

電子計算機簿記と精算表形式決算

— 財産・損益一覧表としての

試算表機能の拡充と強化*—

藤 田 芳 夫

はじめに

筆者はさきに発表した拙稿「伝統的複式精記とコンピューター」⁽¹⁾において、電子計算機により伝統的複式簿記を遂行する場合、経営システムのオートメーション化が強くと実行される場合には、仕訳そのものも廃止される可能性があるが、そうでない場合においても仕訳帳は不必要になり廃止されること、また、元帳転記の正確性を検証する手段としての試算表もその意義を失なうことを明らかにした。

しかし、伝統的複式簿記を電子計算機化する場合、複式簿記の構造にひきおこされる変化は以上の点につきるものではない。こうした変化の一つとして、本稿では前稿で展開した方法を前提にしながら、電子計算機複式簿記における決算は如何に行なわれるか、という問題を取りあげることとする。そして、その結果、電子計算機複式簿記においては、財産・損益一覧表としての試算表の職能が益々強化されることを明らかにしたい。

* 拙稿「伝統的複式簿記とコンピューター」(小樽商科大学, 商学討究, Vol. 19, No. 1) にひきつづき, 本稿の執筆に際しても, 一橋大学名誉教授片野一郎博士, 小樽商科大学教授石河英夫, 同古瀬大六, 同麻田四郎, 同竹内清の諸先生から有益な批評と激励をいただいた。ここに記して謝意を表します。

(1) 拙稿「伝統的複式簿記とコンピューター」, 小樽商科大学, 商学討究, Vol. 19, No. 1, 1968年8月。

1. 決算と勘定の三つの記入要素

(主記憶装置の特徴に対応して)

通常、「決算」は決算予備手続と決算本手続に分けられる。決算予備手続は(1)帳簿記録の正確性を確かめること(ゾル・ベスタント計算の正確性の確認)および(2)財産目録または棚卸表の作成(イスト・ベスタントの確認)をその内容とし、決算本手続といわれるものは(1)帳簿記録(ゾル・ベスタント)と実地棚卸高(イスト・ベスタント)との間に差異がある場合、帳簿記録を後者に合致させるための決算整理記入、および(2)かくして実際の状態に合致させられた勘定から損益計算を総括し、(3)また財産計算の総括を行なうことがその主たる内容をなすといつてよいであろう。

ところで、決算予備手続のうち、帳簿記録の正確性をたしかめることは、試算表の作成に至る手続の正確性を問題とすることであるから、前掲拙稿で明らかにしたように仕訳の「計算上の正確性」と「転記の正確性」を確保するプログラムが電子計算機簿記のプログラムの中に組み込まれていれば、改めて問題にする必要はない。

決算予備手続のうち、第二の財産目録または棚卸表の作成という問題は、電子計算機により自動的に行なうことはきわめて困難である。そもそも、財産目録または棚卸表の作成ということは、(1)実地棚卸を行なうこと、(2)実地棚卸の結果に対し価値づけを行なうこと(評価すること)、そして(3)期末決算のための棚卸表の場合にはゾル・ベスタントたる帳簿残高と評価後の実地棚卸高とを比較し、修正すべき金額を算出するという三つの要素を含んでいる。

化学工業等の場合のようにプロセス・オートメーションが行なわれている場合にはタンク内部の資産の量的測定は自動的に行ないうるであろう。しかし、現在の状態の下では電子計算機が、あらゆる企業のあらゆる勘定項目について自動的に実地棚卸を行なうなどというわけにはゆかない。したがっ

て、実地棚卸は人間が行なうと考えねばならない。実地棚卸の結果に対し価額づけを行なうことから後の段階は、労力を払うことさへいとわなければプログラミングも不可能ではないであろう。

しかし、本稿では問題を単純化し、かつ、伝統的複式簿記を EDP 化する場合の基本的な側面に集中するため決算整理仕訳は通常取引の仕訳カードと同じ形で与えられるものとして考察する。すなわち、各勘定の借方または貸方残高は前掲拙稿までのプログラムによる処理で与えられているものとし、また、人間が行なった実地棚卸の結果、各勘定の修正金額が与えられるものと仮定するのである。

かように考えると、決算のためには勘定配列 A、あるいは勘定配列 A を修正して作った残高試算表がありさえすればよいように思われる。しかし、電子計算機簿記で決算を行なうさい、次の点に注意しなければならない。

すなわち、前掲拙稿で使用した例についていえば、売掛金は A(2,1) および A(2,2) を用いて計算した。したがって、売掛金の借方残高を算出する以前の状態は、下図のように A(2,1) に 40,000 円があり、A(2,2) に 39,000 円がある。ところが借方残高算出前

	A(2, 1)	A(2, 2)	
売掛金借方残高	40000	39000	1,000 円を A(2, 1) に算出すれば A(2, 2) には当然ゼロを入れなければならない。しかし、こうすると今期の掛売上高総計 40,000 円と売掛金回収高 39,000 円という重要な資料は主記憶装置のなかに展開してある勘定配列 A に関するかぎり完全に失なわれてしまう。このことは他の勘定についても同じである。

したがって、各勘定の借方合計金額および貸方合計金額を失なうまいとすれば、それらの金額をライン・プリンターやカード・パンチあるいは磁気テープのような装置を通じて外部に取出しておくか、あるいは今期の取引額を計算する場所と勘定残高を算出する場所とを主記憶装置の内部で区別しておかなければならない。

換言すれば、ペンとインクによる手記複式簿記のもとでは元帳の各勘定

資 産 勘 定

a) 期首繰越高	×××	c) 今期減少高	×××
b) 今期増加高	×××	d) 次期繰越高	×××
(t) <u>×××</u>		(t) <u>×××</u>	

は、たとえば資産勘定に
ついてみれば左図のよう
に前期繰越高と今期増加
高およびその合計額、あ

るいは今期減少高を算出するだけでなく、次期繰越高をも算出し、しかもそれらの数値は歴史的記録として遺されてゆくのである。ただし、手記簿記システムでは、上図のb今期増加高とc今期減少高はそれぞれ一個の合計額としてではなく、合計算出の素材として表われることは言うまでもない。

資産勘定だけでなく、伝統的複式簿記の勘定記入を一般化してみると、それは

- 1 前期繰越高
- 2 各勘定の借方および貸方増加高
- 3 次期繰越高

の三つの要素に区分することができる。そして、それらすべてが歴史的記録として遺されて行くのである。

したがって、電子計算機複式簿記でもこれら基本的な重要資料を残しながら処理して行くプログラムを考えなければならない。

実は、この問題は電子計算機簿記の基本的問題の一つである。というのは電子計算機簿記をその一部分として包含する EDP の（あるいはマネジメント・インフォメーション・システムの）基本的な特徴の一つは帳簿の統合（file consolidation⁽²⁾）であるから、電子計算機複式簿記の元帳も当然この面から考察されなければならない。しかし、同時に、監査という面からみれば、伝統的複式簿記は証拠の体系であり、元帳はその中心的位置を占めるものである。

したがって、本稿では元帳の具体的な条件を論ずるよりも、むしろ決算と

(2) ファイル・コンソリデーションについては拙稿「電子計算機による販売管理会計」（商学討究, Vol. 18, No. 3）pp. 91, 124 を参照されたい。なお同稿に引用した文献を見られたい。

いう面からみると、こうした条件の一つとしてどのようなことが浮び上がってくるかという点を明らかにすることに重点を置いて以下論を進めてゆくことにする。

かように考えると、前掲拙稿で行なったように唯一の二次元配列 $A(I, J)$ を使用するのではなく、前期繰越高について一個の二次元配列 $A(I, J)$ を使用し、今期中の取引について別の二次元配列 $B(I, J)$ を、また次期繰越高にも同じく一個の二次元配列 $C(I, J)$ を割当てて使用しなければならなくなる。

