

# 記憶装置の誤まりの検出と訂正\*

高島 堅 助\*\*

## 1. まえがき

初期の計算機においては計算機の誤まりについては何も考えられていなかった。このため計算結果が予想とかけ離れているとか、計算機が異常な動作を始めたりするということがなければ、とにかくプログラムが一通り終了してしまい、計算機の誤まりに気づかないということもあった。ところが計算機の誤まり、特に記憶装置の誤まりというものはかなり多いもので、これを見逃すと何時間もかけた計算が全部無効なものとなる。そこでまず考えられたのは、記憶装置内の情報に冗長なものをつけ加えておき、これによって誤まりを検出し、もし誤まりが検出された場合には、それ以上の計算を進めても無効であるから、計算機の動作を一切停止し、誤まりを表示ランプまたは音で使用者側に知らせるという方法である。

初期の科学用の簡単な計算機、すなわち、数千語の主記憶装置と簡単な入出力装置を持つものでは、この誤まり検出機能を設けることにより、停止した際の各種のレジスタの内容から記憶装置の誤まりの生じた位置を発見することは比較的容易で、この方法は十分に有効であった。

しかしながら、計算機が次第に高級になり、記憶装置が主記憶装置のみでなく、大容量の磁気ドラム、磁気ディスク、磁気テープなどの補助記憶装置を各種何台も同時に使用するようになると、この検出・停止という方法では不便を生じて来る。すなわち、数多くの記憶装置のどれかに一つでも誤まりを生ずると全計算機の機能が停止してしまい、多重プログラム処理を実行している場合に、障害を生じた記憶装置に関係のないプログラムの進行まで停止してしまう。

次に、障害を生じた場合に、その位置と原因を調べるのに長時間計算機を停止せねばならず、これは高価な計算時間の著しい浪費となる。これを防ぐには、誤まりを生じた際に、その位置、誤まりの種類などに関する情報を発生するようにしておき、これを計算機自身がプログラムによって探し、これらに関する情報お

よび対策を計算機の操作者にわかりやすい文章で印刷機に打たせて外部に出すという形が望ましい。このためには、誤まりに関する情報を示す各レジスタ（各1ビット）に対する条件付飛び越し命令を作って、記憶装置に読み書きするごとに、誤まりの有無を調べる方法を取ればよい。この方法によれば記憶装置のうち、プログラムの部分以外の誤まりに関する限り、計算機自身がかかなり詳細に原因を調査することによって、誤まりに対する操作者の対策を容易、かつ高速化することができる。

いま、プログラム部分以外の誤まりの処理はできると述べたが、それではプログラムの部分が誤まった場合にはどうにもならないものであろうか。これが実は可能なのである。しかし、そのためには割込みという機能を計算機に組み込んでおかねばならない。割込みの機能というのは、計算機が、ある特定の条件に遭遇した場合に、現在実行中の命令より先へのプログラムの進行を中止し、その時の各種のレジスタの内容を記憶装置の特定番地に格納し、別の番地（特定の番地または記憶装置の特定の番地に記憶されている番地）からプログラムを始める機能である。この別の番地以降に置かれているプログラムを優先プログラムと呼ぶが、この優先プログラムにより、割込みの生ずる前に実行していた命令に誤まりがあったかどうかを調べればよい。もちろん、優先プログラムの命令の部分にも誤まりを生じた場合にはお手上げであって、この場合には計算機は全機能を停止せざるを得ない。現在の大型の計算機の多くは割込みの機能を持っており、主記憶装置のみならず、すべての補助記憶装置の誤まりの事後処理は割込みの機能を使って処理するのが普通である。

以後、本文には、計算機における誤まり検出符号、各種の記憶装置に対する誤まり検出方法とその訂正方法、誤まりを生じたのちの処理などについて述べよう。

## 2. 計算機に使用される誤まり検出・訂正の可能な符号

計算機においてこれまで使用されて来たのは、ほとんどが簡単な1ビットだけの誤まりの検出可能な符号

\* Treatment of Errors in Computer Memories, by Kensuke Takashima (Electrical Communication Laboratory)

