからつぎの定理が得られる。

[定理 3.1] 論理系 $\{I_A\}$ では、

- (i) G_{IO} 関数は G を用いずに実現される。
- (ii) $f(X) \equiv 1$ なる関数を実現するために必要かつ十分な G の個数は 1 個である。
- (iii) (i)、(ii) 以外の関数は G をいくつ用いても実現されない。

(定理終)

つぎに、論理系 $\{I_B\}$ および $\{II_B\}$ について考察する。

[補題 3.8] 論理系 $\{I_B\}$ ($\{II_B\}$) により G を用いずに、 $m_0^{(n)}$ 以外のすべての n 変数最小項 ($n \geq 1$) を発生することができる。

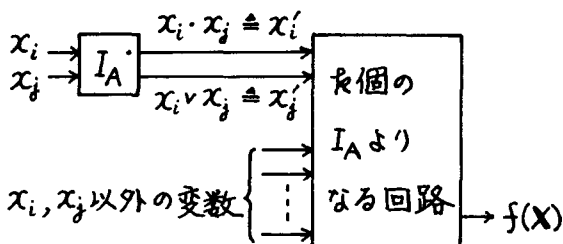


図 3.10 論理系 $\{I_A\}$ により実現される関数

(証明) 変数の数 n に関する帰納法により証明する。

論理系 $\{I_B\}$ の場合。

$n=1$ のとき、 $m_0^{(1)}$ 以外の最小項は変数自身のみであるから補題の成立は明らかである。

$n=k$ のとき、補題が成立すると仮定する。

$n=k+1$ のとき、図 3.11 により補題の成立は明らかである。

論理系 $\{II_B\}$ の場合も同様にして証明される (図 3.5 の一点鎖線で囲まれた部分を参照)。(証明終)

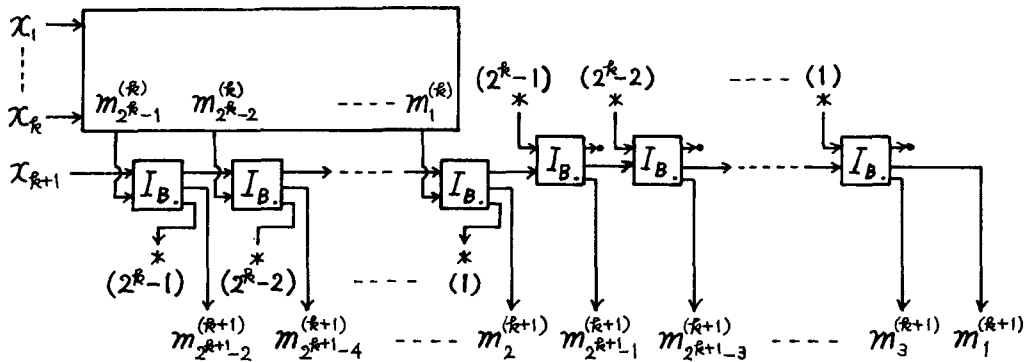


図 3.11 論理系 $\{I_B\}$ による、 m_0 以外の最小項の発生

[定理 3.2] 論理系 $\{I_B\}$ ($\{II_B\}$) では、

(i) $f_{01}(X) \equiv 0$ なる関数、

$f_{02}(X) = x_{i_1} x_{i_2} g(X/i_1, i_2)$ の形に書ける関数、および、

$f_{03}(X) = x_{i_1} x_{i_2} \dots x_{i_p} g(X/i_1, i_2, \dots, i_p)$ ($p \geq 1$) の形に書ける変数の組 $x_{i_1}, x_{i_2}, \dots, x_{i_p}$ が存在し、かつ、 $g(X/i_1, i_2, \dots, i_p) \geq m_0^{(n-p)}$ である関数は、 G を用いずに実現される。

(ii) $f_{11}(X) \geq m_0^{(n)}$ なる関数、および、

$f_{12}(X) = x_i g(X/i)$ の形に書け、かつ、 $g(X/i)$ がつぎの条件 (iii) を満たす $(n-1)$ 変数関数であるような関数を実現するために必要かつ十分な G の個数は 1 個である。

(iii) $f_2(X) \neq m_0^{(n)}$ 、かつ $f_2(X) = x_i g(X/i)$ の形に書ける変数 x_i が存在しない関数を実現するために必要かつ十分な G の個数は 2 個である。

(証明) (i) f_{01} については明らかである。

$f_{02}(X)$ は条件より、

$$f_{02}(X) = x_{i_1} x_{i_2} \overline{(x_{i_2} m_i^{(n-2)})} \overline{(x_{i_2} m_j^{(n-2)})} \dots \overline{(x_{i_2} m_k^{(n-2)})} \quad (3.4)$$

のように表わすことができる。ただし、 $m_i^{(n-2)}$ などは $\{x_1, x_2, \dots, x_n\} - \{x_{i_1}, x_{i_2}\}$ よりなる $(n-2)$ 変数最小項である。式(3.4)は f_{02} が図3.12のように実現されることを意味している。一方、 $x_{i_2} m_i^{(n-2)}$ などは $\{x_1, x_2, \dots, x_n\} - \{x_{i_1}\}$ よりなる $m_0^{(n-1)}$ 以外の $(n-1)$ 変数最小項であり補題3.8により G を用いずが発生される。したがって、 f_{02} は G を用いずを実現される。

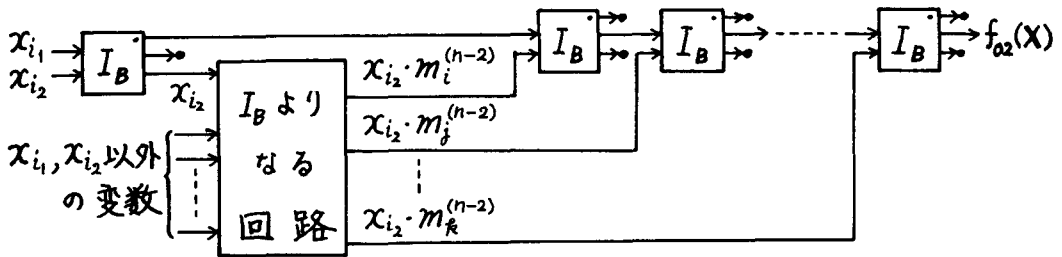


図3.12 論理系 $\{I_B\}$ による $f_{02}(X)$ の実現

$f_{03}(X)$ は、 $f \geq m_0^{(n-p)}$ であるから変数と $m_0^{(n-p)}$ 以外の最小項の否定の積の形、

$$f_{03}(X) = x_{i_1} x_{i_2} \dots x_{i_p} \overline{m_i^{(n-p)}} \overline{m_j^{(n-p)}} \dots \overline{m_k^{(n-p)}} \quad \text{----- (3.5)}$$

に表わすことができる。式(3.5)は f_{03} が図3.13のように実現されることを意味している。一方、補題3.8により G を用いず $m_0^{(n-p)}$ 以外のすべての $(n-p)$ 変数最小項が発生することができる。したがって、 f_{03} は G を用いず実現される。

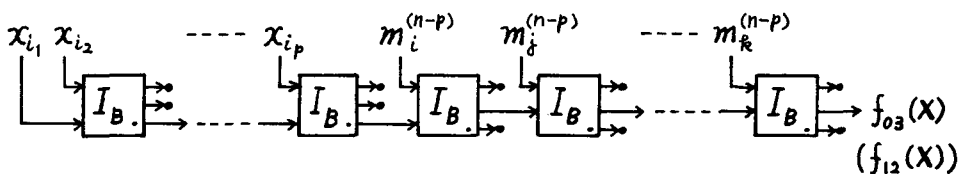
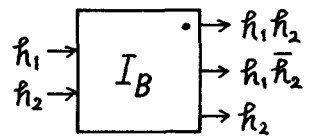


図3.13 論理系 $\{I_B\}$ による $f_{03}(X)$ ($f_{12}(X)$)の実現

(ii) $f_{11} \geq m_0^{(n)}$ であるから、 f_{11} は $m_0^{(n)}$ 以外の最小項を用いて式(3.1)のように表わすことができる。すなわち、 f_{11} は図3.6のように実現される。図3.6と補題3.8とから、 f_{11} は G を1個用いて実現される。また、補題3.1により f_{11} を実現するために G が必要である。
 f_{12} については最後に証明する。

(iii) f_2 が G を2個用いて実現されることは補題3.4により明らかである。条件より、 f_2 を和積形式で表現したとき簡単化されない項 $(x_{j_1} \vee x_{j_2} \vee \dots \vee x_{j_s})$ ($s \geq 2$) が含まれる。 f_2 を実現している回路における最終段の I_B (すなわち、 f_2 を出力とする I_B) を考えると、簡単化されない項 $(x_{j_1} \vee x_{j_2} \vee \dots \vee x_{j_t})$ ($t \geq s$) がこの I_B の入力 r_1, r_2 , あるいは \bar{r}_2 に含まれねばならない(図3.14を参照)。この項が r_1 あるいは r_2 に含まれる場合には、 r_1 あるいは r_2 を新たに実現されるべき関数 f とみなすと f もまた定理の条件(iii)を満たす。そして、 f の実現における最終段の I_B について前と同様に簡単化されない項 $(x_{j_1} \vee x_{j_2} \vee \dots \vee x_{j_u})$ ($u \geq t$) がこの I_B の入力 r_1, r_2 , あるいは \bar{r}_2 に含まれねばならない。



$$f_2 \in \{r_1, r_2, r_1 \bar{r}_2, r_2\}$$

図3.14 最終段の I_B

このように簡単化されない項 $(x_{j_1} \vee x_{j_2} \vee \dots \vee x_{j_u})$ を含む関数を実現するときに、常にこの関数の実現における最終段の I_B の入力 r_1 あるいは r_2 に簡単化されない項 $(x_{j_1} \vee x_{j_2} \vee \dots \vee x_{j_v})$ ($u \geq v$) を含ませる実現方法を用いると、出力端からさかのぼっていったとき永久に入力端(変数あるいは定数)に到達できない。すなわち、このような実現方法では f_2 を実現できない。したがって、入力 r_1, r_2 には簡単化されない項 $(x_{j_1} \vee x_{j_2} \vee \dots \vee x_{j_t})$ ($t \geq s$) を含まず、 \bar{r}_2 にこの項を含み、定理の条件(iii)を満たす関数を出力とする I_B が回路中に少なくとも1個存在する。この I_B の出力 $r_1 \bar{r}_2$ が定理の条件(iii)を満たす関数である。この I_B の入力 r_1 , および r_2 はどちらも $m_0^{(n)}$ を含んでいる。したがって、補題3.1により r_1, r_2 を実現するためにそれぞれ G が必要である。すなわち、 f_2 を実現するために少なくとも G が2個必要である。