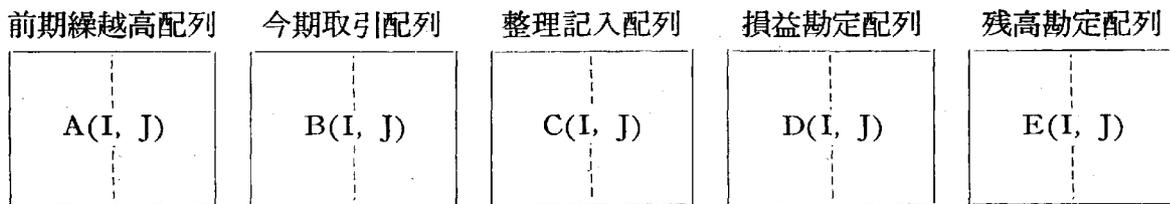
次に、決算本手続の一つである決算整理記入は、結局、今期中の借方増加高または貸方増加高の修正であるから、配列 B に直接記入してもよい。しかし、この方法では伝統的複式簿記に慣れているわれわれにとり、いくぶん奇異の感をまぬがれない。それは決算のための勘定修正過程が眼に見える形で残らないからである。また、決算本手続の他の重要な手続、すなわち損益勘定、残高勘定の作成の過程も伝統的手記複式簿記に慣れているわれわれとしては眼に見える形で残したいという心理的要求を無理に排除しても得るところは少ない。したがって、本稿では伝統的複式簿記の決算手続をコンピュータライズする方法としてきわめて精算表に近い形を採用することにする。

伝統的複式簿記の理論のなかでは、精算表は決算本手続はいうまでもなく、決算予備手続にも入らない。それはきわめて便利なものではあるが、仕訳帳、元帳という複式簿記の基本的記録体系（すなわち帳簿体系）とは切り離されたものであり、精算表の作成は決算手続それ自体の構成分子ではなく、いわば決算の予行演習であり、また教育上便利な道具であるにすぎない。

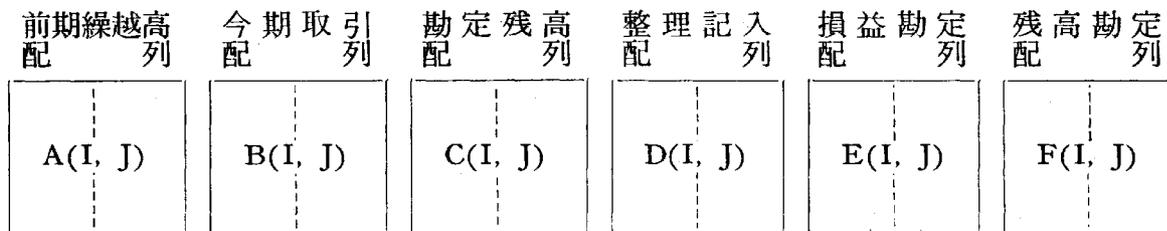
しかし、電子計算機複式簿記では上述したような主記憶装置の特性から精算表ときわめてよく似た形で決算を遂行せざるをえないのである。すなわち前期繰越高、今期増減高、整理記入高、損益勘定、残高勘定等の一連の基本的資料を明白な形で取り出すためである。

2. 決算と三次元配列 A(I, J, K)

前節でのべた考え方にもとづいて決算の方法を具体化すると、まず使用する配列は下図のように並ぶことになる。



しかし、前節で指摘したように帳簿残高と実地棚卸高との比較照合を行なった結果、整理記入が行なわれるのである。したがって、当然のことながら伝統的複式簿記では決算の出発点は合計試算表ではなく残高試算表である(精算表の出発点も同じ)。そこで、われわれも上図に残高試算表に担当する勘定残高配列を加え、決算のための基本的な配列群を(図表2-1)のように変更することにする。



(図表2-1) 決算で使用する勘定配列

かように六個の勘定配列を使用して決算を実行しようとする時、前掲拙稿で行なったように勘定配列Aだけを考えればすむというわけには行かない。すなわち、前期繰越高に関する情報は配列 A(I, J) に入れ、今期取引高に関するものは配列 B(I, J) に入れ等々というように正しく使いわけねばならない。のみならず勘定残高配列から残高勘定配列までの演算は、これらいくつもの配列の間の数値のやりとりである。そこで、各勘定配列の区別と相互の関連を容易に行ないうるようにしておく方が賢明である。

通常 FORTRAN IV コンパイラーでは三次元配列まで使用できる。そこ

で本稿では以下三次元配列 $A(I, J, K)$ を使用する。すなわち、

前期繰越高配列	$A(I, J)$ は	$K=1$	$A(I, J, 1)$
今期取引配列	$B(I, J)$ は	$K=2$	$A(I, J, 2)$
勘定残高配列	$C(I, J)$ は	$K=3$	$A(I, J, 3)$
整理記入配列	$D(I, J)$ は	$K=4$	$A(I, J, 4)$
損益勘定配列	$E(I, J)$ は	$K=5$	$A(I, J, 5)$
残高勘定配列	$F(I, J)$ は	$K=6$	$A(I, J, 6)$

という関係になる。

なお、念のため、この関係を（図表 2 - 1）と同様な形で示すと次のようになる。

前期繰越高配列	今期取引配列	勘定残高配列	整理記入配列	損益勘定配列	残高勘定配列
$A(I, J, 1)$	$A(I, J, 2)$	$A(I, J, 3)$	$A(I, J, 4)$	$A(I, J, 5)$	$A(I, J, 6)$

（図表 2 - 2） 三次元配列 $A(I, J, 6)$ を用いた場合の勘定配列の関係

前掲拙稿に明らかにしたプログラムは仕訳の「計算上の正確性」と「転記の正確性」の二つの要請に答えるプログラムであるが、実はこのプログラムは通常取引を今期取引配列や前期繰越高配列に転記するために使用できるだけでなく、決算整理仕訳を整理記入配列に転記する際にも使用しなければならない。何故なら、決算整理仕訳も通常取引の仕訳と同様に、その仕訳計算の正確性と勘定コードの正当性を検査しなければならないからである。

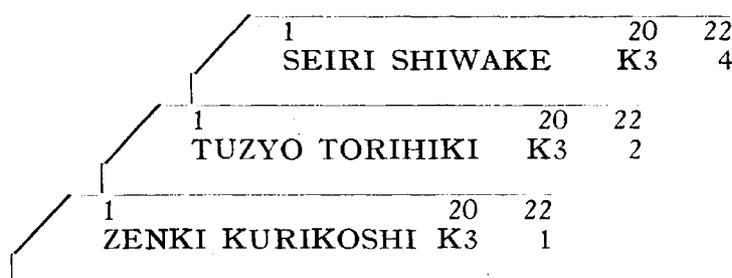
しかし、こうなると前掲拙稿の最後に示したかなり大きなプログラムを何度も書かねばならなくなり、不合理である。そこで、そのプログラムのメイン・プログラムの主要部分をサブルーチンとして使用する。そもそも、前掲拙稿の最後に示したプログラムは仕訳帳を廃止する機能をもっていた。そこ

で、これをサブルーチンとする場合、名前を BENTRY としよう。すなわち基本的転記プログラム (Basic Entry Program) というほどのつもりである。

このように基本的転記プログラム BENTRY をサブ・ルーチンとして使用するとき、たとえば通常取引の仕訳カード・デッキと決算整理取引の仕訳カード・デッキとをどのようにして区別したらよいか、という問題がおきる。