\*\* 電気通信研究所

であって、Hamming 符号その他の高級な誤まりの訂正可能な符号はほとんど使用されていない。これは計算機の記憶装置においては誤まりの確率が小さく、簡単な検出のみで十分に使用にたえ得ること、高級な訂正可能な符号を使用しても、そのための記憶素子の増大、符号化・復号化回路が複雑になり、速度・経済性などの点からおり合わないことがその理由と思われる。以下これまで使用されて来た二、三の符号について述べよう。

(1) 奇偶検査符号 これは一群の  $n$  ビットからなる情報に対して 1 ビット余分のビット、すなわち奇偶検査ビットを設けておき、 $n+1$  ビットの中 1 の個数が偶数(または奇数)になるように検査ビットの値を定め、この  $n+1$  ビットを記憶装置に書き込む。記憶装置から情報を読み出した時、“1”の個数が偶数(または奇数)かどうかを調べ、偶数(または奇数)ならば 1 個または奇数個の誤まりが無いことがわかる。 $n+1$  ビットの中、2 個以上誤まる確率が極めて小さく、誤まりがあってもせいぜい  $n+1$  個中の一つの誤まりしか起らないような場合には、この方法は極めて有効であり、現在最も多く誤まり検出の方法として使用されている。

(2) 5 者択 2 符号\* 10 進数計算機でしばしば使用される符号で、5 ビットで 1 桁の 10 進数を表わす。5 ビットのうち、2 個だけを“1”にした場合の組み合わせは  ${}_5C_2=10$  で、ちょうど 10 個になる。“1”の個数が常に 2 個かどうかを検査することにより誤まりを検出できる。5 ビットのうち、1 が 0 に誤まった個数と 0 が 1 に誤まった個数が等しいような誤まりは検出できないが、そのような誤まりは実用上極めて少ない。0 から 9 までの数値とこの符号との対応のさせ方はいろいろあるが、第 1 表のように 5 ビットに 01247 の重みを付けると、0 を除いては全部符号の重みの和が 10 進数値に対応するようになり便利な符号である。

(3) 2-5 進符号\*\* これは一種の 7 者択 2 符号であって、第 2 表に示すように 7 ビットのうち、2 ビットで 5 の有無を示し、他の 5 ビットのうち 1 ビットを“1”にすることで、01234 のどれかの値を示す。したがって最初の 2 ビットのうち一つが“1”で、残りの 5 ビットのうちの一つが“1”であることを検査することにより誤まり検出できる。この符号はかつて継電器

第 1 表 5 者択 2 符号

重み 数値	0	1	2	4	7
0	0	0	0	1	1
1	1	1	0	0	0
2	1	0	1	0	0
3	0	1	1	0	0
4	1	0	0	1	0
5	0	1	0	1	0
6	0	0	1	1	0
7	1	0	0	0	1
8	0	1	0	0	1
9	0	0	1	0	1

第 2 表 2-5 進符号

重み 数値	0	5	0	1	2	3	4
0	1	0	1	0	0	0	0
1	1	0	0	1	0	0	0
2	1	0	0	0	1	0	0
3	1	0	0	0	0	1	0
4	1	0	0	0	0	0	1
5	0	1	1	0	0	0	0
6	0	1	0	1	0	0	0
7	0	1	0	0	1	0	0
8	0	1	0	0	0	1	0
9	0	1	0	0	0	0	1

式の計算機で使用されたが、最近の電子計算機ではあまり使用されない。

(4) 2 進化 10 進符号\*\*\* 10 進数計算機では通常 4 ビットで 1 桁の 10 進数を表わす。4 ビットで 0 から 16 までの数字が表現されるが、このうち 0 から 9 までの 10 個しか使用しないから、10 から 15 までの 6 個の符号は、10 進計算機内では表われないはずである。したがって、10 以上の値を持つ符号を検出することによって誤まりの一部を検出できる。これは完全な誤まりの検出法ではないが、10 進法計算機ではしばしば採用される方法である。

