

解説

不揮発性メモリ向け符号化方式

Coding Scheme for Non-volatile Memory

鎌部 浩



フラッシュメモリに代表される不揮発性半導体メモリは、可動部品がない、消費電力が少ない、アクセススピードが高速などの理由で、様々な機器、特に携帯機器での使用が急速に拡大している。ハードディスクなどの記録装置では、誤り訂正符号のほかに、記録に適した系列に変換するために制約符号を利用するが、不揮発性半導体メモリに対しても、寿命を延ばす、読出し時の誤り確率の低減、などを目的とした符号化を利用する方式が提案され活発な研究が行われている。本稿では、そうした符号化の幾つかについて解説する。

キーワード：不揮発性メモリ、フラッシュメモリ、制約符号、相転移メモリ、メモリスタ

1. ま え が き

機械的な可動部分がないため対衝撃性が高く、アクセススピードも速い、などの理由でモバイル機器を中心として、不揮発性メモリが外部記憶装置として広く使用されるようになってきている。ビット当りのコストでは、まだハードディスクドライブ（HDD）の方が有利であるが、アクセス時間の点では、固体素子ドライブ（ソリッドステートドライブ、SSD）の方がかなり有利であるため、現在のSSDでも容量的に十分な機器では、記録装置としてSSDを採用するようになってきている。

また、不揮発性、大容量、超高速アクセスという特徴を持つ半導体メモリを主記憶メモリの延長として捉え、その特性を生かしたシステムの研究も進められている。不揮発性メモリを主記憶に用いると、電源をオフにしてもメモリの内容が消えないため、電源をオンにしたときに前回の作業の環境を簡単に再現できる。更に、数百GByteの膨大なデータをオンメモリで利用することがパーソナルコンピュータでも可能になるため、新しい応用の開発が期待されている。最近製品化された相転移メモリ（Phase Change Memory, PCM, 若しくはPhase Change RAM, PRAM）のアクセススピードは非常に高

速であり、こうした応用が期待されている⁽¹⁾。

ところで、フラッシュメモリには記録素子の物理的な特性から、あるパターンのデータは記録しにくい、ある順序で記録すると素子の寿命が短くなるなどの性質がある。このため記憶装置として利用するためには、工夫が必要になってくる。

フラッシュメモリでは、ウェアレベリングと呼ばれる手法を使ってセルを均一に使用することによって、長寿命化が図られている。近年 A. Jiang らは、これとは別に符号化を利用することで、フラッシュメモリの長寿命化、高密度化が達成できることを示した^{(2), (3)}。これらの技術はまだ実用化されていないが、そのアイデアは、将来のフラッシュメモリの技術に影響を与えられる⁽⁴⁾。最近、多レベルのPCMが開発されたが、それは信頼度を向上させるために、制約符号を利用している⁽⁵⁾。また、2008年にメモリスタ（Memristor）と呼ばれる第4の回路素子が発見され、それをメモリに応用する際に、制約符号を有効に利用する方法が提案されている⁽⁶⁾。

本稿では、固体素子を使った不揮発性メモリのための幾つかの符号化方式について、その概要を述べる。

2. フラッシュメモリのための符号

2.1 フラッシュのための符号の定式化

フラッシュメモリは「セル」と呼ばれる基本的な記憶素子が集まって構成され、以下の性質を持っている。

鎌部 浩 正員：シニア会員 岐阜大学工学部電気電子・情報工学科情報コース
E-mail kamabe@ieee.org
Hiroshi KAMABE, Senior Member (Faculty of Engineering, Gifu University, Gifu-shi, 501-1193 Japan).
電子情報通信学会誌 Vol.96 No.6 pp.417-421 2013年6月
©電子情報通信学会 2013

- ① セルの状態若しくは電荷レベルは増加させることができるが、下げるときには0レベルまで一気に下げることしかできない。
- ② 状態の値には上限がある。
- ③ セルが幾つか集まってブロックを構成しており、ブロック内のあるセルを0レベルにリセットするときには、そのブロック内の全てのセルを同時にリセットしなければならない。このリセット動作をブロック消去と呼ぶ。
- ④ ブロック消去できる回数には上限がある。