この区別をするため、通常取引の仕訳カードや決算整理取引の仕訳カードに、それぞれの分類コードをパンチしておいてもよい。しかし、こうすると非常に多くのカードに分類コードをパンチしなければならなくなり、能率的な方法ではなくなる。また、カード設計の点からみても、それだけスペースが減少し、他の必要事項がパンチできなくなる。そこで、前期繰越高カード、通常取引カード、決算整理仕訳カードの先頭に制御カードを一枚ずつ付けてやり、この制御カードによって転記先の勘定配列を区別することにする。

すなわち、次図に示すように前期繰越高を示すカードの先頭には K3=1



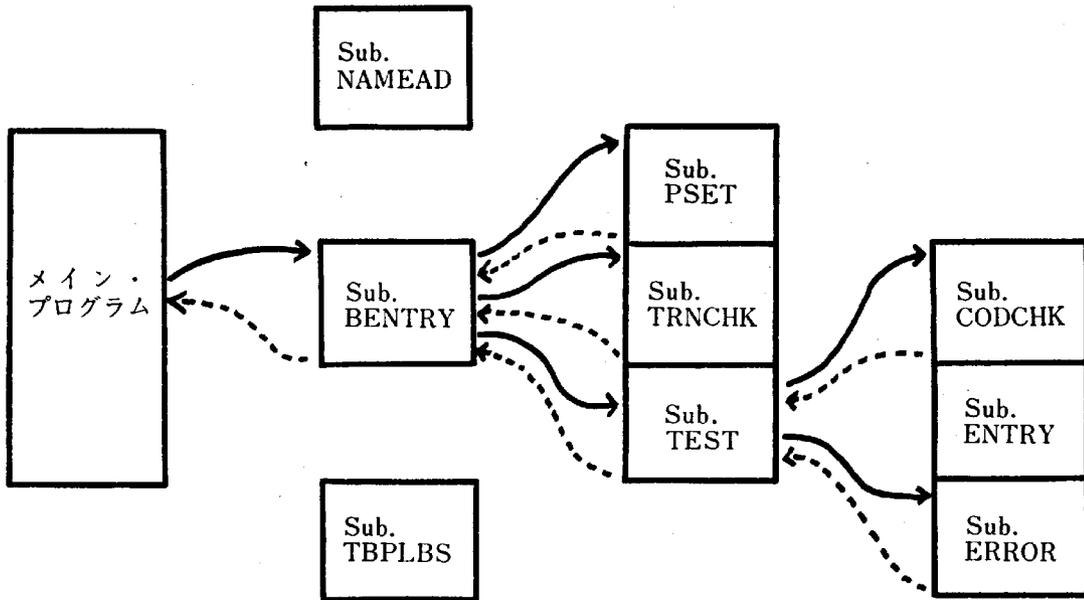
の制御カードを、今期取引の仕訳カードの先頭には K3=2 の制御カードを、そして整理仕訳カードの先頭には

K3=4 の制御カードをつけてやるのである。

3. 決算手続をふくむ電子計算機複式簿記の

メイン・プログラム

上述したように、前掲拙稿で展開した電子計算機複式簿記のプログラムは、通常取引の処理に際して使用されるだけでなく、決算をもふくむ、すべ

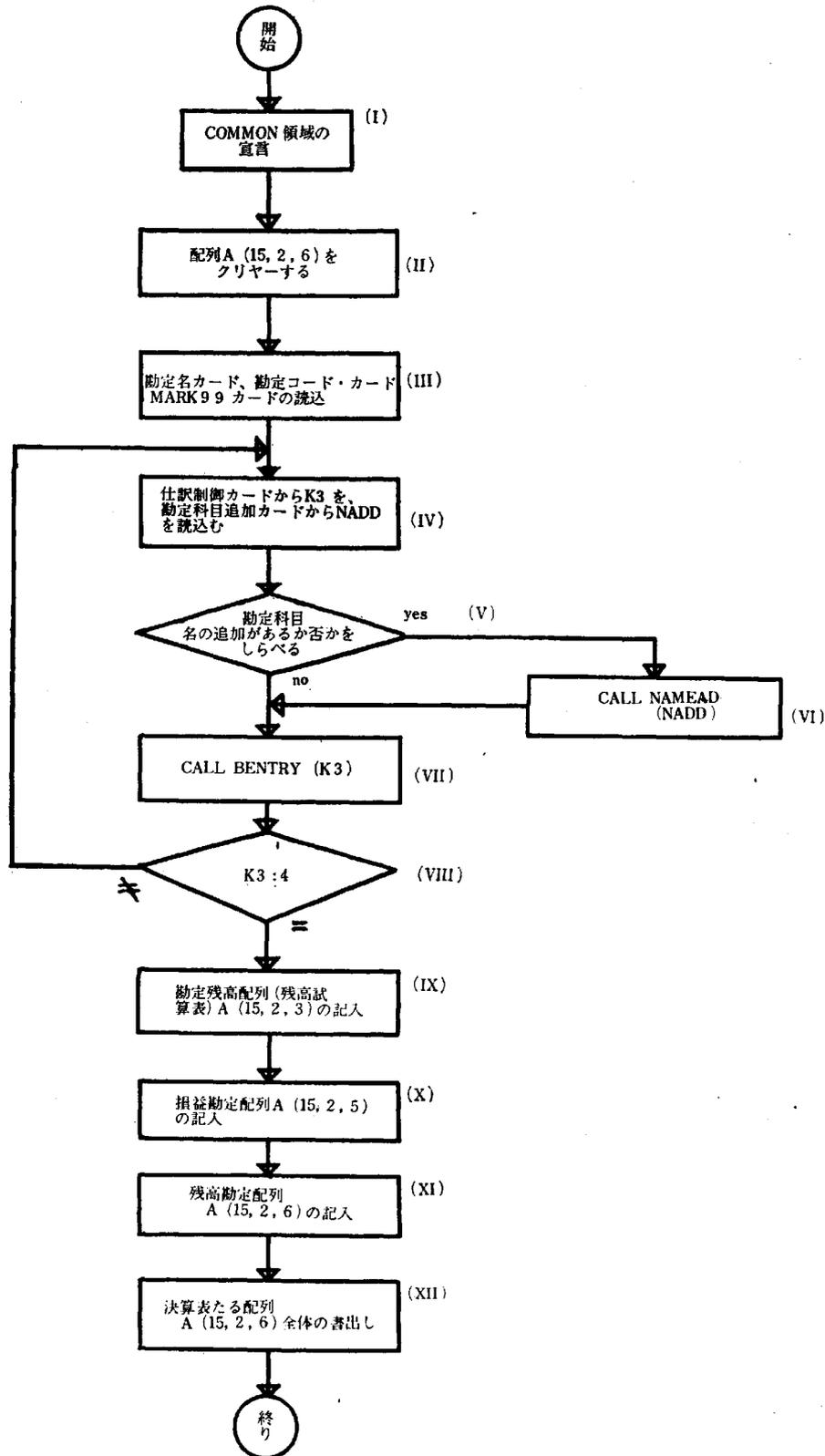


（図表 3-1） メイン・プログラムとサブルーチンの関係

ての複式簿記のプロセスの中で使用されるがゆえに、それ自体をサブルーチンとして使用する。したがって、決算を含む複式簿記の一般的プログラムは（図表 3-1）のような構造を持つことになる。たとえば、通常取引の仕訳カードを処理している場合、メイン・プログラムは仕訳カードの先頭につけられている制御カードでサブルーチン BENTRY をよび、そこで一枚の仕訳カードの処理の仕事をする。その際必要に応じて更に下位のサブルーチンを使用することは当然である。もし、この時仕訳カードに「計算上のエラー」が発見されるならば、サブルーチンの ERROR に飛び必要なエラー・メッセージを出してくれることになる。