### 3. 各種の記憶装置における誤りの特徴と検出法

記憶装置の誤まりの原因を大別すると、次のようになる。

- i) 記憶媒体の機械的破損または物理的特性変化
- ii) 周辺回路素子の特性変化
- iii) 周辺回路の配線またはコネクタなどの劣化
- iv) 動的記憶装置(磁気テープなど)の記憶媒体表面に対する一時的異物の附着

記憶装置の種類によって上記の原因の占める割合が違うので、誤まりの対策も、それぞれ多少異なって来る。以下代表的な装置の種類それぞれについて、誤まり検出または訂正方法を述べよう。

#### 3.1 磁心記憶装置

磁心記憶装置の誤まりの原因としては、周囲温度の変化による磁心特性の変動、周辺回路内の特性変化による駆動電流の変化、読取り増幅器の利得変化、配線不良などが主要なものである。一たん誤まりが出た場合には、自動的に正規の状態に復旧することはなく、それ以後その記憶装置を引続き使用することは無意味で、何らかの人手による調整または修理を要する。磁心記憶装置では、一般に 1 語ごとに並列に読み書きす

\* two out of five code

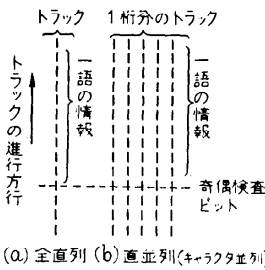
\*\* biquinary code

\*\*\* binary-coded decimal code

る並列方式がとられるが誤まりの検出法としては、1語をさらに幾つかに分けて、数ビットないし10ビットごとに1ビットの奇偶検査ビットを用いるのが普通である。誤まりの確率だけからいえば、実用上1語に1ビットでも良いが、全ビットに対する2を法とする和を作る回路の段数が増加し、このための所要時間が増すことを考慮に入ると、数ビットごとに入れておくことが楽であり、また、読み出した時に全部の奇偶検査ビットを表示するようにしておけば、誤まりの位置の診断が容易になる。

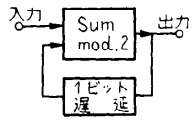
3・2 磁気ドラム・磁気ディスク記憶装置

これらの場合には1語は1トラック内に第1図(a)のように全部直列に記憶されるか、(b)のように10



第1図 ドラム・ディスクの奇偶検査

進の1桁または、一つの文字のビット数だけ並列に記憶させる使い方が多い。この場合にはドラム上の各トラック上の1語の区切りの所に1ビットの検査ビットを設ける。この場合には各トラックごとに情報は直列に出し入れされるから、第2図のような入力の排他的論理和回路と1ビット分の遅延素子からなる奇偶検査作製(検出)回路を設けておけばよい。



第2図 直列型偶検査・作製回路

3・3 磁気テープ記憶装置

この場合には、前2者と異なり、(1)機械的に丈夫でないテープが機構部と接触しながら急激な運動を行なうこと、(2)テープの起動・停止のためにかなり複雑な運動機構を使用していることにより、誤まりの起こる確率が前に述べたものよりもはるかに大きい。磁気テープ装置特有の主な誤まりの原因を挙げると次のようなものがある。