フローティング符号 (floating code) は以下のように定義される。

長さが k の情報記号列が、 n 個のセルにひとまとめにして記録される。ここで n と k は正整数で、 $k \leq n$ である。 V を情報記号列の集合、 C をメモリが取り得る状態の空間とする。

定義 1 写像 $D: C \rightarrow V \cup \{\perp\}$ を、フローティング符号と呼ぶ。 $\alpha \in C$ に対して、 $\beta = D(\alpha)$ と置く。もし $\beta \in V$ であれば、セルの状態 α は値 β を表現しているという。もし $\beta = \perp$ であれば、 α は何も表現していないとする。

D は実際には復号を行う写像である。符号化関数 E は、現在のセルの状態とそれが表現しているデータ、そしてそのデータのどの桁をどの値に変更するか、という情報が与えられたときに、そのデータを表現するために次に遷移すべきセルの状態を与える関数である。 E と D の間には、 E の出力を D に入力すると、記録されているべきデータが出力される、という関係が必要である。

フラッシュメモリのための符号化の目的の一つは、ブロック消去の間隔をなるべく大きくすることである。以下ではブロック消去の間隔を t とする。

2.2 WOM 符号

表現できる情報量を犠牲にすると、一度しか記録できない媒体、例えば CD-R や紙テープ、パンチカードなどに、複数回の書込みができるようになる。これを実現するための符号を Write-Once-Memory 符号 (WOM 符号) と呼ぶ⁽⁷⁾。この符号は t を大きくするための符号と考えることもでき、フラッシュメモリへの応用を目的として研究されている。また、最近では誤り訂正能力を持つ WOM 符号も研究されている⁽⁸⁾。

2.3 最悪ケースを改善する符号

各セルが、 q 個の異なる状態を取り得るとすると、 n 個のセルでは q^n 個の状態を表現できる。しかし、フラッシュデバイスでは、ブロック消去を行うまでは、電

表 1 平均性能を改善する復号関数

	0	1	2	3
0	00	01	11	10
1	10	00	01	11
2	11	10	00	01
3	01	11	10	00

荷レベルを上げることしかできないため、表現できる状態の個数は高々 $n(q-1)$ であることが分かる。つまり、 $t \leq n(q-1)$ である。

$n=k$ の場合を考える。情報ビットをそのまま記録する方法では、 $t=q-1$ となることがすぐに分かる。これに対して A. Jiang らは、 $n=k \geq 3$ であれば、 $t=2(q-1)$ となるような符号化が存在することを示している⁽³⁾。

2.4 平均性能を改善する符号

Finucane⁽⁹⁾ らは、 t の平均値を改良する符号を提案している。今、 $n=k=2$ であり、データは2進系列で、セルは四つの値を表現できるとする。表1で定義される復号関数を考える。二つのセルの値が i と j であったとき、この表の i 行 j 列目の要素を返す関数として D を定義する。この表の1行目は、長さが2のグレイ符号になっている。そして、それ以下の行は一つ上の行を右へ一つ巡回置換して得られる。この表は、「各要素の上下左右の記号は1 bit だけ違っている」という性質を持っている。この性質から、復号関数を容易に構成できる。 q が4の倍数であれば、表1をつなぎ合わせることで同様の符号を構成できる。

この符号化方法は、データ系列を利用して二つのセルをランダムに選ぶ方法を与えていると解釈することができる。Finucane らは、この方法によって、ブロック消去の周期が高い確率で $2(q-1)-o(q)$ となることを示している。この方法は $n=8$, $k=4$ の場合にも拡張できる⁽¹⁰⁾。

2.5 Index-less Indexed Flash Code

MahdaviFar らは、Index-less Indexed Flash Code を提案している。これはブロックを幾つかに分割し、次に変更するビットを記憶する方法である⁽¹¹⁾、Suzuki⁽¹²⁾、Kaji⁽¹³⁾ らによって改良した方法が報告されている。