すなわち、前掲拙稿で作成したプログラムのメイン・プログラムは本稿では四階建の建物の二階の一部分に相当するサブルーチンとなり、前掲拙稿でサブルーチンとして現われたものは本稿ではこの建物の三階および四階に位置するサブルーチンと考えることができる。

このプログラムの二階にはサブルーチン BENTRY のほか、決算その他の



(図表 3 - 2) 決算終了までの複式簿記手続きのフローチャート

事情で勘定科目を追加する必要がある場合、勘定を追加するためのサブルーチン NAMEAD と、残高試算表、損益勘定、残高勘定等に相当する勘定配列の作成にも同時に使用されるサブルーチン TBPLBS も置かれることになる。

プログラム全体の構造をこのように考えると、メイン・プログラムのフローチャートは（図表 3-2）のようになる。

（図表 3-2）に示したメイン・プログラムのフローチャートについてきわめて簡単に説明を加えておこう。第Ⅳボックスで制御カードから K3 の内容を読み込み、以下に続く仕訳カードの種類を判定する。また勘定科目の追加があるか否かを判定し、追加がある場合にはまず勘定科目の追加を先に処理してから仕訳カードの処理に移るのである。

この仕訳カードの処理はサブルーチン BENTRY で行なわれる。一種類の仕訳カードの処理が終了すると、そのブロックの終りを示すブロック・エンド・カードが現われ、メイン・プログラムに戻る。第（Ⅷ）ボックスのテストは、こうしたブロック毎の処理を繰り返して、決算整理仕訳カードの処理まで、全部終了したか否かを検査することを示しているのである。

したがって、決算整理仕訳がない場合でもデータ・カード・デッキの構成は K3=4 の制御カード、NADD=0 の勘定名の追加がないことを示すカード、およびブロック・エンド・カードを最後に付けてやることを忘れてはならない。

かように考えるとメイン・プログラムは（図表 3-3）に示すようになる。

このメイン・プログラムについて IA とは採用している勘定の総数のことであり、IREV というのは損益計算表勘定の最初の勘定の番号である。すなわち、本稿のプログラムで複式簿記を実行するためには COMMON 宣言をそれぞれの勘定体系の実情に応じて訂正してやらなければならないことは勿論であるが、その他は、この IA と IREV を訂正するだけで、他は何等修

```

1      COMMON A(15, 2, 6), NAME(15), ICODE(15)
2      IA=15
3      IREV=8
4      DO 1 K=1, 6
5      DO 1 J=1, 2
6      DO 1 I=1, IA
7      1 A(I, J, K)=0.0
10     READ(5, 104) NAME
11     READ(5, 104) ICODE
12     READ(5, 104) MARK99
13 104 FORMAT(10A7)
14     2 READ(5, 101) K3
15 101 FORMAT(20X, I2)
16     READ(5, 101) NADD
17     IF(NADD. EQ. 0) GO TO 3
20     CALL NAMEAD(NADD)
21     3 CALL BENTRY(IA, MARK99, K3)
22     IF(K3. NE. 4) GO TO 2
23     CALL TBPLBS(IA, 1, IA-1, 1, 3)
24     CALL TBPLBS(IA, IREV, IA-2, 3, 5)
25     CALL TBPLBS(IA, 1, IREV-1, 3, 6)
26     WRITE(6, 102) (NAME(I), ((A(I, J, K), J=1, 2), K=1, 5), I=1, IA)
27 102 FORMAT(1H1, 9X, 16HZENKI KURI KOSHI, 5X,
              14HKONKI TORIHIKI, 9X, 7HZANDAK
30     1A, 11X, 11HSEIRI KINYU, 13X, 3HP/L/ (1H0, A7, 10F10.0))
31     WRITE(6, 103) (NAME(I), (A(I, J, 6), J=1, 2), I=1, IA)
32 103 FORMAT(1H1, 15X, 3HB/S/(1H0, A7, 2F10.0))
33     STOP
34     END

```

(図表 3 - 3) 決算を含む電子計算機複式簿記のメイン・プログラム

正する必要がないのである。

また、最後に出てくる決算表は（図表 3 - 3）の26行および31行の書出し命令を見れば明らかなように、一行百二十字のライン・プリンター用紙では一頁に入れると見にくくなることを恐れて、前期繰越高から損益勘定までを一頁に打出し、残高勘定は二頁目に書出してある。したがって、両者を横につないで見るのである。ただし付録として末尾に示した決算表は印刷の都合上一頁にまとめてある。

4. 基本的転記プログラム BENTRY と 勘定追加サブルーチン NAMEAD

(a) 基本的転記プログラム BENTRY

前述したように前掲拙稿「伝統的複式簿記とコンピューター」ではメイン・プログラムであったものを本稿ではサブルーチンとして使用するのであるから、いくらか修正しなければならない。

修正を完了したサブルーチン BENTRY は（図表4-1）に示すとおりであるから、どこが修正されたかは両者を比較すれば明らかである。すなわち、基本的転記プログラムをサブルーチンとして使用するために変化した部分は、メイン・プログラム（図表3-3）へ移った部分がほとんどで、個々のステートメントでその内容を修正した点は次の三箇所しかない。

すなわち、サブルーチン TEST をコールする

```
6 CALL TEST(N, M, NUMB2, HSUM1, HSUM2)
```

および

```
7 CALL TEST(N, M, NUMB2, HSUM1, HSUM2)
```

というステートメントのアーギュメントに K3 を追加してやることと、勘定配列の合計を算出する

```
8 A(IA, J)=A(IA, J)+A(I, J) というステートメントを
```

```
8 A(IA, J, K3)=A(IA, J, K3)+A(I, J, K3)
```

というように修正することだけである。

(b) 勘定追加サブルーチン NAMEAD

複式簿記の実行に際して、勘定科目を追加する必要が発生しない等ということはある。したがって、少しでも現実的な電子計算機複式簿記のプログラムを作成しようとするれば、勘定科目の追加および削除を正しく処理しうるプログラムでなければならない。

```

1   SUBROUTINE BENTRY(IA, MARK99, K3)
2   COMMON A(15, 2, 6), NAME(15), ICODE(15)
3   COMMON DCA(3, 2), IXY(3, 2, 2), CODE(3, 2), NZ(2)
4   LOGICAL CODE. CODE3. NZ
5   IB=IA-1
6   NUMB2=0
7   CALL PSET(N, M, HSUM1, HSUM2)
10  2 READ(5, 101) TA, ID, IC, NUMB
11  101 FORMAT(F6.0, 2A2, I3)
12  IF(TA. GE. 999999.0) GO TO 7
13  IF(NUMB2. EQ. 0) NUMB2=NUMB
14  IF(NUMB2. NE. NUMB) GO TO 6
15  3 IF(IC. NE. MARK99) GO TO 4
16  CALL TRNCHK(IA, N, TA, ID, HSUM1,1)
17  GO TO 2
20  4 IF(ID. NE. MARK99) GO TO 5
21  CALL TRNCHK(IA, M, TA, IC, HSUM2,2)
22  GO TO 2
23  5 CALL TRNCHK(IA, N, TA, ID, HSUM1,1)
24  CALL TRNCHK(IA, M, TA, IC, HSUM2,2)
25  GO TO 2
26  6 CALL TEST(K3, N, M, NUMB2, HSUM1, HSUM2)
27  NUMB2=NUMB
30  GO TO 3
31  7 CALL TEST(K3, N, M, NUMB2, HSUM1, HSUM2)
32  DO 8 J=1, 2
33  DO 8 I=1, IB
34  8 A(IA, J, K3)=A(IA, J, K3)+A(I, J, K3)
35  RETURN
36  END

```

(図表4-1)