最後に、 f_2 について証明する。条件より、 f_2 は式(3.5)のように表わすことができる。したがって、 f_2 は図3.13のように実現される($p=1$ の場合)。図3.13と補題3.3により f_2 は G を1個用いて実現される。つぎに、 $f_2(x) = x_i g(x/i)$ が G を含まない回路で実現されるとすると、この回路において変数 x_i を G に置き換えて得られる回路(G を1個含む)により関数 $g(x/i)$ が実現されることになる。一方、 $g(x/i)$ は定理の条件(iii)を満たす $(n-1)$ 変数関数である。これは(iii)の結果に矛盾する。

論理系 $\{I_B\}$ の場合も同様にして証明される。 (証明終)

つぎに、論理系 $\{I_A\}$ について考察する。

[定義3.2] G_{II_0} 関数

1変数 G_{II_0} 関数は、定数0と変数自身である。
対称な変数 x_i, x_j が存在し、

$$f(x) = x_i x_j g_1(x/i, j) \oplus (x_i \oplus x_j) g_2(x/i, j) \oplus g_3(x/i, j) \quad \text{----- (3.6)}$$

と表わしたとき、

$$f'(x') \cong x'_{n-1} g'_2(x'/n-1) \oplus g'_3(x'/n-1)$$

が $(n-1)$ 変数 G_{II_0} 関数 ($n \geq 2$) であるとき、 $f(x)$ を n 変数 G_{II_0} 関数という。ただし、 $g'_r(x'/n-1)$ ($r = 2, 3$) は $g_r(x/i, j)$ の変数名をかえて得られる関数を表わす。 (定義終)

[補題 3.9] $f(x)$ が論理系 $\{II_A\}$ により G を用いずに実現されるための必要十分条件は、 $f(x)$ が G_{II_0} 関数であることである。

(証明) (十分) 変数の数に関する帰納法により証明する。

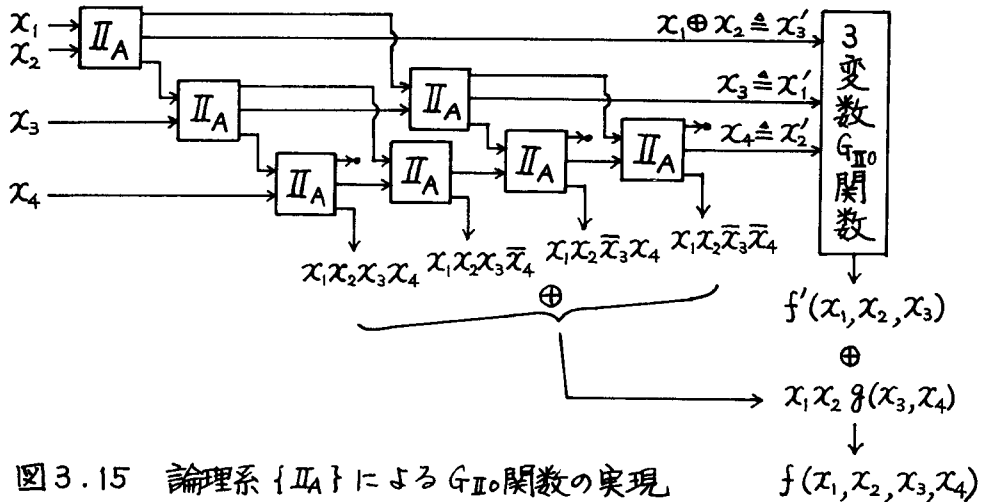
1 変数 G_{II_0} 関数 (すなわち、定数 0 と変数) が G を用いずに実現されることは明らかである。

l 変数 G_{II_0} 関数は G を用いずに実現されると仮定する。

$f(x)$ を $(l+1)$ 変数 G_{II_0} 関数とする。 $f(x)$ は式 (3.6) のように表わされ、かつ、

$$f'(x') \cong x'_l g'_2(x'/l) \oplus g'_3(x'/l)$$

が l 変数 G_{II_0} 関数であるから、 f は図 3.15 のように G を用いずに実現される。図 3.15 は $i = 1, j = 2, l = 3$ の場合であるが、一般の場合も明らかであろう。



(必要) 変数の数に関する帰納法により証明する。

補題 3.1 により、1 変数関数 $f(x)$ が G を用いずに実現されるならば、 $f(x)$ は 0 または x である。すなわち、 $f(x)$ は G_{II} 関数である。

G を含まない回路で実現される n 変数関数は G_{II} 関数であると仮定する。

$f(x)$ を G を含まない回路で実現される $(n+1)$ 変数関数とする。このとき、対称な変数の組が少なくとも 1 組存在する。なぜならば、 $f(x)$ を実現している回路において 2 入力とも変数であるような Π_A が少なくとも 1 個存在し、この二つの変数 x_i, x_j に関しては $x_i x_j, x_i \oplus x_j$ を単位として取り扱われることになり、 $f(x)$ はこれらの変数に関して対称となる。そこでいま、 $f(x)$ が x_i, x_j に関して対称であると仮定する。このとき、 $f(x)$ は式 (3.6) のように表わされる。ここで $x_j = 0$ とおくと、

$$f(x | x_j = 0) = x_i g_2(x/i, j) \oplus g_3(x/i, j)$$

となる。この n 変数関数 $f(x | x_j = 0)$ は G を用いずに実現される。すなわち、 $f(x)$ を実現している G を含まない回路から変数 x_j および冗長な基本論理セル Π_A を取り除いて得られる回路により $f(x | x_j = 0)$ は実現されている。したがって、この関数の変数名をかえて得られる n 変数関数、

$$f'(x') \equiv x'_n g'_2(x'/n) \oplus g'_3(x'/n)$$

は G を含まない回路で実現される。帰納法の仮定により $f'(x')$ は (n) 変数 G_{II} 関数である。したがって、 $f(x)$ は $(n+1)$ 変数 G_{II} 関数である。

(証明終)

補題 3.6 と補題 3.9 とからつぎの定理が得られる。

[定理 3.3] 論理系 $\{\Pi_A\}$ では、

(i) G_{II} 関数は G を用いずに実現される。

(ii) G_{II} 関数以外の関数を実現するために必要かつ十分な G の個数は 1 個である。

(定理終)

最後に、複数種類の基本論理セルよりなる複合論理系に関して成立する定理をいくつか示す。

[定理 3.4] 論理系 $\{I_A, \Pi_A\}$ に関して、定理 3.3 が成立する。

(定理終)

[定理 3.5] 論理系 $\{I_B, \Pi_B\}$ に関して、定理 3.2 が成立する。

(定理終)

[定理 3.6] 論理系 $\{I_A, I_B\}$ ($\{I_A, \Pi_B\}, \{I_A, I_B, \Pi_A\}, \{I_A, I_B, \Pi_B\}, \{I_A, \Pi_A, \Pi_B\}, \{I_A, I_B, \Pi_A, \Pi_B\}, \{I_B, \Pi_A\}, \{I_B, \Pi_A, \Pi_B\}, \{\Pi_A, \Pi_B\}$) では、

- (i) $f(x) \neq m_0^{(n)}$ なる関数は G を用いずに実現される。
- (ii) $f(x) \geq m_0^{(n)}$ なる関数を実現するために必要かつ十分な G の個数は 1 個である。 (定理終)

3.4 結 言

この章では、バブル・バブル論理の保存性を記述するのに適した二入力保存形論理セルを基本論理セルに選び、それによる論理関数の構成を行なって、バブルの保存性の観点から磁気バブル論理系の性質を明らかにした。

最後に、非保存的な回路としてバブル分割回路をも考えた場合には、定理 3.1 ~ 定理 3.3 はそれぞれつぎのように変更される。

[定理 3.1'] 論理系 $\{I_A\}$ では、

- (i) 定数 1 以外の単調関数は G を用いずに実現される。
- (ii) $f(x) \equiv 1$ なる関数を実現するために必要かつ十分な G の個数は 1 個である。
- (iii) 単調関数でない関数は G をいくつ用いても実現されない。 (定理終)

[定理 3.2'] 論理系 $\{I_B\}$ ($\{II_B\}$) では、

- (i) 関数 f_{01} , f_{02} , f_{03} , および f_{12} は G を用いずに実現される。
- (ii) 関数 f_{11} および f_2 を実現するために必要かつ十分な G の個数は 1 個である。 (定理終)

[定理 3.3'] 論理系 $\{II_A\}$ では、

- (i) $f(x) \neq m_0^{(n)}$ なる関数は G を用いずに実現される。
- (ii) $f(x) \geq m_0^{(n)}$ なる関数を実現するために必要かつ十分な G の個数は 1 個である。 (定理終)

第4章 磁気バブル応用装置

4.1 緒言

現段階では、磁気バブル素子応用の最大の目標は大容量バブルメモリにおかれており、第2章でも述べたようにその研究開発は活発でバブルメモリは実用化の段階に達している。[17]~[24]

一方、磁気バブル素子の記憶機能以外の特質を利用した装置の構成に関する研究もなされており、メモリ装置以外の新しい応用装置がいくつか発表されている。[49]~[56] いずれも実験あるいは方式設計の段階ではあるが、磁気バブル素子応用の新しい可能性を示しており注目される。

この章では、磁気バブル素子の論理機能および二次元性を利用した特殊な機能を有するメモリの構成例をいくつか提案し、磁気バブル素子応用の新しい方向を示す。とくに、メモリ内部に或る種の情報処理機能を有するメモリ、すなわち、機能メモリは、磁気バブル素子の特質から考えて最も有望な応用装置のひとつである。