3. Rank Modulation

3.1 Rank Modulation の考え方

フラッシュメモリの記録密度を向上させるために、マルチレベルのフラッシュメモリが開発されている。これによって、一つのセルに記録できるデータを増やすことはできるが、各レベルのスレッシュホールドに合わせて、適

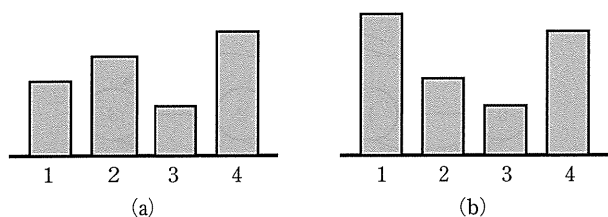


図1 Rank Modulation における「状態」

切に電荷を注入することはセルプログラミングという動作が必要になる。また、経年変化によってスレッショルドレベルが変化すると、誤りを引き起こすことになる。

そこで、セルの電荷レベルの絶対値ではなく、電荷レベルの相対的な大きさの組合せに注目して符号化することが提案されている⁽⁴⁾。

一つのブロックは n 個のセルから成るとする。そのブロックの「状態」はセルが持っている電荷レベルの組合せではなく、電荷レベルの高い順にセルを並べたときのインデックスの並びとする。例えば、図1(a)のように、四つのセルに電荷が注入されていると仮定する。このとき、電荷レベルの高さの順にセルを並べ、その添字だけを並べると (4, 2, 1, 3) となる。これが、Rank Modulation の一つの「状態」である。

Jiang らはある状態から別の状態へ遷移させるための基本的な操作として「push-to-the-top」操作を提案した。これは、指定したセルの電荷レベルを、その他のセルの電荷レベルよりも大きくする操作である。例えば、図1の状態では、セル1の電荷レベルをそれ以外のセルの電荷レベルより高くすると、図1(b)のようになる。そして状態は (1, 4, 2, 3) に遷移する。

3.2 Rank Modulation のためのグレイ符号

S を状態の集合とする。 s_1, s_2, \dots, s_m を、異なる要素のリストとする。このとき、もしある写像 $f: S \rightarrow S$ があって、全ての $1 \leq i \leq m$ に対して、 $s_{i+1} = f(s_i)$ が成立すれば、このリストを「グレイ符号」と呼ぶ。書換えのコストを考えなければ、この符号によって、 $\lceil \log m \rceil$ 桁の情報を記録できる。もし $f(s_m) = s_1$ が成立するのであれば、この符号は「巡回的である」という。

Rank Modulation を考えるときには、 S は $1, 2, \dots, n$ の全ての順列の集合となり、 f は push-to-the-top 操作になる。 n 個のセルから成るブロックに対するこのような符号を、長さ n の Rank Modulation Gray Code と呼び、(n -RMGC) と書く。

3.3 符号化：復号関数と更新関数

グレイ符号を情報の記録に利用する場合、状態と情報系列とを関係付ける関数が必要となる。

定義2 復号関数 (Interpretation function) $\phi: W_n \rightarrow [l]$ は、状態 $s \in W_n$ を $[l]$ の要素に写像する。ここで、 W_n は n 個のセルから成る状態空間 (長さが n の順列の全体) S_n の部分集合であり、また、 $[l] = \{1, 2, \dots, l\}$ と定義する。現在の状態 $s \in W_n$ と新しく記録される情報記号 $i \in [l]$ が与えられたとき、次に遷移すべき状態を決める「更新関数」 $\mu: W_n \times [l] \rightarrow W_n$ とは、 $\phi(\mu(s, i)) = i$ を満たす関数である。

あらゆるデータが表現できるために、 ϕ は上への写像でなければならないが、一対一である必要はない。

3.4 コスト最小の符号

ϕ と μ が与えられると、次のようにデータを符号化できる。現在の状態を s とし、記録すべきデータを $i \in [l]$ とする。必要であれば、2進系列を l 以下の数値に対応させる。 s' を $s' = \mu(s, i)$ とする。グレイ符号の中で s' の位置を求める。 $s' = f^k(s)$ であれば、 s に f を k 回実行して、つまり push-to-the-top 操作を k 回実行して、セルの状態を s' にする。