基本的転記プログラムとしてのサブルーチン BENTRY

この最も単純な方法は、すでに前掲拙稿「伝統的複式簿記とコンピューター」の第三章二節「勘定コード検査の考え方」⁽³⁾においてなしたように、勘定配列の中に、将来の必要を考慮して若干の空白部分を設けておく方法である。

いま、前者で使用した勘定配列と勘定コードの関係をそのまま使用するこ

(3) 前掲拙稿 P. 44。

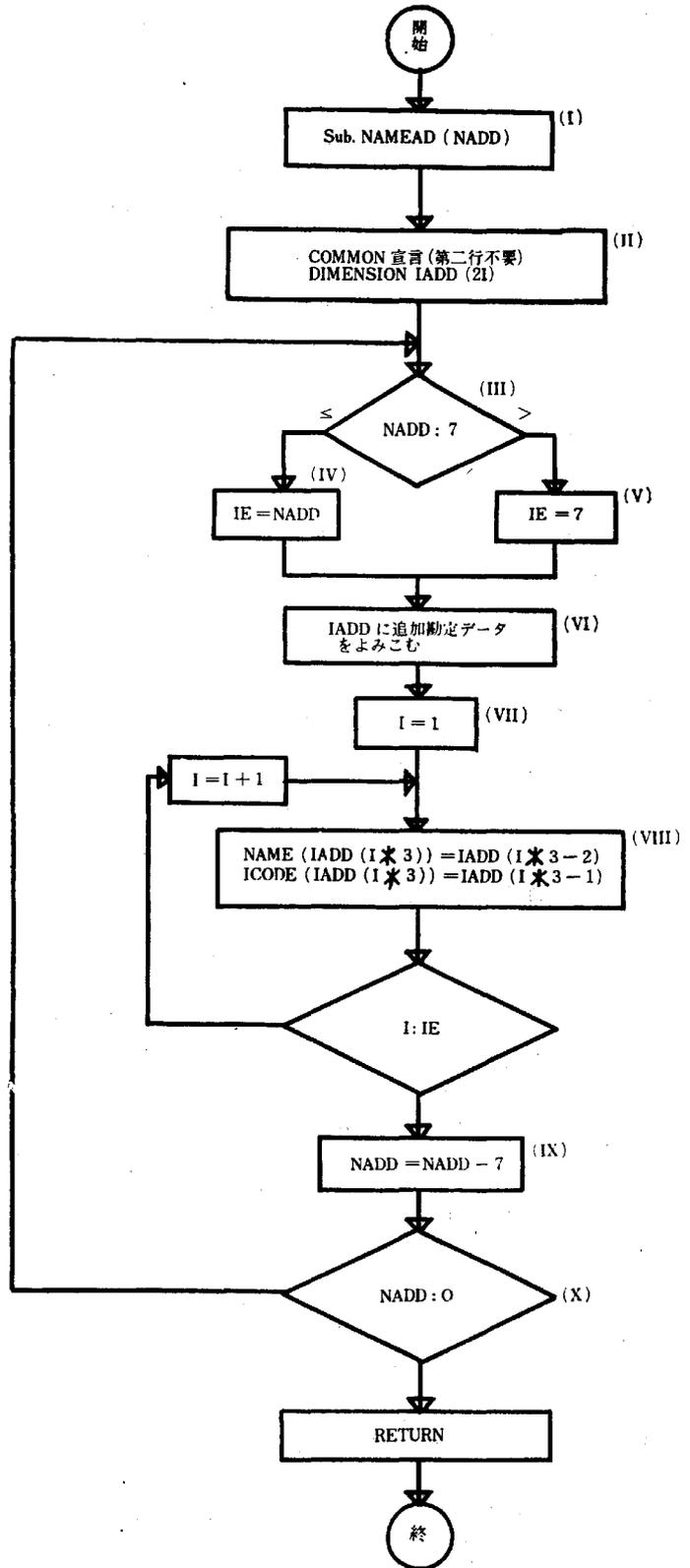
勘定	行数	勘定コード
現金勘定	1	A1
売掛金勘定	2	A2
備品勘定	3	A3
(余 裕)	4	(b)
買掛金勘定	5	L1
(余 裕)	6	(b)
資本金勘定	7	C1
売上勘定	8	R1
仕入勘定	9	E1
給料勘定	10	E2
営業費勘定	11	E3
(余 裕)	12	(b)
(余 裕)	13	(b)
(余 裕)	14	(b)
合 計 算 出 場 所	15	(b)

(図表4-2) 各勘定の割当

とにする。

もし、勘定を追加する必要がなければ制御カードの次におかれる勘定追加カードにゼロをパンチし直ちに仕訳カードを続ける。もし、勘定を追加する必要があるれば追加される勘定の数をパンチし、次にそれら追加される勘定の勘定名、勘定コード、勘定配列における行数等のデータを与えるカードをそれに続けるのである。

たとえば、前掲拙稿で使用した例で、商品の期末棚卸高が46,250円あり、売掛金について



(図表4-3)

サブルーチン NAMEAD (NADD) のフローチャート

貸倒引当金を10円設定することになると、次のような勘定を余裕行のなかに読込んでやらねばならない。

追加さるべき勘定	勘定名	勘定コード	行数
繰越商品勘定	INVT	A4	4
貸倒引当金勘定	KDHIKAT	L2	6
貸倒償却勘定	K·DAORE	E4	12
当期純損益	NET P/L	(なし)	14

なお、このとき、上図に示すように各勘定について三つの情報を与えてやらねばならぬことは言うまでもない。

このように考えると、勘定追加のためのサブルーチンのフロー・チャートは(図表4-3)のようになる。そして、サブプログラム NAMEADは(図表4-4)に示すようになる。

この間の論理をわかりやすくするために勘定を追加するためのデータが配列 IADD のなかにどのような形で収容されるかを図示すれば(図表4-5)のとおりである。

```

35C
36     SUBROUTINE NAMEAD(NADD)
37     COMMON A(15, 2, 6), NAME(15), ICODE(15)
40     DIMENSION IADD(21)
41     1 IF(NADD. GT. 7) GO TO 2
42     IE=NADD
43     GO TO 3
44     2 IE=7
45     3 READ(5, 101) IADD
46 101  FORMAT(7(A7, A2, I2))
47     DO 4 I=1, IE
50     NAME(IADD(I*3))=IADD(I*3-2)
51     4 ICODE(IADD(I*3))=IADD(I*3-1)
52     NADD=NADD-7
53     IF(NADD. LE. 0) RETURN
54     GO TO 1
55     END
    
```

(図表4-4) 勘定追加のためのサブルーチン NAMEAD

IADD(1) IADD(2) IADD(3) IADD(4) IADD(5) IADD(6) ……
 勘定名 勘定コード 位置 勘定名 勘定コード 位置

INVT	A4	4	KDHIKAT	L2	6	…… 省略 ……
繰越商品勘定のデータ 追加ナンバー I=1			貸倒引当金勘定のデータ 追加ナンバー I=2			省略 …… I=3 ……
IADD(I*3-2) → IADD(1)…INVT			IADD(I*3-2) → IADD(4)…KDHIKAT			
IADD(I*3-1) → IADD(2)… A4			IADD(I*3-1) → IADD(5)… L2			
IADD(I*3) → IADD(3)… 4			IADD(I*3) → IADD(6)… 6			

(図表 4-5) 追加される勘定についての情報が配列 IADD に格納される関係

5. 残高試算表，損益勘定，残高勘定作成の ためのサブルーチン TBPLBS

周知のように複式簿記においては残高試算表と損益勘定および残高勘定とはきわめて密接に関連し合っており，損益勘定と残高勘定は残高試算表の部分計算にすぎない。したがって，これら三者を演算処理するには一箇のサブルーチン TBPLBS (Trial Balance, Profit and Loss, and Balance Sheet)があれば足りるはずである。