4.2 スイッチング機能をもつ二次元バブルメモリ

磁気バブル素子は、媒体平面内でのバブルの駆動が可能なことから二次元性を有している。このことは、二次元情報処理装置への応用の可能性を示唆している。この節では、二次元処理装置の一例としてスイッチング機能をもつ二次元バブルメモリの構成を行なう。この装置は素子の二次元性を利用して二次元情報を直並列的に処理するため、処理時間が短い。論理回路には動作マージンが大きなバブル・電流論理回路、すなわち、ゲート回路を用いている。装置の機能は二次元アレイ情報を記憶し、メモリ内での記憶位置を二次元的に変換することである。記憶されている二次元アレイ情報を図形情報と見なせば、この装置を図形処理に応用することができる。すなわち、図形の平行移動、回転、拡大、縮小、あるいはこれらの基本的な処理をいくつか組合わせたような複雑な図形処理を行なうことができる。しかも、任意の処理をオーダ n の一定時間で行なうことができる。*ただし、 $n \times n$ の大きさの二次元アレイ情報を考えている。

* 後述するように、処理時間は $K(2n + \log_2 n) T_R$ である。ただし、 T_R は回転駆動磁界の周期であり、 K は3あるいは5である。

4.2.1 2次元変換アレイ

装置の機能を数学的に記述するため、2次元変換アレイという概念を導入する。

[定義 4.1] 2次元変換アレイ $T^{n \times n}$ はつぎに示すような $n \times n$ 2次元アレイの集合 $M^{n \times n}$ から $M^{n \times n}$ への写像である。

$$T^{n \times n} = \begin{pmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,n} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,n} \\ \vdots & \vdots & & \vdots \\ t_{n,1} & t_{n,2} & \cdots & t_{n,n} \end{pmatrix} : M^{n \times n} \rightarrow M^{n \times n}$$

$$t_{i,j} = (k, l) \text{ または } \times.$$

ただし、 $1 \leq i, j, k, l \leq n$ である。

いま、

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,l} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,l} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{k,1} & a_{k,2} & \cdots & a_{k,l} & \cdots & a_{k,n} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,l} & \cdots & a_{n,n} \end{pmatrix} \in M^{n \times n}$$

としたとき、

$$T^{n \times n}(A) = \begin{pmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,j} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,j} & \cdots & b_{2,n} \\ \vdots & \vdots & & \vdots & & \vdots \\ b_{i,1} & b_{i,2} & \cdots & b_{i,j} & \cdots & b_{i,n} \\ \vdots & \vdots & & \vdots & & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,j} & \cdots & b_{n,n} \end{pmatrix} \in M^{n \times n}$$

$$b_{i,j} = \begin{cases} a_{k,l} & ; t_{i,j} = (k, l) \text{ のとき} \\ \text{不定} & ; t_{i,j} = \times \text{ のとき} \end{cases}$$

である。

(定義終)

装置は2次元変換アレイで表わされる写像機能を実現する。すなわち、平面内に記憶された2次元アレイ情報の (k, l) 成分を位置 (i, j) に移す。

(例) 2次元変換アレイ

$$T^{4 \times 4} = \begin{pmatrix} (2, 1) & (3, 1) & (4, 1) & (1, 1) \\ (2, 2) & (3, 2) & (4, 2) & (1, 2) \\ (2, 3) & (3, 3) & (4, 3) & (1, 3) \\ (2, 4) & (3, 4) & (4, 4) & (1, 4) \end{pmatrix} \quad \text{により}$$

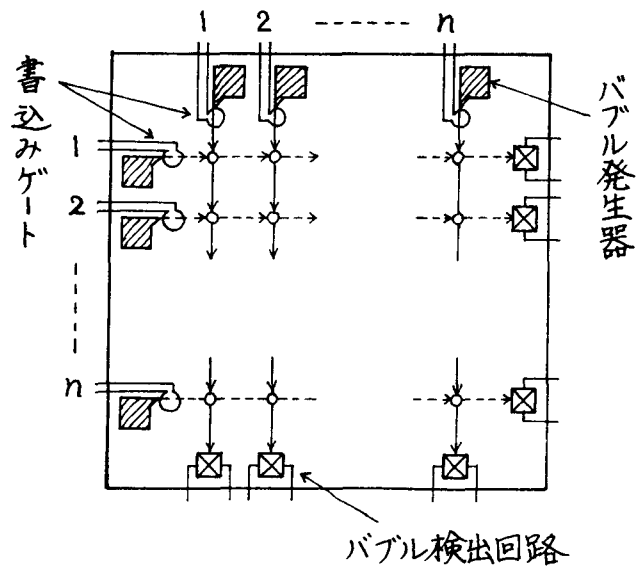
$$A = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \\ 1 & \cdot & 1 & \cdot \\ 1 & \cdot & \cdot & \cdot \end{pmatrix} \in M^{4 \times 4} \quad \text{を变换すると.}$$

$$T^{4 \times 4}(A) = \begin{pmatrix} 1 & 1 & 1 & \cdot \\ 1 & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot \end{pmatrix} \in M^{4 \times 4} \quad \text{となる.}$$

(例終)

4.2.2 装置の方式と回路

情報は図4.1に示す2次元シフトアレイと呼ばれる機能平面に2次元アレイ状に記憶される。2次元シフトアレイは情報の書込み、縦横方向へのシフト、および読出し機能を備えた平面である。その回路構成は文献[52]に示されているように正逆回転磁界駆動による二つのシフトモードをもち、 $n \times n$ 2次元アレイ情報を縦(横)方向シフトモードのもとで平面の上(左)端より書込み、下(右)方向にシフトして下(右)端より読出



実線矢印 : 縦方向転送回路

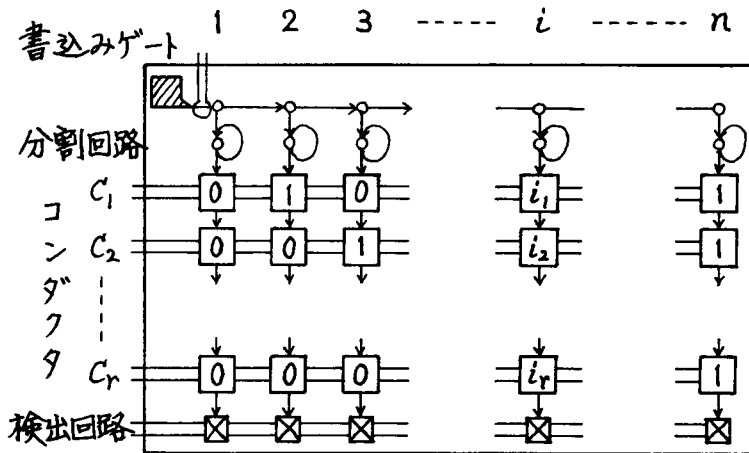
破線矢印 : 横方向転送回路

図4.1 2次元シフトアレイ

すことができる。

磁気バブル素子ではバブルが平面内を移動して情報の伝達が行なわれるため平面に記憶された二次元アレイ情報の位置変換をその平面内で行なおうとすれば、隣接した位置間での変換を繰返さねばならない。このような変換方式ではバブルの制御が複雑となり、平面内のバブル全体を一様に駆動することは困難である。また、変換によっては処理時間が非常に長くなることがある。

この点を解決するため、ここで述べる装置では図4.2に示す変換回路により情報の位置変換を時間変換として処理する方式をとっている。



$$i-1 = i_r i_{r-1} \dots i_1 \text{ (r桁2進表示)}$$

図4.2 変換回路

変換回路の動作原理はつぎのとおりである。まず、処理されるべき長さnの情報系列を書込みゲートより右方向にシフトしながら平面に書込む。この際、変換回路を過去に使用したために分割回路およびスイッチ回路中に残っているバブルを、分割回路を動作させずに下方方向に転送して消去することにより回路の初期設定を同時に行なう。つぎに、いま書込んだバブルを分割回路に転送する。分割回路で分割したバブルの一方はソースバブルとして分割回路に保持し、もう一方のバブルを下方方向に転送する。このとき、平面に配線された導体 C_1, C_2, \dots, C_r ($n=2^r$)に流す電流の有無により途中に設けられたスイッチのオン・オフを制御して、或る位置の情報を検出回路まで転送することができる。スイッチには1オンスイッチと0オンスイッチの2種類があり、その回路は図4.3に示すとおりである。1オンスイッチでは、導体に電流が流れている

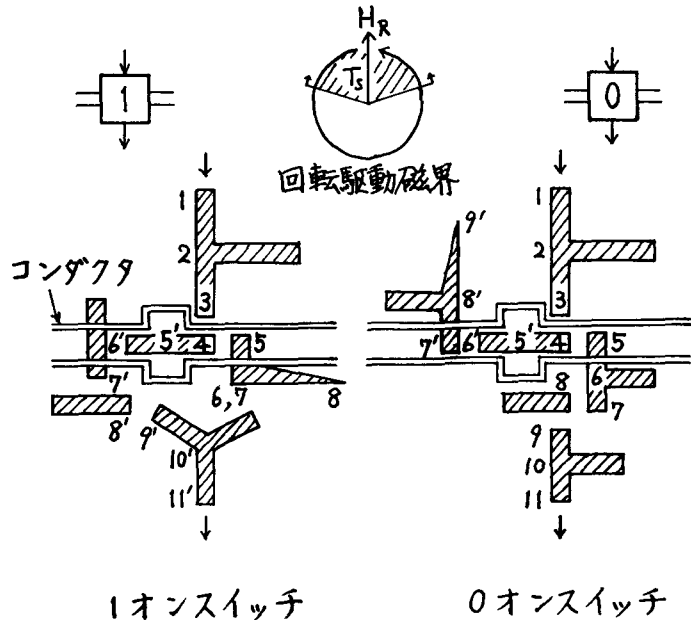
ときにはその誘導磁界により、上から転送されてきたバブルは吸引され、以後の転送路が変更されて下に移動する。すなわち、情報は下方に伝達する。一方、導体に電流が流れていないときにはバブルは消去回路に転送されそこで消滅する。すなわち、情報はそこで遮断される。

0オンスイッチではこれとは逆に、導体に電流が流れているときには情報が遮断され、電流が流れていないときには情報は下方に伝達する。

2種類のスイッチを適当に配置しておけば、導体に流す電流の有無により或る位置の情報のみを検出回路まで伝達することができる。図4.2に示した回路では、

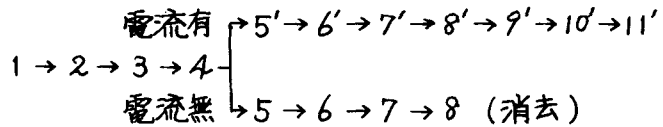
$$C_r C_{r-1} \dots C_1 \text{ (r桁2進表示)} = i-1$$