このようにフラッシュメモリを使用すると仮定すると、問題になるのは push-to-the-top 操作の回数である。この操作を行うことによって、ブロック中のセルの電荷レベルの最大値は必ず k だけ増加するので、この操作の回数を小さくすることが必要となる。

定義3 二つの状態 $s_1, s_2 \in W_n$ に対して、 s_1 から s_2 への状態変化のコストは、 s_1 から s_2 に変化させるために必要な push-to-the-top 操作の最小回数である。

この意味でのコストを最小にする符号として、「Prefix-Free Code」が提案されている⁽⁴⁾。

3.5 Compressed Rank Modulation

push-to-the-top 操作を利用して、 $[4, 1, 3, 2] \rightarrow [1, 4, 2, 3]$ という状態遷移を実現する場合を考える。この遷移は以下のように実現される。

$$[4, 1, 2, 3] \rightarrow [2, 4, 1, 3] \rightarrow [4, 2, 1, 3] \rightarrow [1, 4, 2, 3]$$

もし、電荷レベルが離散的に設定されていたとすれば、3回の遷移操作によって、最大の電荷レベルは3レベル分だけ増加する。この最大電荷レベルの増加分を減らすことができれば、ブロック消去回数が減少することが期待できるので、結果的に寿命を長くできる可能性がある。

最大の電荷レベルを1だけ増加させるだけで (一般にはある定数分だけ) 状態を変化させる符号化を、Compressed Rank Modulation と呼ぶ。詳しい定義は文献

(14)に述べられている。以下では、電荷レベルを1だけ増加させるような状態遷移だけを考える。

3.6 支配集合と符号化

状態空間を S_n としたとき、 s_1, s_2 に対して、 s_1 から s_2 へコスト1で遷移できるとき、 s_1 から s_2 へ向かう有向枝を付加することで、有向グラフを定義できる。このようにしてできたグラフを $G_n(S_n, E_n)$ と書くことにする。ここで、 E_n は上で定義したグラフの枝の集合である。

有向グラフ $G(V, E)$ に対して、ある集合 $D \subset V$ を考える。任意の $v \in V \setminus D$ に対して、ある $u \in D$ が必ず存在して、 v から u に向かう枝が必ず存在するとき、 D を支配集合と呼ぶ。

今、 $G_n(S_n, E_n)$ の状態集合が、素な支配集合 D_1, D_2, \dots, D_l に分解できたと仮定する。関数 $\phi: S_n \rightarrow [l]$ を、 $u \in D_i$ であれば、 i を返す関数とする。これを復号関数とするような更新関数 μ は次のように定義される。 $s \in S_n$ とする。記録すべき値を $k \in [l]$ としたとき、支配集合の定義から、ある要素 $u \in D_k$ があって、 s から u へ向かう枝がある。つまり、上のような遷移で s から u に遷移が可能である。この u に対して $\mu(s, k) = u$ と定義する。このようにすると、符号化率が $R = \frac{1}{n} \log_2 l$ の符号を構成できる。

支配集合 D の要素数については次の不等式が成立する： $|D| \geq n! / \left(\frac{3}{4} 2^{n-1} \right)$ (文献(14), Theorem 4)。

Gad らは、 $n=4$ と $n=5$ の場合に、具体的な支配集合を与えている。しかし、 $n \geq 6$ の場合については、具体的な構成方法は与えられていない。

$n=4$ 及び $n=5$ の場合に、支配集合が満たすべき条件が幾つか分かっている⁽¹⁵⁾。これらの条件は、一般の n に対して支配集合を構成する方向を考察するとき役に立つものと思われる。

4. 相転移メモリ (PCM)