(a) 勘定残高配列 A(I, J, 3) の演算と記入

直ちにサブルーチン TBPLBS の説明に入るまえに勘定残高配列の処理だけを行なうサブルーチンを考えてみよう。

勘定残高配列は前期繰越高配列と今期取引配列の借方合計と貸方合計を比較し，差額を勘定残高配列のしかるべき側に記入すればよい。したがって，いまこのサブルーチンを TB (Trial Balance) と名づけるとすると，この構造は（図表 5-1）に示すようになる。

この勘定残高配列への記入プログラムは実は損益勘定配列および残高勘定配列への記入プログラム的一部分となることは明らかである。なぜなら，損益勘定配列および残高勘定配列への記入もこの勘定残高配列の記入プログラ

```

SUBROUTINE TB(K)
COMMON ...省略...
IB=IA-1
DO 4 I=1, IB
ANET=(A(I, 1, 1)+A(I, 1, 2))-(A(I, 2, 1)+A(I, 2, 2))
IF(ANET) 1, 4, 2
1 J=2
GO TO 3
2 J=1
3 A(I, J, K)=ABS(ANET)
A(IA, J, K)=A(IA, J, K)+A(I, J, K)
4 CONTINUE
RETURN
END

```

(図表 5 - 1) 勘定残高配列のサブルーチン

ムと同じ仕事を行なわねばならず、異なる点は純損益を算出するという
 だけであるからである。

(b) 損益勘定配列と残高勘定配列の演算と記入を行なうサブルーチン

PLBS

いまかりに、損益勘定配列 $A(I, J, 5)$ と残高勘定配列 $A(I, J, 6)$ の二つ
 の勘定配列だけ処理するサブルーチン PLBS (Profit and Loss, Balance
Sheet) を考えると (図表 5 - 2) のようになる。

このサブルーチンの上半分は上述の勘定残高配列 $A(I, J, 3)$ を処理する
 ためのサブルーチン TB と基本的に同じであることは一見して明らかであ
 る。

両者の異なる点は、DO ループのパラメーターだけである。

サブルーチン PLBS の下半部の第一行目は言うまでもなく当期純利益ま
 たは当期純損失 ANPL (amount of net profit or loss) を算出しているの
 である。

この計算は伝統的な手記複式簿記に慣れ親んだ目から見ると少し異様であ

```

1   SUBROUTINE PLBS (IS, IL, K)
2   COMMON  ...省略...
3   DO 4 I=IS, IL
4   ANET=(A(I, 1, 3)+A(I, 1, 4))-(A(I, 2, 3)+A(I, 2, 4))
5   IF(ANET) 1, 4, 2
6   1 J=2
7   GO TO 3
10  2 J=1
11  3 A(I, J, K)=ABS(ANET)
12  A(IA, J, K)=A(IA, J, K)+A(I, J, K)
13  4 CONTINUE
-----
14  ANPL=A(IA, 1, K)-A(IA, 2, K)
15  IF(ANPL) 5, 8, 6
16  5 J1=1
17  J2=2
20  GO TO 7
21  6 J1=2
22  J2=1
23  7 A(IA-1, J1, K)=ABS(ANPL)
24  A(IA, J1, K)=A(IA, J2, K)
25  8 RETURN
26  END

```

(図表 5-2)

損益勘定配列と残高勘定配列の二つを処理するサブルーチン PLBS

るが、サブルーチン PLBS の上半部を実行したとき、すでに損益勘定配列の最下行（すなわち合計行）に借方合計額と貸方合計額が算出済であるから、両者の差額が純損益になるのである。このことは残高勘定配列の場合にもまったく同じである。

したがって、ANPL の正負を判定し、純利益が出ようと、あるいは純損失が出ようと、少ない方の下から二行目、すなわち慣行的な精算表の純損益算出場所にこの差額を加算してやればよい。また、その少ない方の最下行すなわち合計行の金額を大きい方の金額で置きかえてやれば、損益勘定配列の場合でも、残高勘定配列の場合でも正しく処理しうるのである。

(c) サブルーチン TBPLBS の完成

かように勘定残高配列を記入するプログラムと損益勘定配列および残高勘定配列を完成するプログラムとは基本的に同じものである。したがって、この両者を合せて一個のサブルーチン TBPLBS を作成してみよう。

そのためには、まず第一に（図表 5-2）の上半分と下半分の中間に

IF (K. EQ. 3) RETURN

というステートメントを入れなければならない。こおすると勘定残高配列の処理を行なうさい、純損益算出の仕事をしないうでメイン・プログラムに戻すことができる。

第二の修正点はアーギュメントを追加し、これによって、勘定残高配列の処理を行なう場合と損益・残高両勘定配列の処理を行なう場合に正しい ANET を算出できるようにしてやることである。

ANET の計算式は勘定残高配列の場合

$$\text{ANET} = (\text{A}(\text{I}, 1, 1) + \text{A}(\text{I}, 1, 2)) - (\text{A}(\text{I}, 2, 1) + \text{A}(\text{I}, 2, 2))$$

であり、損益勘定配列と残高勘定配列の場合

$$\text{ANET} = (\text{A}(\text{I}, 1, 3) + \text{A}(\text{I}, 1, 4)) - (\text{A}(\text{I}, 2, 3) + \text{A}(\text{I}, 2, 4))$$

である。したがって

$$\begin{aligned} \text{ANET} = & (\text{A}(\text{I}, 1, \text{IK}) + \text{A}(\text{I}, 1, \text{IK} + 1)) \\ & - (\text{A}(\text{I}, 2, \text{IK}) + \text{A}(\text{I}, 2, \text{IK} + 1)) \end{aligned}$$

のように修正すれば、いずれの場合にも使用できることになる。

したがって、サブルーチン TBPLBS のダミー・アーギュメントに IK を追加してやればよい。かくして勘定残高配列 $\text{A}(\text{I}, \text{J}, 3)$ 、損益勘定配列 $\text{A}(\text{I}, \text{J}, 5)$ および残高勘定配列 $\text{A}(\text{I}, \text{J}, 6)$ のいずれの場合でも共通に使用できるサブルーチン TBPLBS が（図表 5-3）のように完成するのである。

なお、このサブルーチン TBPLBS は前期繰越高がない場合でも正しく処理できることを念のため追記しておく。

```
SUBROUTINE TBPLBS(IA, IS, IL, IK, K)
COMMON A(15, 2, 6)
DO 4 I=IS, IL
ANET=(A(I, 1, IK)+A(I, 1, IK+1))-(A(I, 2, IK)+A(I, 2, IK+1))
IF(ANET)1, 4, 2
1 J=2
GO TO 3
2 J=1
3 A(I, J, K)=ABS(ANET)
A(IA, J, K)=A(IA, J, K)+A(I, J, K)
4 CONTINUE
IF(K. EQ. 3) RETURN
ANPL=A(IA, 1, K)-A(IA, 2, K)
IF(ANPL) 5, 8, 6
5 J1=1
J2=2
GO TO 7
6 J1=2
J2=1
7 A(IA-1, J1, K)=ABS(ANPL)
A(IA, J1, K)=A(IA, J2, K)
8 RETURN
END
```

(図表 5-3) 完成したサブルーチン TBPLBS

6. 決算を含む電子計算機複式簿記プログラムの 完成とその意義

以上の分析の他、本稿では使用する勘定配列が $A(I, J, 1)$ から $A(I, J, 6)$ の六つに増加したため、仕訳カードの種類も多くなっている。そこで、仕訳カードのエラーもその種類が増加することになる。しかし、これはエラーの種類が増加したというよりも、仕訳カードの種類が増加であるから、エラー・メッセージの中に、その仕訳がどの仕訳カード・ブロックに属するかを明らかにしてやればよい。このため、前掲拙稿「伝統的複式簿記とコンピューター」で示したサブルーチン ERROR を修正しておいた。