となるように各 C_j ($1 \leq j \leq r$) の値 (0 または 1) を選んだとき左から第 i 番目の情報のみが検出回路まで伝達するように2種類のスイッチを配置している。ただし、 $C_j = 0$ は導体 C_j に電流を流さないことを意味し、 $C_j = 1$ は導体 C_j に電流を流すことを意味する。



バブル転送路

(1オンスイッチ)



(0オンスイッチ)

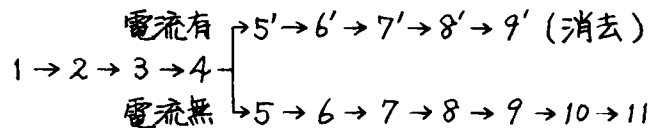


図4.3 スイッチ回路図

以上の原理により、分割回路を連続動作させて導体 C_1, C_2, \dots, C_r に流す電流の有無をシーケンシャルに適当に変化させることにより、変換した結果を検出回路から時系列として得ることができる。

(例) 情報系列 00111010 を大きさ $8 (= 2^3)$ の変換回路で処理するとき、導体 C_1, C_2, C_3 に流す電流の有無をつぎのように変化させると下記の実出力系列が得られる。

時刻	1	2	3	4	5	6	7	8	9	10	
C_1	1	0	1	1	0	1	0	0	-	-	
C_2	-	1	0	0	1	0	0	1	1	-	
C_3	-	-	0	1	0	1	0	1	1	0	
位置				4	5	2	8	1	6	7	3
出力系列				1	1	0	0	0	0	1	1

ただし、時刻は1ビット転送するのに要する時間、すなわち、回転駆動磁界の周期 T_R を単位として表わしている。また、位置はもとの情報系列において左端から数えたものである。(例終)

装置は、大きさ $n \times n$ の二次元シフトアレイと n 個の変換回路(大きさ n) を図4.4に示すように接続した構成をもつ。

装置の処理方式はつぎのとおりである。まず、二次元シフトアレイに記憶されている $n \times n$ 二次元アレイの n 本の列アレイ(長さ n) を縦方向シフトモードのもとで二次元シフトアレイの下端より読出し、各列アレイを変換回路に書込む。

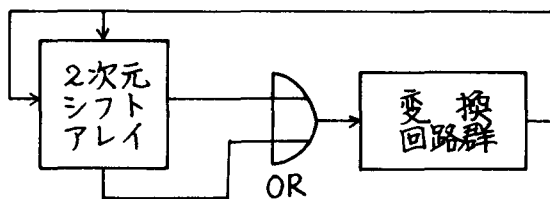


図4.4 装置の構成

つぎに、変換回路により処理を行ない、その処理結果の時系列を二次元シフトアレイの上端より書込む。これを列変換という。同様に、横方向シフトモードのもとで n 本の行アレイ(長さ n) を変換回路群で並列処理するのを行変換という。以上の列変換と行変換とを交互に繰返して二次元的な位置変換を行なうことができる。これが、ここで述べる装置の処理方式である。

なお、 $n_1 \times n_2$ 二次元アレイ ($n_1, n_2 \leq n$) の場合には、このアレイを $n \times n$ 二次元アレイに埋込んで処理することができる。

4.2.3 2次元変換アレイの行・列分解

4.2.2節で述べたように、列変換と行変換とを交互に繰返して位置変換を行なうため、2次元的位置変換を列変換と行変換に分解する必要がある。この節では、その分解方法について述べる。

列変換と行変換を数学的に記述するため、1次元変換アレイという概念を導入する。

〔定義4.2〕 1次元変換アレイ T^n はつぎに示すような長さ n の1次元アレイの集合 V^n から V^n への写像である。

$$T^n = (t_1, t_2, \dots, t_n) : V^n \rightarrow V^n$$

$$t_i = j, \quad 1 \leq i, j \leq n.$$

$$u = (u_1, u_2, \dots, u_n) \in V^n \quad \text{としたとき、}$$

$$T^n(u) = (v_1, v_2, \dots, v_n) \in V^n, \quad v_i = u_j$$

である。

(定義終)

列変換、行変換は n 組の1次元変換アレイで表わされ、それぞれ同一列アレイ、同一行アレイ内における情報の位置変換である。

〔定義4.3〕 2次元変換アレイ中に現われる (i, j) の個数を (i, j) の重複度といい、 $m_{i,j}$ で表わす。また、 x の個数を m_x で表わす。

(定義終)

〔定義4.4〕 $m_x = 0$ のとき、その2次元変換アレイは完全であるといい、 $m_x \neq 0$ のとき不完全であるという。

(定義終)

2次元変換アレイが不完全であるとき、その中の x を m_x 個の $m_{i,j} = 0$ である (i, j) で置きかえることにより完全な2次元変換アレイにすることが常に可能である。というのは、 $m_{i,j} = 0$ である (i, j) が少なくとも m_x 個存在するからである。そこで、以下では完全な2次元変換アレイについて考察することにする。

(I) すべての $m_{i,j}$ が1である2次元変換アレイ

このような2次元変換アレイを2次元置換アレイという。2次元置換アレイにより表わされる位置変換は置換であり、よく知られているように〔57〕、列置換・行置換・列置換により実現される*。置換の列置換・行置換・列置換への分

* $\forall 1 \leq i, j \leq n, t_i = t_j \Leftrightarrow i = j$ である1次元変換アレイを1次元置換アレイといい、 n 組の1次元置換アレイにより表わされる変換を列置換または行置換という。

解は、例えば文献[57]のアルゴリズムにより行なうことができる。

(例) 前述の例に示した位置変換は置換であり、図4.5のように分解される。

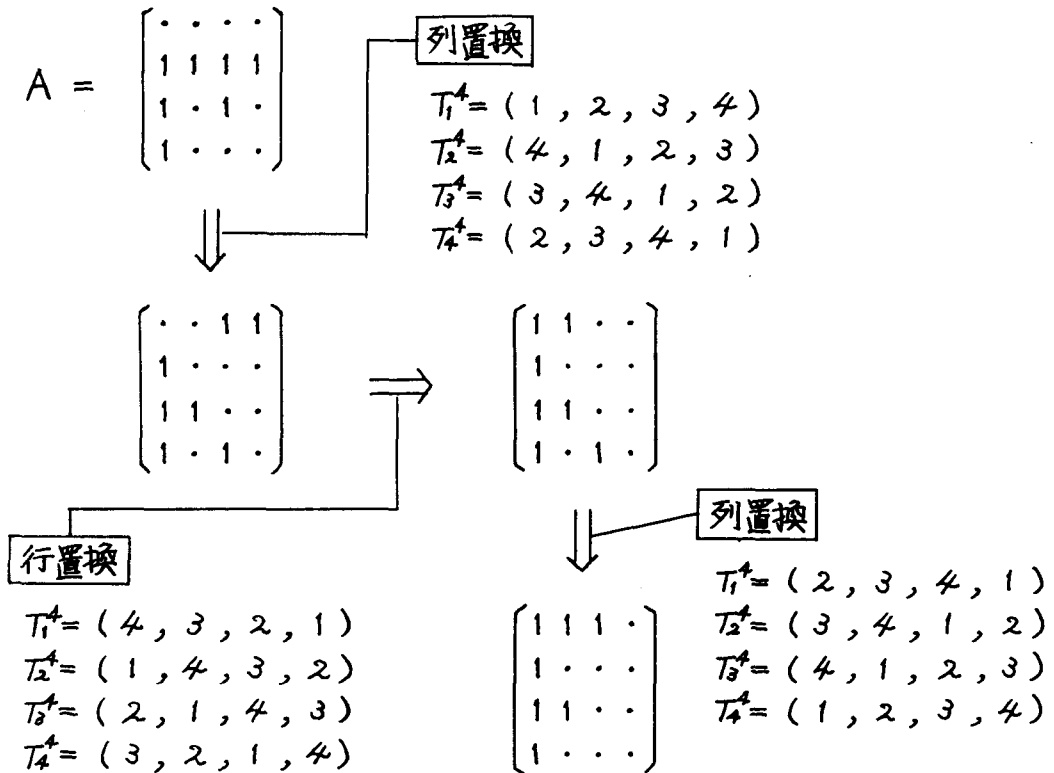


図4.5 二次元置換アレイの行・列分解

(例終)

(II) $m_{i,j} \geq 2$ である (i, j) が存在する二次元変換アレイ

このような二次元変換アレイにより表わされる位置変換は、まず各 (i, j) 成分を $m_{i,j}$ 個だけ複写し、つぎにそれらを置換して実現することができる。後述するように複写は列置換・行置換・列置換により実現できる。また、置換は(I)でも述べたように列置換・行置換・列置換により実現できる。これを続けて行なうと複写の最後の列置換と置換の最初の列置換を或る一つの列置換にまとめることができる。結局、この位置変換は列置換・行置換・列置換・行置換・列置換により実現されることになる。

各 (i, j) 成分を $m_{i,j}$ 個だけ複写する方法を以下に述べる。

つぎのアルゴリズムにより計算される $M_i (1 \leq i \leq n)$ を第 i 行の充足度という。

充足度を求めるアルゴリズム

ステップ 1 $M_i \leftarrow m_{i,1} \quad (1 \leq i \leq n)$

$j \leftarrow 1$

ステップ 2 $j \geq n$ ならば止まれ。

ステップ 3 $j \leftarrow j + 1$

ステップ 4 $m_{i,j} \quad (1 \leq i \leq n)$ を大きい順に並べる。

いま、 $m_{k_1,j} \geq m_{k_2,j} \geq \dots \geq m_{k_n,j}$ であるとする。

ステップ 5 $M_i \quad (1 \leq i \leq n)$ を小さい順に並べる。

いま、 $M_{i_1} \leq M_{i_2} \leq \dots \leq M_{i_n}$ であるとする。

ステップ 6 $M_{i_l} \leftarrow M_{i_l} + m_{k_l,j} \quad (1 \leq l \leq n)$

ステップ 2 へ進む。

このとき、つぎの定理が成立する。

[定理 4.1] 上述のアルゴリズムにより充足度を計算するとき、ステップ 2 の実行直前において $M_p \geq M_g$ ならば、

$$\max_{1 \leq s \leq j} \{ m_{p_s, s} \} \geq M_p - M_g$$

が成立する。ただし、その時点において

$$M_p = \sum_{s=1}^j m_{p_s, s}, \quad M_g = \sum_{s=1}^j m_{g_s, s}$$

であるとする。

(証明) j に関する帰納法により証明する。

$j = 1$ のとき、 $M_p = m_{p,1}$ 、 $M_g = m_{g,1} (\geq 0)$ であるから、

$$\max \{ m_{p,1} \} = m_{p,1} \geq m_{p,1} - m_{g,1} = M_p - M_g$$

となり、確かに定理は成立している。

$j = r$ のとき、定理が成立していると仮定する。

$j = r + 1$ のとき、

$$M_p = \sum_{s=1}^{r+1} m_{p_s, s} = M_p' + m_{p_{r+1}, r+1}$$

$$M_g = M'_g + m_{g_{r+1}, r+1}$$

とする。

(i) $M'_p \geq M'_g$ のとき

$m_{p_{r+1}, r+1} \leq m_{g_{r+1}, r+1}$ であるから、

$$\begin{aligned} M_p - M_g &= (M'_p - M'_g) - (m_{g_{r+1}, r+1} - m_{p_{r+1}, r+1}) \\ &\leq M'_p - M'_g \end{aligned}$$