4.1 リセット操作の伝搬

PCM は、結晶相とアモルファス相とで抵抗値が異なることを利用してデータを記録する方法である。相転移させるために加熱するので、高密度化したときに熱のクロストークが生じるが、これを防ぐための制約符号が提案されている⁽¹⁶⁾。多レベルの PCM では、こうした制約符号が本質的な役割を果たすことが報告されている⁽⁵⁾。

例として4レベルのPCMを考える。レベルは、0から3までで、レベルを上昇させる操作をセット操作、0レベルに戻す操作をリセット操作と呼ぶ。リセット操作は実際には高温に加熱することによって、アモルファス相に変化させることに相当する。現在の状態が0でない

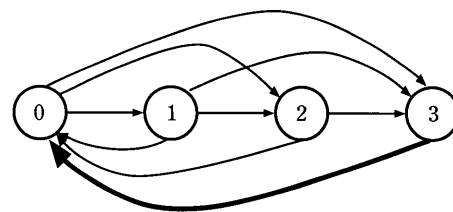


図2 PCMのレベルの変化 (文献(16), Fig. 1)

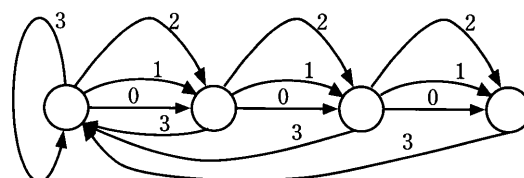


図3 PCMの制約符号の例 (文献(16), Fig. 23)

値であれば、値を上昇させるか、若しくは状態0にリセットすることしかできない。この様子を図2に示す。リセット操作は、セルを高温にすることに対応している。したがって、高い密度でセルが配置されていると、熱が隣のセルの相に影響を与えることが考えられる。

4.2 リセット操作の伝搬を制限する符号

リセット操作の伝搬は、書込みの遅延と消費電力の増加につながるので、なるべく短かい方が望ましい。そこで、図3で示されるような制約を満たす系列だけを記録することを考えてみる。上述のように、セルをレベル3にすると、リセット操作の伝搬は終了する。したがって、この制約を満たす系列を記録する場合には、リセット操作は高々三つしか伝搬しない。

実際にデータを記録するためには、データ系列を図3が表す制約を満たす系列に変換する必要があるが、そのための符号化と復号の規則を構成するための一般的な方法はよく知られている^{(17), (18)}。

5. メモリスタ (Memristor)

その存在が理論的に予想され、第4の回路素子と呼ばれていたメモリスタ (memristor) に対応する物理現象が2008年にヒューレットパッカード研究所によって発見された (文献(19)に良い解説がある)。この素子は、通過した電荷を抵抗値として記憶するため、記憶素子として用いることができる。この素子についても制約符号が有効であることが報告されている⁽⁶⁾。

メモリスタを記憶装置として利用するときには、縦横に導線を並べ、その交差したところにメモリスタを配置する。メモリスタは低抵抗状態と高抵抗状態を持つが、低抵抗状態のメモリスタが多いと大量の電流が流れ電力

を消費してしまう。そこで、縦の導線上にある低抵抗状態のメモリスタの割合がある値以下であり、横の導線について同じ制約を満たすようなパターンだけにデータを符号化する方法が考えられている。

E. Oldentlich らは、こうした条件を満たす二次元パターンにデータを符号化する非常に巧妙な方法を与えるとともに⁽⁶⁾、そのようなパターンの数の漸近的な振舞いを解析している⁽²⁰⁾。

6. ま と め

不揮発性固体メモリのための制約符号について述べた。今後も様々な記憶素子が開発されると予想されるが、高い記録密度、低い電力消費などを達成するために、素子の特徴に応じた制約符号を構成する問題が、記録システム全体の設計の中で、より重要な位置を占めるようになると思われる。

謝辞 フラッシュメモリの研究については科研費（基盤研究（C）課題番号 23560439）の助成を受けている。また、記録のための制約符号の研究については情報ストレージ研究推進機構（SRC）の援助を受けていることを付記し、ここに謝意を表します。