このような修正を施すと、決算を含む伝統的複式簿記は電子計算機により

完全に遂行することができ、またそのときの論理的構造は上述の分析の示す通りである。

このいわば完全な電子計算機複式簿記のプログラムをまとめて示すと（図表6-1）のようになる。

そして前掲拙稿以来使用してきた例について決算を行なわない場合のデータと演算結果は紙幅の都合で省略し、決算整理仕訳を含む場合の演算結果だけを付録一に示してある。

電子計算機複式簿記の決算の結果を精算表のような形式でなく、通常の複式簿記における損益勘定および残高勘定のような形でアウト・プットすることもできるし、また報告書形式で出すこともできる。しかし、こうしたアウト・プットの形式は単なるプログラミング技術の問題であり、電子計算機複式簿記の本質にはかかわりのない問題であるから論じない。

しかし、注意すべき点は、決算を行なわない場合のアウト・プットである。すなわち、付録一で行なった二つの決算整理仕訳がない場合には、決算を行なっていないにもかかわらず損益勘定配列と残高勘定配列に記入が行なわれ、しかも下から二行目に33,000円という数字があたかも純損失であるかの如く記入されることになるだろう。

これが純損失でないことは余りにも明瞭である。これは決算整理記入をしない場合の損益勘定配列の借方合計額と貸方合計額の差額を示しているにすぎない（残高勘定配列についても同じ）。

このようなことは伝統的複式簿記ではほとんど実行されることはない。したがって、決算整理記入がない場合には、損益勘定配列と残高勘定配列の記入を行なわないようにプログラムを修正した方が伝統的複式簿記により忠実であることになる。

しかし、これは次の重要な事実を示している。すなわち、毎日のように実地棚卸を行ない、ゾル・ベスタントとイスト・ベスタントの調整額を決定することが実行可能か否かという点を除いて考えても、電子計算機複式簿記で

はかように毎日決算を行なうことが少なくともコンピューターの性能上は可能であるし、また決算整理記入ぬきの仮りの数値ではあっても、財産・損益一覧表としての試算表の機能をかくの如く強化、拡充することができることを示している。

なおまた、配当政策その他の財務会計政策にもとづく価値判断を加えたものこそ真の意味の決算である。したがって、このような意味での決算は毎日とか毎週とか行なう必要もないし、行なったとしても無意味である。しかし、原価会計における月次決算のように管理会計的な決算であるならば、実はできるだけ頻繁に、しかも迅速に行なうことは重要な意義を持っている。

本稿のはじめに、電子計算機があらゆる企業のあらゆる勘定項目について自働的に実地棚卸を行なうわけにはゆかない、と書いた。しかし決算の意味を狭く限定し、試算表の持つ財産・損益一覧表という機能の拡充されたものとして解釈するならば、電子計算機複式簿記は「この限定された仮決算」を毎日行なうことができるという重要な発展を示しているのである。

何故なら、ほとんどすべての活動勘定項目について、EDPは「帳簿棚卸」を人手による場合よりも飛躍的に充実したものに行なうことができる。したがって、この電子計算機簿記の「帳簿棚卸」と人間による実地棚卸とをたくみに総合し、組織するならば、非財務会計的という限界内ではあるが、ゾル・ベスタントとイスト・ベスタントとの差額の調整という決算手続を容易に日常のルーチン・ワークにすることができるからである。

このように非財務会計的決算をいわば毎日毎日行ないうることこそ、実は電子計算機複式簿記の最も重要な特質の一つであり、そのことは同時に財産・損益一覧表としての試算表の意義が拡充され、毎日作成される「仮決算表」に転化するということにほかならない。そして、ここに電子計算機複式簿記がもたらす最も重要な変化の一つが存在するのである。

```

1C  SHO GAKU TOKYU VOL. 19, NO. 2.
2C  TRADITIONAL DOUBLE ENTRY BOOKKEEPING AND COMPUTER
3C
4   COMMON A(15, 2, 6), NAME(15), ICODE(15)
5   IA=15
6   IREV=8
7   DO 1 K=1, 6
10  DO 1 J=1, 2
11  DO 1 I=1, IA
12  1 A(I, J, K)=0.0
13  READ(5, 104) NAME
14  READ(5, 104) ICODE
15  READ(5, 104) MARK99
16  104 FORMAT(10A7)
17  2 READ(5, 101) K3
20  101 FORMAT(20X, I2)
21  READ(5, 101) NADD
22  IF(NADD. EQ. 0) GO TO 3
23  CALL NAMEAD(NADD)
24  3 CALL BENTRY(IA, MARK99, K3)
25  IF(K3. NE. 4) GO TO 2
26  CALL TBPLBS(IA, 1, IA-1, 1, 3)
27  CALL TBPLBS(IA, IREV, IA-2, 3, 5)
30  CALL TBPLBS(IA, 1, IREV-1, 3, 6)
31  WRITE(6, 102) (NAME(I), ((A(I, J, K), J=1, 2), K=1, 5), I=1, IA)
32  102 FORMAT(1H1, 9X, 16HZENKI KURI KOSHI, 5X,
                14HKONKI TORIHIKI, 9X, 7HZANDAK
                1A, 11X, 11HSEIRI KINYU, 13X, 3HP/L/(1H0, A7, 10F10.0))
33  WRITE(6, 103) (NAME(I), (A(I, J, 6), J=1, 2), I=1, IA)
35  103 FORMAT(1H1, 15X, 3HB/S/(1H0, A7, 2F10.0))
36  STOP
37  END
40C
41  SUBROUTINE NAMEAD(NADD)
42  COMMON A(15, 2, 6), NAME(15), ICODE(15)
43  DIMENSION IADD(21)
44  1 IF(NADD. GT. 7) GO TO 2
45  IE=NADD
46  GO TO 3
47  2 IE=7
50  3 READ(5, 101) IADD

```

(図表6-1) 決算を含む電子計算機複式簿記のプログラム (その1)

```
51 101 FORMAT(7(A7, A2, I2))
52     DO 4 I=1, IE
53     NAME(IADD(I*3))=IADD(I*3-2)
54     4 ICODE(IADD(I*3))=IADD(I*3-1)
55     NADD=NADD-7
56     IF(NADD. LE. 0) RETURN
57     GO TO 1
60     END
61C
62     SUBROUTINE BENTRY(IA, MARK99, K3)
63     COMMON A(15, 2, 6), NAME(15), ICODE(15)
64     COMMON DCA(3, 2), IXY(3, 2, 2), CODE(3, 2), NZ(2)
65     LOGICAL CODE, CODE3, NZ
66     IB=IA-1
67     NUMB2=0
68     CALL PSET(N, M, HSUM1, HSUM2)
69     2 READ(5, 101) TA, ID, IC, NUMB
70 101 FORMAT(F6.0, 2A2, I3)
71     IF(TA. GE. 999999.0)GO TO 7
72     IF(NUMB2. EQ. 0) NUMB2=NUMB
73     IF(NUMB2. NE. NUMB) GO TO 6
74     3 IF(IC. NE. MARK99) GO TO 4
75     CALL TRNCHK(IA, N, TA, ID, HSUM1, 1)
76     GO TO 2
77     4 IF(ID. NE. MARK99) GO TO 5
78     CALL TRNCHK(IA, M, TA, IC, HSUM2, 2)
79     GO TO 2
80     5 CALL TRNCHK(IA, N, TA, ID, HSUM1, 1)
81     CALL TRNCHK(IA, M, TA, IC, HSUM2, 2)
82     GO TO 2
83     6 CALL TEST(K3, N, M, NUMB2, HSUM1, HSUM2)
84     NUMB2=NUMB
85     GO TO 3
86     7 CALL TEST(K3, N, M, NUMB2, HSUM1, HSUM2)
87     DO 8 J=1, 2
88     DO 8 I=1, IB
89     8 A(IA, J, K3)=A(IA, J, K3)+A(I, J, K3)
90     RETURN
91     END
92 120C
93 SUBROUTINE TBPLBS(IA, IS, IL, IK, K)
```