となる。一方、帰納法の仮定により、

$$\max_{1 \leq s \leq r} \{m_{p_s, s}\} \geq M'_p - M'_g$$

である。したがって、

$$M_p - M_g \leq \max_{1 \leq s \leq r} \{m_{p_s, s}\} \leq \max_{1 \leq s \leq r+1} \{m_{p_s, s}\}$$

となり、定理が成立する。

(ii) $M'_p \leq M'_g$ のとき

$m_{p_{r+1}, r+1} \geq m_{g_{r+1}, r+1} \geq 0$ であるから、

$$\begin{aligned} M_p - M_g &= m_{p_{r+1}, r+1} - m_{g_{r+1}, r+1} - (M'_g - M'_p) \\ &\leq m_{p_{r+1}, r+1} \leq \max_{1 \leq s \leq r+1} \{m_{p_s, s}\} \end{aligned}$$

となり、定理が成立する。

(証明終)

[系] 充足度 M_p, M_g に関して、 $M_p \geq M_g$ ならば、

$$\max_{1 \leq s \leq n} \{m_{p_s, s}\} \geq M_p - M_g$$

が成立する。ただし、 $M_p = \sum_{s=1}^n m_{p_s, s}$ とする。

(系終)

各行の充足度を $M_i = \sum_{j=1}^n m_{i,j}$ ($1 \leq i \leq n$) としたとき、複写はつぎのようにして実現できる。

まず、つぎの n 組の 1 次元変換アレイで表わされる列置換を行なう。

$$T_j^n = (1_j, 2_j, \dots, n_j), \quad (1 \leq j \leq n)$$

続いて、つぎの n 組の 1 次元変換アレイで表わされる行変換を行なう。

$M_p > n$ である行 p に関しては、

$$T_p^n = (\underbrace{1 \cdots 1}_{m_{p,1} \text{個}} \underbrace{2 \cdots 2}_{m_{p,2} \text{個}} \cdots \underbrace{n \cdots n}_{m_{p,n} \text{個}} \underbrace{x_p \cdots x_p}_{(m_{dp, x_p} - M_p + n) \text{個}})$$

を施す。ただし、 $m_{dp, x_p} = \max_{1 \leq j \leq n} \{m_{p_j, j}\}$ であるとする。また、 T_p^n の最初の $(M_p - m_{dp, x_p})$ 個の要素には x_p は含まれない。
 $M_g \leq n$ である行 g に関しては、

$$T_g^n = (\underbrace{1 \cdots 1}_{m_{g,1} \text{個}} \underbrace{2 \cdots 2}_{m_{g,2} \text{個}} \cdots \underbrace{n \cdots n}_{m_{g,n} \text{個}} \underbrace{1 \cdots 1}_{(n - M_g) \text{個}})$$

を施す。

最後に、 $M_g < n$ である行の右端 $(n - M_g)$ 個の位置に、 $M_p > n$ である行の超過分 $(M_p - n)$ 個の (α_p, x_p) 成分を埋め込むことを列変換により行なう。系により、

$$(m_{dp, x_p} - M_p + n) - (n - M_g) = m_{dp, x_p} - (M_p - M_g) \geq 0$$

であるから、任意の埋め込みが列変換により可能である。例えば、つぎのようにして埋め込むことができる。充足度が n よりも小さい行を $g_1 \leq g_2 \leq \cdots \leq g_i$ とし、充足度が n よりも大きい行を $p_1 \leq p_2 \leq \cdots \leq p_j$ とする。行 g_1 の右端 $(n - M_{g_1})$ 個の位置、行 g_2 の右端 $(n - M_{g_2})$ 個の位置、 \cdots 、行 g_i の右端 $(n - M_{g_i})$ 個の位置を順につないだリストを Q とする。同様に、行 p_1 の超過分 $(M_{p_1} - n)$ 個の (α_{p_1}, x_{p_1}) 成分、行 p_2 の超過分 $(M_{p_2} - n)$ 個の (α_{p_2}, x_{p_2}) 成分、 \cdots 、行 p_j の超過分 $(M_{p_j} - n)$ 個の (α_{p_j}, x_{p_j}) 成分を順につないだリストを P とする。リスト P とリスト Q とは同じ長さである*。そこで、二つのリストの同じ位置同志を 1 対 1 対応させて超過成分の埋め込み位置を決める。この埋め込みを実現する列変換を表わす n 組の 1 次元変換アレイはつぎのようにして求められる。上述の埋め込みにおいて、位置 (i, j) に超過成分 (α_p, x_p) を埋め込む場合には第 j 列に対応する 1 次元変換アレイ T_j^n において $c_i = p$ とし、それ以外の場合には $c_i = i$ とする。以上のようにして複写を行なうことができる。

(例) 2次元変換アレイ

$$T^{4 \times 4} = \begin{pmatrix} (2, 2) & (1, 1) & (1, 1) & (4, 1) \\ (3, 2) & (1, 1) & (4, 1) & (1, 3) \\ (1, 4) & (2, 2) & (3, 1) & (1, 3) \\ (2, 1) & (1, 2) & (1, 2) & (3, 1) \end{pmatrix}$$

* 完全な 2次元変換アレイを考えているから。

で表わされる位置変換を考える。この二次元変換アレイの重複度は、

$$m = \begin{bmatrix} 3 & 2 & 2 & 1 \\ 1 & 2 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{bmatrix}$$

である。前述のアルゴリズムにより各行の充足度を求めると、

$$M_1 = m_{1,1} + m_{4,2} + m_{1,3} + m_{2,4} = 3 + 0 + 2 + 0 = 5$$

$$M_2 = m_{2,1} + m_{1,2} + m_{2,3} + m_{1,4} = 1 + 2 + 0 + 1 = 4$$

$$M_3 = m_{3,1} + m_{2,2} + m_{3,3} + m_{3,4} = 2 + 2 + 0 + 0 = 4$$

$$M_4 = m_{4,1} + m_{3,2} + m_{4,3} + m_{4,4} = 2 + 1 + 0 + 0 = 3$$

となる。1行目が超過行、4行目が不足行であり、

$$\max \{ m_{1,1}, m_{4,2}, m_{1,3}, m_{2,4} \} = m_{1,1}$$

であるから、超過成分(1, 1)を位置(4, 4)に埋め込むことになる。分解した結果を図4.6に示す。

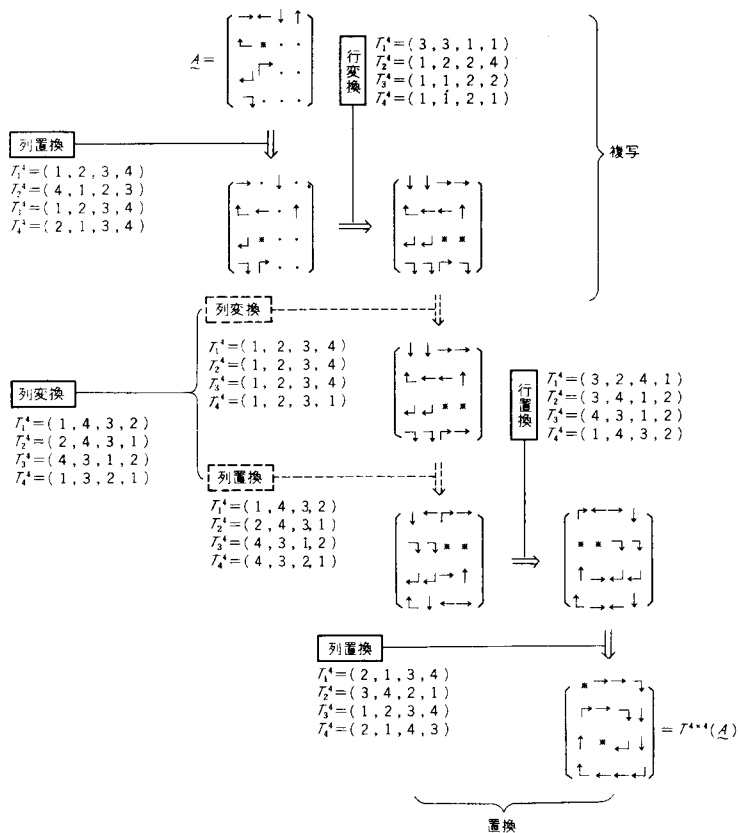


図4.6 二次元変換アレイの行・列分解

(例終)

4.3 ソーティングメモリ

磁気バブル素子は記憶機能のほかに論理機能を有しており、チップ上で論理演算を行なうことが可能なことから、情報を外部に読出さずにメモリ内部で或る種の情報処理を行なう機能メモリの構成は興味深い。また、磁気バブル素子の二次元性と遠隔制御性は並列処理装置の構成を可能にしている。この節では並列情報処理機能をもつ記憶装置の例として二次元シフトレジスタ機能を利用したソーティングメモリの構成例を述べ、磁気バブル応用の新しい可能性を示す。

ここで述べる装置は図4.7に示すように記憶部、論理部および交換部から成り、語をキーワードの大きさの順に並べかえるという機能をもつ。語は図4.8に示すようにキーワード部とデータ部とから成り、装置内では1、0をバブルの有無に対応させて記憶している。

回路の動作を装置の処理方式と対応させて以下に示す。

(i) 書込み

まず、処理すべき語を記憶部の上辺にある書込み/読出し回路から装置に書込み、記憶部に記憶する。記憶部は二次元シフトレジスタでできており、記憶した情報を上下左右々方向にシフトする機能をもつ。書込みは1語ずつ下方向にシフトしながら行なわれ、その操作はビット並列、語直列である。なお、書込みを開始する前に回路中の不要なバブルはすべて消去しているものとする*。