文 献

- (1) “特集：半導体ストレージ、迫る主役交代,” 日経エレクトロニクス, 2月7日号, pp. 38-59, 2011.
- (2) A. Jiang, V. Bohossian, and J. Bruck, “Floating codes for joint information storage in write asymmetric memories,” Proc. ISIT 2007, pp. 1391-1395, 2007.
- (3) A. Jiang and J. Bruck, “Rewriting codes for joint information storage in flash memories,” IEEE Trans. Inf. Theory, vol. 56, no. 10, pp. 5300-5313, Oct. 2010.
- (4) A. Jiang, R. Mateescu, M. Schwartz, and J. Bruck, “Rank modulation for flash memories,” IEEE Trans. Inf. Theory, vol. 55, no. 6, pp. 2659-2673, June 2009.
- (5) “IBM scientists demonstrate computer memory breakthrough,” <http://www.zurich.ibm.com/news/11/pcm.html>
- (6) E. Oldentlich and R. Roth, “Low complexity two-dimensional weight-

constrained codes,” IEEE Trans. Inf. Theory, vol. 58, no. 6, pp. 3892-3899, June 2012.

- (7) R. Rivest and A. Shamir, “How to reuse a write-once memory,” Inf. Control, vol. 55, pp. 1-19, 1982.
- (8) E. Yaakobi, P. Siegel, A. Vardy, and J. Wolf, “Multiple error-correcting WOM-codes,” IEEE Trans. Inf. Theory, vol. 58, no. 4, pp. 2220-2230, April 2012.
- (9) F. Chierichetti, H. Finucane, Z. Liu, and M. Mitzenmacher, “Designing floating codes for expected performance,” IEEE Trans. Inf. Theory, vol. 56, no. 3, pp. 968-978, March 2010.
- (10) H. Kamabe, “Floating codes with good average performance,” ISITA 2010, pp. 867-872, Taipei, 2010.
- (11) H. Mahdavi, P. Siegel, A. Vardy, J. Wolf, and E. Yaakobi, “A nearly optimal construction of flash codes,” Proc. ISIT 2009, pp. 1239-1243, 2009.
- (12) 鈴木理気, 和田山 正, “平均性能を指標とした Index-less Indexed フラッシュ符号,” 信学論(A), vol. J94-A, no. 12, pp. 991-1000, Dec. 2011.
- (13) Y. Kaji, “The expected write deficiency of index-less indexed flash codes,” IEICE Trans. Fundamentals, vol. E95-A, no. 12, pp. 2130-2138, Dec. 2012.
- (14) E. Gad, A. Jiang, and J. Bruck, “Compressed encoding for rank modulation,” ISIT 2011, pp. 849-853, 2011.
- (15) 戸部雅人, 鎌部 浩, “Compressed encoding を利用した Rank Modulation の最悪時最適化符号,” SITA 2011, no. 8.4.1, pp. 514-519, 荻宿, 2011.
- (16) A. Jiang, J. Bruck, and L. Hao, “Constrained codes for phase-change memories,” ITW 2010, pp. 1-5, 2009.
- (17) B. Marcus, P. Siegel, and J. Wolf, “Finite-state modulation codes for data storage,” IEEE J. Sel. Areas Commun., vol. 10, no. 1, pp. 5-37, 1992.
- (18) K.A.S. Immink, Codes for Mass Data Storage Systems, 2nd ed., Shannon Foundation Publishers, 2004.
- (19) R. Williams, “Finding the missing memristor,” <http://www.casttv.com>, <http://wn.com/Calit2ube>
- (20) E. Oldentlich, F. Parvaresh, and R. Roth, “Asymptotic enumeration of binary matrices with bounded row and column weights,” ISIT 2011, pp. 116-120, 2011.

(平成 25 年 2 月 4 日受付)



鎌部 浩 (正員：シニア会員)

昭 57 豊橋技科大・工・情報卒。昭 59 同大学院修士課程了。同年三重大助手。平 10 岐阜大講師。現在、同大学教授。博士（工学）。記録のための符号に関する研究に従事。