- (i) 磁気テープの製造時における磁性膜面の欠陥または異物の混入による雑音の発生またはドロップ・アウト\*

プ・アウト\*。

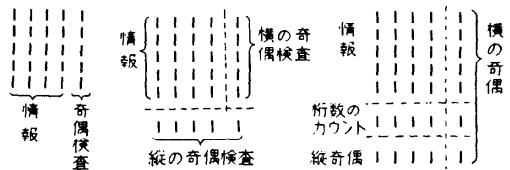
- (ii) 長期使用後の磁気テープの摩耗による磁性膜面の一部の脱落によるドロップ・アウト。
- (iii) 磁気テープと磁気ヘッドの間に一時的に異物が入るための一時的なドロップ・アウト。
- (iv) 磁気ヘッド上に磁性体その他の物が付着して、ヘッド面と磁気テープが密着しなくなることによる読取りヘッド出力レベルに基づくドロップ・アウト
- (v) 磁気テープの横ぶれにより、磁気テープと読取りヘッドのトラックがずれることによる読取り出力減少に基づくドロップ・アウト。
- (vi) 数台の磁気テープ装置を用いる際、ヘッドの調整位置の狂いにより他の装置で書いたテープを他の装置で読む時に生ずる出力減少による読取り不能。
- (vii) 磁気テープ装置の駆動停止機構の誤動作による、一部情報の重複書き込み、または読み飛ばし。

磁気テープを実際に使用すると、上記の誤まりのいずれかが極めて頻々と生ずるので、これらに対する誤まりの検出と訂正方法は極めて重要である。

ここでは磁気テープ装置自身における局所的な各種の誤りに対する検出および訂正の方法について述べよう。

3・3・1 誤まりの検出または訂正のための磁気テープ上の情報パターンについて

まず最も普通に考えられるのは第3図に示すように情報トラックの他に奇偶検査のために1トラック増設し、1行ごとに奇偶検査を行なう方法がある。さらに情報1ブロックに対し、第4図に示すように縦方向にも1行追加して、奇偶検査を行なえばさらに完全である。この方法によれば単一のドロップ・アウトに対し



第3図 奇偶検査トラック

第4図 縦・横の奇偶検査

第5図 桁数または語数の誤まり検出

\* ここで、ドロップ・アウトとは磁気テープ状に書かれたはずの信号を読みとった時、出るべきパルスが出ない現象をいう。

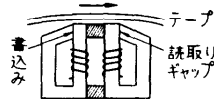
てはかなり有力であるが、前節(v)のような誤まりに対しては見逃がす可能性が大きいので、第5図のように情報の次に1ブロックの全語数または桁数を、ある数 $n$ を法として表わしたものを書き込んでおき、読み取りの際に全語数または桁数の計数値と照合すれば(v)のような誤まりを検出することができる。

以上の方法は通常の装置で現在最も広く行なわれている方法で、実用上は十分であろう。

その他にMITのTX-2その他で使われている方法として、情報に必要なトラック数の2倍のトラックを使用し、全く同じ情報を二つ同時に記憶する方法がある。磁気テープの場合には、読み取りの際には信号が消えることが多く、雑音によってふえる確率は極めて少ない、いわゆる非対称なチャンネルであるから、MITでは二つのトラックのパルス出力のORをとって、その他にはなんら誤まり検出機構を設けていない。

### 3.3.2 書き込み時における誤まり訂正

磁気テープ装置で大切なことは磁気テープ上に正しい情報を書くことである。前節(i)(ii)のような理由で磁気テープ上には必ず書き込み不能の箇所があると考えなければならない。この部分避けるには二つの方法がある。一つのトラックをクロックに使用し、まず全テープにすべて"1"の情報\*を書き込み、次にこの情報を読み取り、脱落がある部分のクロックを消し、実際に使用する場合には、このクロックを頼りに読み書きする方法がある。第2の方法は第6図に示すような二重ギャップヘッドを使用し、第1のギャップで書くと同時に第2のヘッドで読み取り、横方向の奇偶検査を行ない、もし



第6図 二重ギャップ・ヘッド

誤まりがあれば、その情報ブロックの始めの位置にテープをもどし、そのブロックを消して、その後新しく書き直すのである。この第2の方法が現在最も広く用いられているようである。

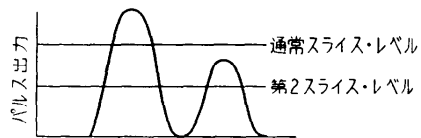
### 3.3.3 読み取り時における誤まりの訂正

前に述べた(iii)のように、一時的読み取り不能となることが磁気テープでは最も多い。したがって誤まりが検出された場合、テープを戻して読み直すということを何回か繰り返せば正しく読み取れる場合が多い。数回以上読み直しても読み取れない場合は磁気テープに欠陥を生じたか、装置のどこかに修理を要する故障を生じたもので、放置しておいて自動的に復旧す