(ii) 比較準備

処理すべき語の書込みが終了すると、つぎの(iii)で行なうキーワードの大小比較の準備のために、右シフトにより記憶部の語を交換部に移す。記憶部から交換部に至るバブル転送回路を図4.9に示す。語を交換部に移す際、分岐点Aおよび分岐点Cでは外側の転送路を通す。ここで行なう操作はビット直列、語並列である。なお、装置は図4.9に示した単位回路を $n/2$ 個繰返し並べて構成される。ただし、装置の記憶容量を n 、あるいは $n+1$ とする。

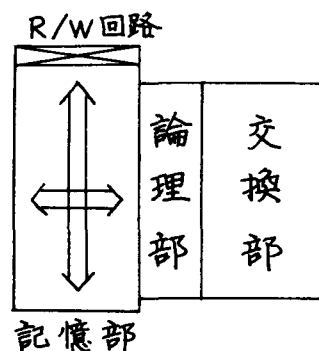


図4.7 装置の構成

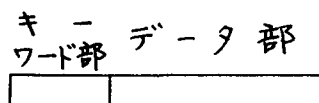


図4.8 語の構成

* 後述する論理部の制御用バブル以外のバブルは不要である。

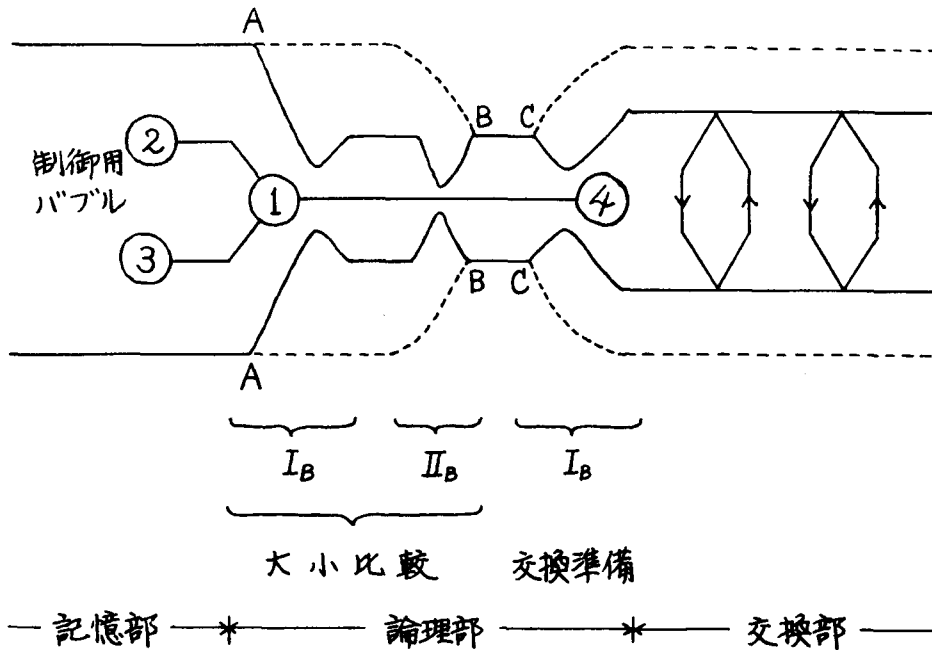


図4.9 装置の単位回路

(iii) 比較

隣りあった語のキーワードの大きさを比較するため、(ii)で交換部に移した語を左シフトにより論理部を通して再び記憶部にもどす。このとき、キーワード部は分岐点Bで内側の転送路を通し、データ部は外側の転送路を通してもどす。キーワード部を分岐点Bで内側の転送路を通そうとするとき、隣りあった語のキーワードの対応するビットが一致して1である場合*、すなわち、両方にバブルがある場合には互いに反発しあって分岐点Bで内側の転送路を通ることができず、外側の転送路を通ってもどることになる。したがって、バブルが内側の転送路を通ってもどるのは、隣りあった語のキーワードの対応するビットが異なるときである。ここで利用した論理演算は第3章で述べた II_B である。

一方、論理部には一つの制御用バブルが常駐し、①~④のいずれかの位置に存在している。比較を行なうときには、①に制御用バブルを置いておく。分岐点Bからキーワード部の一一致しなかったビットに対応するバブルが内側の転送路を通ってくると、そのバブルにより①の制御用バブルは反発されて②あるいは③に移動し、そこにとどまる。すなわち、制御用バブルは最初に下側の転送

* 一致して1である場合には、どちらにもバブルが存在しないから、どちらの転送路を通して情報が転送されても同じである。

路にバブルが現れた場合には②に移動し、最初に上側の転送路にバブルが現れた場合には③に移動する。ここで利用した論理演算は I_B である。語を交換部から記憶部にもどし終わったとき、上にある語のキーワードの方が下にある語のキーワードよりも小さい場合には制御用バブルは②に、大きい場合には③に、等しい場合には①にそれぞれ存在することになる*。以上で、隣りあった語のキーワードの大小比較が完了する。この操作もビット直列、語並列である。

(iv) 交換準備

(iii)の比較結果に基づき、語の順序をかえる準備をする。交換が必要なのは上にある語のキーワードの方が下にある語のキーワードよりも小さい場合、すなわち、操作(iii)を終了したとき制御用バブルが②にある場合である。キーワードが等しく制御用バブルが①にある場合には交換してもしなくてもよいが、ここではバブルの反発回数をできるだけ少なくするために交換を行なうことにする。

そこでまず、①のバブルを②に移し、③のバブルを④に移した後、右シフトにより記憶部の語を論理部を通して交換部に移す。このとき、分岐点Aでは外側の転送路を、分岐点Cでは内側の転送路を通す。分岐点Cで内側の転送路を通そうとすると、④に制御用バブルがあるとそれに反発されて内側の転送路に入ることができず、外側の転送路に入ってしまう。すなわち、交換が必要でない場合には語は外側の転送路に入り、交換が必要な場合には内側の転送路に入る。ここで利用した論理演算は I_B である。以上で交換の準備が完了する。この操作もビット直列、語並列である。

(v) 交換

交換が必要な場合に、隣りあった上下の語を入れかえる。交換部の内側の転送路上にある語を、図々、9の六角形で示した巡回転送路(ループ)を用いて上の語は左側を下方向に、下の語は右側を上方向にシフトして入れかえる。交換が不要であれば、語は外側の転送路上にあり、この操作を施しても語の入れかえは起こらない。この操作はビット並列、語並列である。

つぎに、左シフトにより交換した結果を交換部から記憶にもどす。このとき、制御用バブルを①に移し、分岐点Bでは外側の転送路を通す。この操作はビット直列、語並列である。以上で隣りあった2語ずつの比較、交換が完了したことになる。

(vi) 反復処理準備

この装置で実現しているソーティングアルゴリズムは隣りあった二つの語の比較、交換を繰返し行なってソートする、いわゆるバブルソートである。そこ

* 1、0をバブルの有無に対応させているから。

でつぎに、他の側の隣接語との比較、交換を行なうための準備をする。これは記憶部の語を上方向あるいは下方向に1語だけシフトして、つぎに比較される語の組合わせをかえるだけでよい。この操作はビット並列、語並列である。

(ii) ~ (vi) の操作を $(m-1)$ 回繰返すことによりソーティングが完了する。ただし、 m 語のソーティングを考えている。

(vii) 読出し

ソートされた結果を読出す場合には、1語ずつ上方向にシフトしながら読出し回路からキーワードの大きい順に読出すことができる。この操作はビット並列、語直列である。

この装置では記憶部と交換部との間を又往復すれば1回の比較、交換が行なわれ、これを $(m-1)$ 回繰返してソーティングが完了するので、その処理時間は*、およそ $(m-1) \times T_R$ である。ただし、語の長さを l ビット、処理する語の数を m 、1ビットシフトに要する時間を T_R とする。

4.4 ループ結合形機能メモリ

この節では、磁気バブルチップ上に計算システム等を構成する際に有効なループ結合形機能メモリについて述べる。

ループ結合形機能メモリは図4.10に示すようにデータを記憶する巡回シフトレジスタ(ループ)と二つのループの結合部に設けた論理部とから成る。データが論理部を通過してループ上をシフトされている間に、結合されたもう一方のループ上のデータとの間で論理演算が行なわれ、その結果が新しくループ上に得られる仕組みである。データに何ら処理を施さず、入力をそのまま出力する機能を論理部に与えておくことにより、通常はそのモードで論理部を動作させてデータをそれぞれのループに保持しておくことができる。そして、必要に応じて演算を行なうモードに切換えられるように論理部を構成しておく。こうすることにより、チップ上の或る場所で演算を行なっているときに、演算を行わない他の場所のバブルも一様に駆動することが可能となり駆動機構が簡単になる。第2章で述べたセルループ方式バブルメモリも一種のループ結合形機能メモリである。この場合には、結合部に導体に流れる電流で制御するゲートを設置し、ゲート

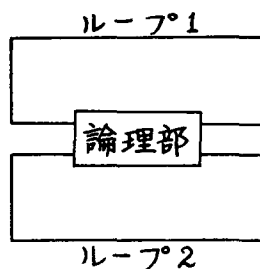


図4.10 ループ結合形機能メモリ

*、ソーティングを行なうのに要する時間。

の開閉によってデータの保持と、ループ間でのデータの交換を行なっている。

ここでは、バブルにより論理部のモードを制御するループ結合形機能メモリの例として加算メモリの構成例を示す。

図4.11(a)は長さ $l (= m+n)$ のループを二つ交叉結合し、交叉点に図4.11(b)に示す論理部を設けて構成した加算メモリである。論理部の機能は制御入力 a により切換えることができる。すなわち、 a にバブルを供給しているときには出力として、

$$\hat{x} = x, \quad \hat{y} = y$$

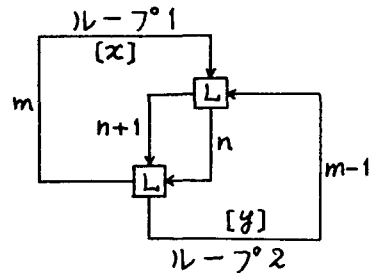
が得られ、論理部は単なる交叉として動作しデータはそれぞれのループにそのまま保持される。一方、 a へのバブルの供給をとめると出力として、

$$\hat{x} = x \oplus y, \quad \hat{y} = x \oplus y$$

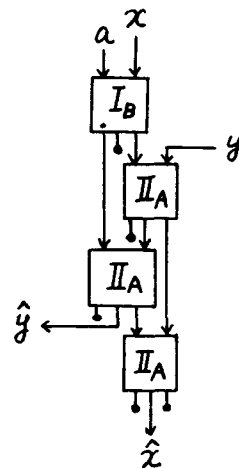
が得られ、論理部は半加算器として動作する。また、二つの論理部間の距離は図4.11(a)に示すようにループ1とループ2とで1ビットだけ違っているため、論理部で演算されるビットの組が1ビットずつずれていく。このずれにより桁上げが行なわれ、ループ1に加算結果 $[x] + [y]$ が、ループ2に0がそれぞれ得られる。なお、加算に必要な時間はおよそ $(l^2/2) T_R$ である。ただし、 T_R は1ビットシフトするのに要する時間である。

図4.12も加算メモリの一例である。制御入力 a にバブルを供給しないときにはループ1のデータとループ2のデータとは接触しないで、それぞれのループに保持される。一方、 a にバブルを供給するとループ2のデータがループ1の方に行き、そこでループ1のデータとの演算が行なわれる。桁上げを行なう原理は図4.11の場合と同様である。演算の結果、ループ1に加算結果が、ループ2に0が得られる。なお、加算時間はおよそ $l^2 T_R$ である。

図4.13はループ1とループ2の接触点を図4.12の場合の l 倍に増加した



(a) ループ交叉結合



$$\begin{cases} \hat{x} = ax \oplus xy \oplus axy \\ \hat{y} = y \oplus x \oplus ax \end{cases}$$

(b) 論理部

図4.11 加算メモリ(1)

もので、加算時間は $1/k$ に短縮される。

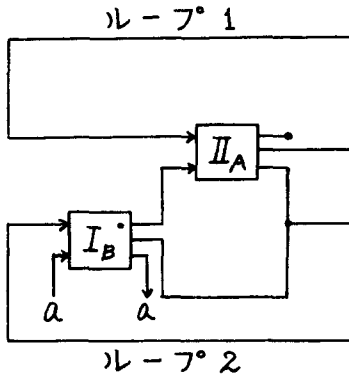


図4.12 加算メモリ(2)

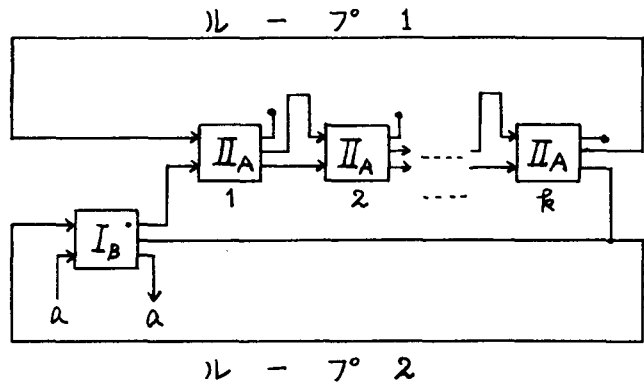


図4.13 加算メモリ(3)

4.5 結言

この章では、磁気バブル素子の記憶機能以外の特性の応用について論じた。とくに、論理機能、二次元性、遠隔制御性も考慮した応用装置の構成例をいくつか述べ、磁気バブル応用の新しい方向を示した。

4.2節のスイッチング機能をもつ二次元バブルメモリでは、磁気バブル素子の二次元性と遠隔制御性（一様制御性）を考慮して、二次元シフトアレイを利用した二次元アレイ情報処理装置の構成を示し、その際の処理方式について論じた。磁気バブル素子はファラデー効果を有し、ディスプレイ装置への応用が考えられているが、その際、ここで述べた二次元シフトアレイを用いれば図形処理機能をもつディスプレイ装置が構成できる。

4.3節のソーティングメモリでは、並列処理が可能な情報処理の代表的なものとしてソーティングを選び、これを二次元シフトレジスタ機能を利用した並列処理装置に実現した。ここで示した構成法による並列処理装置は磁気バブル素子の諸特性に適しており、今後、最も有望な磁気バブル応用装置であると思われる。なお、装置に実現する情報処理としてはソーティングや探索などのように或る簡単な演算の繰返しアルゴリズムによるものが適している。

4.4節のループ結合形機能メモリは、チップ上に計算システム等を構成する場合のように、異種の機能モジュールがチップ上に点在する場合にもチップ上の全バブルを一様に駆動することができるという点で有効な構成法である。

第5章 結論

本研究で得られたおもな成果と今後に残された問題をつぎにまとめる。

第2章では、バブルメモリのアクセス時間を記憶方式の観点から考察し、ダイナミックメモリの構造を応用してバブル転送路にバイパスを設けることにより、従来の方式よりも低価格で短いアクセス時間を達成できるセルループ方式という新しい記憶方式を提案した。ここで述べた方式では制御のための周辺回路をできるだけ簡単にするためにループ結合部のゲートを一様に制御しているが、メモリリクエストの番地発生傾向によってゲートの制御を工夫することにより実質的なアクセス時間の短縮が可能である。これは、いわゆるダイナミックなデータの再配置である。セルループ方式におけるダイナミックなデータの再配置は今後に残された問題である。

第3章では、バブル発生器の個数に着目して、論理関数の実現を考察し、磁気バブル論理系の性質を明らかにした。つぎに、完全性に関する興味深い結果を一般の論理関数論の結果などに対応させて示しておく。

$\{I_A, G^k\}$ は完全系ではない。 \leftrightarrow $\{\cdot, \vee, 1\}$ で実現される関数は、
ただし、 k は正整数である。 単調関数である。

$\{I_B, G^2\}, \{II_B, G^2\}$ は完全系 \leftrightarrow $\{p, 1\}$ は完全系をなす。ただし、
をなす。 p は二項演算で、 $p(x, y) = x\bar{y}$ である。

$\{II_A, G\}$ は完全系をなす。 \leftrightarrow $\{\oplus, \cdot, 1\}$ は完全系をなす。
 \leftrightarrow 万能論理回路である半加算器アレイ
は定数1を2個必要とする。ただし、
 n は論理関数の変数の数である。

$\{I_A, I_B, G\}$ は完全系をなす。 \leftrightarrow $\{I_A, G\}, \{I_B, G\}$ はどちらも完
全系ではない。

実際に磁気バブル論理回路を構成する際の問題である、最少個数の基本論理セルによる論理関数の実現問題、および、遅延時間を考慮した場合の最少段数による論理関数の実現問題は今後に残された問題である。また、分岐が禁止されている場合の完全系が満たす必要十分条件を求める問題は論理関数論の立場から興味深い。

第4章では、磁気バブル素子の記憶機能、論理機能、二次元性、遠隔制御性を考慮した、新しい応用装置の構成例を示した。とくに二次元シフトレジスタを利用した並列処理装置は磁気バブル素子の諸特性に適しており、今後、最も有望な応用装置であると思われる。ソーティング以外の並列処理が可能な情報処理について、それをここで示した構成法で装置に実現してみることは今後

残された問題である。なお、実現する情報処理としては、単純な演算を繰返すアルゴリズムによるものが適している。

また、磁気バブル素子はここで述べた特性以外にも応用上興味深い特性もっており、たとえば、光や温度との相互作用を利用した装置や、渦状磁区をそのまま利用したアナログ装置などへの応用も興味深い。

謝 辞

本研究の全過程を通じて、直接理解ある御指導を賜わり、つねに励ましていただいた尾崎弘教授、樹下行三助教授ならびに白川功助教授に衷心より感謝の意を表する。

大学院修士、博士両課程において電子工学一般および各専門分野に関し御指導、御教示を賜わった電子工学教室中井順吉教授、小山次郎教授、見玉慎三教授、電子ビーム研究施設裏克己教授、堀輝雄教授、産業科学研究所松尾幸人教授、中村勝吾教授、角所収教授ならびに喜田村善一名誉教授に深謝する。

磁気バブルに関する有益な御助言および御援助をいただいた富士通株式会社伝送事業部川島将男部長はじめ関係各位の方々に深く感謝する。

本研究に関し、福井大学谷口慶治助教授、名古屋工業大学山本勝助教授、基礎工学部細見輝政助手、基礎工学部柏原敏伸助手、富士通株式会社伝送事業部塩浜二郎氏には本学大学院在学中に有益な御助言、御討論をいただき、心から謝意を表する。

また、琉球大学喜屋武盛基教授ならびに佐賀大学高松雄三講師には、いろいろ御助言、御援助をいただき、厚く御礼申し上げる。

筆者の属している尾崎研究室の藤原秀雄助手、河田亨助手、戸松重一技官、大学院学生川端信賢氏、坂本明雄氏、笹尾勤氏、千葉徹氏、築山修治氏、トランディン・アム氏、また同研究室の藤田基美子氏には種々の面で御協力いただいた、ここに記して感謝する次第である。