\* すべての情報位置に磁化反転がある情報

るものではない。

誤まりの訂正の一つの方法として、2レベル読み取り法がある。これは第7図のように、読み取られたパルスを通常は上のスライス・レベルで、誤まりを生じた場合には下のスライス・レベルで読み取り、もし誤まりが生じなければこれを正しいものとして読み取る。この方法は何らかの原因による出力レベルの一律低下という障害を救うことができる。



第7図 2レベルによる面読み取り

## 4. プログラムによる誤まりの処理

以上、各記憶装置自体についての誤まりの検出・訂正の方法について述べた。計算機システムとしては誤まりがいくつかの記憶装置で発生した際には、再読み取り、修理などの処置をする必要があるため、その位置、故障の種類を保守者に知らせる作業などを、プログラムにより自動的に行なわせることが望ましい。このプログラムとして最も参考になるのはSAGEシステムに用いられた方法である。

これについては、本会誌の文献紹介にすでに2回ほどかかげられているが、その要点を述べよう。

SAGEシステムではFIXと呼ばれるプログラムにより誤動作分析と自動訂正が行なわれる。プログラムはコア50語、ドラムを5,000語使う。FIXは(1)誤動作監視、(2)誤動作分析、(3)報告書作成、(4)回復の四つの部分からなる。

(1) 誤動作監視 これは記憶装置の奇偶検査のほか、オーバー・フロー、くりかえしの時間超過などを検査し、誤まりが生じた時、FIXプログラムへの割り込みを生ずる。

(2) 誤動作分析 アキュムレータ、インデックス・レジスタなどをコアに保存する。次に(i)誤まりを生じた位置、(ii)その誤まりは続いて起るものか、(iii)続けて起る性質なら Learning Table を作製し、診断の助けとする。

(3) 報告書作製 すべての誤動作を Learning Table に記録すると同時に、テレ・タイプによって保守者、オペレータにわかるように報告書を打ち出す。

(4) 復元 復元方法は多種多様であるが、割り

込みの際に退避した情報をもどして安全なプログラム・ステップから再開するのに 30 ms を要する。

SAGE システムの誤動作に関する統計は

ドラムの奇偶誤まり	53%
コアの奇偶誤まり	10%
無意味なくり返し	21%
その他（電源、操作誤まり）	16%

これによれば全動作の 84% は FIX で処理できる。しかし長期テストの結果は 92% が FIX で処理され、6% が操作誤まり、残り 2% が予定外の保守を要するものであった。

最後に、通研で現在開発中の料金計算用大形計算機 (CAMAC) の誤まり対策について述べよう。この計算機は優先順序を持つ三つのプログラム・レベルと、単に最優先のモニタ・レベルを持つ。三つのレベルの上位への割り込みは、周辺装置の動作終了その他の原

因により行なわれる。この際、各種のレジスタの内容は自動的に退避される。磁気テープその他の周辺装置の誤まりは、動作終了により割り込まれたプログラムの中で位置、種類が調べられる。三つのプログラムレベル内のいずれかにおいて、主磁心記憶装置自体の誤まりその他本体関係の誤まりが生じた時には、モニタ・レベルに上がり、メタル・カード記憶装置の特定番地に飛び、ここで監視プログラムに入り、誤まりの種類位置が調べられる。

モニタ・レベルで上記の誤まりが生じた時には万事休すで、計算機は全面的停止の止むなきに至る。したがって、監視プログラムで誤まっていると覚しき語を読む時には、プログラムレベルを下に落す必要があるのはもちろんである。これらのプログラムの作製は今後のわれわれの一つの課題である。

(昭和 38 年 11 月 29 日受付)