参 考 文 献

- [1] A.H.BOBECK, H.E.D.SCOVIL, "MAGNETIC BUBBLES", SCIENTIFIC AMERICAN, 224, 6 (1971).
- [2] A.H.BOBECK, "PROPERTIES AND DEVICE APPLICATIONS OF MAGNETIC DOMAINS IN ORTHOFERRITES", B.S.T.J., 46, 8 (1967).
- [3] A.H.BOBECK, F.P.FISCHER, A.J.PERNESKI, J.P.REMEIKA, L.G.VAN UITERT, "APPLICATION OF ORTHOFERRITES TO DOMAIN WALL DEVICES", IEEE TRANS., MAG-5, 3 (SEPT. 1969).
- [4] A.H.BOBECK, "A SECOND LOOK AT MAGNETIC BUBBLES", IEEE TRANS., MAG-6, 3 (SEPT. 1970).
- [5] A.H.BOBECK, "MAGNETIC DOMAIN DEVICES: A TUTORIAL", IEEE TRANS., MAG-7, 3 (SEPT. 1971).
- [6] A.H.BOBECK, R.F.FISCHER, A.J.PERNESKI, "A NEW APPROACH TO MEMORY AND LOGIC CYLINDRICAL DOMAIN DEVICES", AFIPS CONF. PROC., 35 (1969).
- [7] H.CHANG, "CAPABILITIES OF THE BUBBLE TECHNOLOGY", NATIONAL COMPUTER CONF., (1974).
- [8] 古尾谷, "磁気バブル素子", 情報処理, 13, 10 (Oct. 1972).
- [9] 桜井, "磁気バブル概論", システムと制御, 18, 9 (1974).
- [10] A.J.PERNESKI, "PROPAGATION OF CYLINDRICAL MAGNETIC DOMAINS IN ORTHOFERRITES", IEEE TRANS., MAG-5, 3 (SEPT. 1969).
- [11] J.L.SMITH, P.I.BONYHARD, "DOLLAR-SIGN TRANSFER FOR MAGNETIC BUBBLES", IEEE TRANS., MAG-9, 3 (SEPT. 1973).
- [12] J.A.COPELAND, "CONDUCTOR PROPAGATION CIRCUITS FOR HIGH DENSITY BUBBLE DOMAIN MEMORIES", INTERMAG. CONF. (1972).
- [13] N.HAYASHI, L.T.ROMANKIW, H.CHANG, S.KRONGELB, "FABRICATION AND OPERATION OF INTENDED ANGELFISH BUBBLE-DOMAIN SHIFT REGISTER", IEEE TRANS., INTERMAG. CONF. (SEPT. 1972).
- [14] F.YAMAUCHI, K.YOSHIIMI, S.FUJIWARA, T.FURUOYA, "BUBBLE SWITCH AND CIRCUIT UTILIZING YY OVERLAY", IEEE TRANS., INTERMAG. CONF. (SEPT. 1972).
- [15] 杉田, 鴨下, "磁気バブル回路", システムと制御, 18, 9 (1974).
- [16] 渡辺, "磁気バブル回路の諸問題", 昭和48年度電気四学会連合大会, 論文番号222.
- [17] P.I.BONYHARD, I.DANYLCHUK, D.E.KISH, J.L.SMITH, "APPLICATION OF BUBBLE DEVICES", IEEE TRANS., MAG-6, 3 (SEPT. 1970).
- [18] P.C.MICHAELIS, I.DANYLCHUK, "MAGNETIC BUBBLE REPARTORY DIALER MEMORY", IEEE TRANS., MAG-7, 3 (1971).
- [19] P.I.BONYHARD, J.E.GEUSIC, A.H.BOBECK, Y.CHEN, P.C.MICHAELIS, J.L.SMITH, "MAGNETIC BUBBLE MEMORY CHIP DESIGN", IEEE TRANS., MAG-9, 3 (SEPT. 1973).

- [20] P.C.MICHAELIS, P.I.BONYHARD, "MAGNETIC BUBBLE MASS MEMORY-MODULE DESIGN AND OPERATION", IEEE TRANS., MAG-9, 3 (SEPT. 1973).
- [21] P.C.MICHAELIS, W.J.RICHARDS, "MAGNETIC BUBBLE MASS MEMORY", IEEE TRANS., MAG-11, 1 (JAN. 1975).
- [22] H.CHANG, J.FOX, D.LU, L.L.ROSIER, "A SELF-CONTAINED MAGNET BUBBLE-DOMAIN MEMORY CHIP", IEEE TRANS., MAG-8, 2 (JUNE 1972).
- [23] M.TAKAHASHI, H.NISHIDA, T.KASAI, Y.SUGITA, "FABRICATION OF BUBBLE MEMORY CHIPS", IEEE TRANS., MAG-10, 4 (DEC. 1974).
- [24] W.F.BEASOLEIL, D.T.BROWN, B.E.PHELPS, "MAGNETIC BUBBLE MEMORY ORGANIZATION", IBM J. RES. DEVELOP. 16 (Nov. 1972).
- [25] T.C.CHEN, C.TUNG, "STORAGE MANAGEMENT OPERATIONS IN LINKED UNIFORM SHIFT-REGISTER LOOPS", IBM RES., RJ1576(#23535), APRIL 23, 1975, COMPUTER SCIENCE.
- [26] C.TUNG, T.C.CHEN, "BUBBLE LADDER FOR INFORMATION PROCESSING", IBM Res., RJ1556(#23304), APRIL 21, 1975, COMPUTER SCIENCE.
- [27] 宮永, "磁気バブルメモリ", 電気学会技術報告(II部) (昭50-7).
- [28] 古尾谷, "磁気バブルの応用(1)バブルメモリ", システムと制御, 18, 9 (1974).
- [29] H.S.STONE, "DYNAMIC MEMORIES WITH ENHANCED DATA ACCESS", IEEE TRANS., C-21, 4 (APRIL 1972).
- [30] A.V.AHO, J.D.ULLMAN, "DYNAMIC MEMORIES WITH RAPID RANDOM AND SEQUENTIAL ACCESS", IEEE TRANS., C-23, 3 (MARCH 1974).
- [31] R.M.SANDFORT, E.R.BURKE, "LOGIC FUNCTIONS FOR MAGNETIC BUBBLE DEVICES", IEEE TRANS., MAG-7, 3 (SEPT. 1971).
- [32] Y.S.CHEN, J.E.GEUSIC, J.L.SMITH, "CHARACTERIZATION OF MAGNETIC BUBBLE GENERATORS", IEEE TRANS., MAG-10, 1 (MARCH 1974).
- [33] R.L.GRAHAM, "A MATHEMATICAL STUDY OF A MODEL OF MAGNETIC DOMAIN INTERACTIONS", B.S.T.J., 49, 8 (Oct. 1970).
- [34] A.D.FRIEDMAN, P.R.MENON, "MATHEMATICAL MODELS OF COMPUTATION USING MAGNETIC BUBBLE INTERACTIONS", B.S.T.J., 50, 6 (JULY-AUG. 1971).
- [35] E.YODOKAWA, "ON A MATHEMATICAL MODEL OF MAGNETIC BUBBLE LOGIC", NATIONAL COMPUTER CONF. (1973).
- [36] S.Y.LEE, H.CHANG, "MAGNETIC BUBBLE LOGIC", IEEE TRANS., MAG-10, 4 (DEC. 1974).
- [37] R.C.MINNICK, P.T.BAILEY, R.M.SANDFORT, W.L.SEMON, "CASCADE REALIZATIONS OF MAGNETIC BUBBLE LOGIC USING A SMALL SET OF PRIMITIVES", IEEE TRANS., C-24, 2 (FEB. 1975).
- [38] R.C.MINNICK, P.T.BAILEY, R.M.SANDFORT, W.L.SEMON, "MAGNETIC BUBBLE LOGIC", PROC. 1972 WESCON.

- [39] R.C.MINNICK, "A SYSTEM OF MAGNETIC BUBBLE LOGIC", IEEE TRANS., C-24, 2 (FEB. 1975).
- [40] M.R.GAREY, "RESIDENT-BUBBLE CELLULAR LOGIC USING MAGNETIC DOMAINS", IEEE TRANS., C-21, 4 (APRIL 1972).
- [41] H.CHANG, T.C.CHEN, C.TUNG, "THE REALIZATION OF SYMMETRIC SWITCHING FUNCTIONS USING MAGNETIC BUBBLE TECHNOLOGY", NATIONAL COMPUTER CONF. (1973).
- [42] 棚橋, 高橋, "磁気バブルの論理への応用", 信学会, 電子回路部品・材料研資, CPM71-81 (1972-03).
- [43] 棚橋, 今村, "磁気バブル論理機能回路の検討", 信学会, 電子部品・材料研資, CPM73-34 (1973-06).
- [44] 笹尾, 樹下, "磁気バブル素子による論理回路の最小形", 信学論(D), 57-D, 8 (昭49-08).
- [45] 笹尾, 樹下, "保存形論理素子の万能性について", 信学論(D), 58-D, 11 (昭50-11).
- [46] P.I.BONYHARD, "APPLICATION OF BUBBLE DEVICES IN DIGITAL SYSTEMS", IEEE TRANS., MAG-6, 4 (SEPT. 1970).
- [47] 小林, "磁気バブルの応用(II)", システムと制御, 18, 9 (1974).
- [48] 坪谷, "Magnetic Bubbleの応用", 信学会誌, 53, 12 (1970-12).
- [49] R.C.MINNICK, "MAGNETIC BUBBLE COMPUTER SYSTEMS", Proc. AFIPS CONF., 41, PART II (DEC. 1972).
- [50] P.I.BONYHARD, T.J.NELSON, "DYNAMIC DATA REALLOCATION IN BUBBLE MEMORIES", B.S.T.J., 52, 3 (MARCH 1973).
- [51] G.S.ALMASI, "MAGNETOOPTIC BUBBLE-DOMAIN DEVICES", IEEE TRANS., MAG-7, 3 (SEPT. 1971).
- [52] Y.KITA, F.INOSE, M.KASAI, "TWO-DIMENSIONAL SHIFT ARRAY OF MAGNETIC BUBBLES AND ITS APPLICATION TO PATTERN PROCESSING", IEEE TRANS., INTERMAG. CONF. (SEPT. 1972).
- [53] 小原, 高橋, 村上, "磁気バブルデバイスによる記号列パターンの識別処理", 昭和49年度信学会全国大会, 論文番号1772.
- [54] 市川, 石原, 関, "磁気バブルを用いたリ-距離計算回路", 昭和49年度信学会全国大会, 論文番号269.
- [55] 菅田, 森田, 福留, "磁気バブルによるセル構造オートマトンの同期について", 昭和49年度信学会全国大会, 論文番号1537.
- [56] 白川, 尾西, 松下, 森田, 桜井, "[I] 泡磁区材料としての $Gd \cdot Co$ 非結晶薄膜の作成と特性 [II] 泡磁区を用いた連想記憶", 信学会研資, CPM73-32 (1973-06).
- [57] H.R.RAMANUJAM, "DECOMPOSITION OF PERMUTATION NETWORKS", IEEE TRANS., C-22, 7 (JULY 1973).
- [58] R.S.KRUPP, L.A.TOMKO, "SWITCHING NETWORKS OF PLANAR SHIFTING ARRAYS", B.S.T.J., 52, 6 (JULY-AUG. 1973).

- [59] 竹井, 富永, “シフトレジスタ機能を用いたソーティング”, 信学会, 電子計算機研資, EC72-8 (1972-06).
- [60] 尾崎, 樹下, “デジタル代数学”, 共立出版 (昭41).
- [61] 尾崎, 白川, “グラフとネットワークの理論”, コロナ社 (昭48).