(図表6-1) 決算を含む電子計算機複式簿記のプログラム (その2)

```

122     COMMON A(15, 2, 6)
123     DO 4 I=IS, IL
124     ANET=(A(I, 1, IK)+A(I, 1, IK+1))-(A(I, 2, IK)+A(I, 2, IK+1))
125     IF(ANET) 1, 4, 2
126     1 J=2
127     GO TO 3
130     2 J=1
131     3 A(I, J, K)=ABS(ANET)
132     A(IA, J, K)=A(IA, J, K)+A(I, J, K)
133     4 CONTINUE
134     IF(K. EQ. 3) RETURN
135     ANPL=A(IA, 1, K)-A(IA, 2, K)
136     IF(ANPL) 5, 8, 6
137     5 J1=1
140     J2=2
141     GO TO 7
142     6 J1=2
143     J2=1
144     7 A(IA-1, J1, K)=ABS(ANPL)
145     A(IA, J1, K)=A(IA, J2, K)
146     8 RETURN
147     END
150C
151     SUBROUTINE PSET(N, M, HSUM1, HSUM2)
152     COMMON A(15, 2, 6), NAME(15), ICODE(15)
153     COMMON DCA(3, 2), IXY(3, 2, 2), CODE(3, 2), NZ(2)
154     LOGICAL NZ
155     N=0
156     M=0
157     HSUM1=0.0
160     HSUM2=0.0
161     NZ(1)=.TRUE.
162     NZ(2)=.TRUE.
163     RETURN
164     END
165C
166     SUBROUTINE TRNCHK(IA, NM, TA, ICD, HSUM12, JA)
167     COMMON A(15, 2, 6), NAME(15), ICODE(15)
170     COMMON DCA(3, 2), IXY(3, 2, 2), CODE(3, 2), NZ(2)
171     LOGICAL CODE, NZ
172     NM=NM+1

```

(図表 6 - 1) 決算を含む電子計算機複式簿記のプログラム (その 3)

```
173   DCA(NM, JA)=TA
174   IXY(NM, JA, 2)=ICD
175   HSUM12=HSUM12+TA
176   IF(TA. LE. 0.0) NZ(JA)=.FALSE.
177   IF(IXY(NM, JA, 2). NE. ICODE(IA)) GO TO 2
200  1 CODE(NM, JA)=.FALSE.
201   IXY(NM, JA, 1)=0
202   RETURN
203  2 DO 3 J=1, IA
204   IF(J. EQ. IA) GO TO 1
205   IF(ICODE(J). NE. IXY(NM, JA, 2)) GO TO 3
206   CODE(NM, JA)=.TRUE.
207   IXY(NM, JA, 1)=J
210   RETURN
211  3 CONTINUE
212   END
213C
214   SUBROUTINE TEST(K3, N, M, NUMB2, HSUM1, HSUM2)
215   COMMON DUMMY(234), NZ(2)
216   LOGICAL CODE3, NZ, TESTS
217   CALL CODCHK(N, M, CODE3)
220   TESTS=.FALSE.
221   IF(ABS(HSUM1-HSUM2). LT. 0. 1. AND. NZ(1). AND. NZ(2))
                                                    TESTS=.TRUE.
222   IF(TESTS) GO TO 1
223   CALL ERROR(K3, N, M, NUMB2, HSUM1, HSUM2, 1)
224  1 IF(CODE3) GO TO 2
225   CALL ERROR(K3, N, M, NUMB2, HSUM1, HSUM2, 2)
226   GO TO 3
227  2 IF(TESTS) CALL ENTRY(N, M, K3)
230  3 CALL PSET(N, M, HSUM1, HSUM2)
231   RETURN
232   END
233C
234   SUBROUTINE CODCHK(N, M, CODE3)
235   COMMON A(15, 2, 6), NAME(15), ICODE(15)
236   COMMON DCA(3, 2), IXY(3, 2, 2), CODE(3, 2)
237   LOGICAL CODE, CODE3
240   DO 1 I=1, N
241   IF(CODE(I, 1)) GO TO 1
242   GO TO 3
```

```

243 1 CONTINUE
244 DO 2 I=1, M
245 IF(CODE(I, 2)) GO TO 2
246 GO TO 3
247 2 CONTINUE
250 CODE3=.TRUE.
251 RETURN
252 3 CODE3=.FALSE.
253 RETURN
254 END
255C
256 SUBROUTINE ENTRY(N, M, K3)
257 COMMON A(15, 2, 6), NAME(15), ICODE(15)
260 COMMON DCA(3, 2), IXY(3, 2, 2)
261 DO 1 I=1, N
262 1 A(IXY(I, 1, 1), 1, K3)=A(IXY(I, 1, 1), 1, K3)+DCA(I, 1)
263 DO 2 J=1, M
264 2 A(IXY(J, 2, 1), 2, K3)=A(IXY(J, 2, 1), 2, K3)+DCA(J, 2)
265 RETURN
266 END
267C
270 SUBROUTINE ERROR(K3, N, M, NUMB2, HSUM1, HSUM2, K)
271 COMMON A(15, 2, 6), NAME(15), ICODE (15)
272 COMMON DCA(3, 2), IXY(3, 2, 2), CODE(3, 2), NZ(2)
273 LOGICAL CODE, NZ
274 WRITE(6, 300) K3, NUMB2
275 GO TO (11, 12), K
276 11 WRITE(6, 101)
277 IF((N. EQ. 1). AND. (M. EQ. 1). AND. (.NOT. NZ(1)). AND.
      (.NOT. NZ(2))) GO TO 2
300 1 IF(N. EQ. 0) WRITE(6, 102)
301 IF(N. NE. 0) WRITE(6, 103) (DCA(I, 1), IXY(I, 1, 2), I=1, N)
302 IF(N. NE. 0) WRITE(6, 104) HSUM1
303 IF(M. EQ. 0) WRITE(6, 105)
304 IF(M. NE. 0) WRITE(6, 106) (DCA(I, 2), IXY(I, 2, 2), I=1, M)
305 IF(M. NE. 0) WRITE(6, 107) HSUM2
306 RETURN
307 2 IF(.NOT.((HSUM1. EQ. 0.0). AND.(HSUM2. EQ. 0.0))) GO TO 1
310 WRITE(6, 301) (IXY(1, J, 2), J=1, 2)
311 RETURN
312 12 WRITE(6, 201)

```

```
313 IF(N. EQ. 0) WRITE(6, 102)
314 IF(N. NE. 0) WRITE(6, 202) (DCA(I, 1), IXY(I, 1, 2),
                                CODE(I, 1), I=1, N)
315 IF(M. EQ. 0) WRITE(6, 105)
316 IF(M. NE. 0) WRITE(6, 203) (DCA(I, 2), IXY(I, 2, 2),
                                CODE(I, 2), I=1, M)
317 RETURN
320 300 FORMAT(1H0, 3HK3=, I3, 4X, 11HSHIWAKE NO., 2X, I3)
321 101 FORMAT(1H , 19HSHIWAKE NO MACHIGAI)
322 102 FORMAT(1H , 10X, 18HKARI KATA DATURAKU)
323 103 FORMAT(1H , 15X, F7.0, 2X, A2, 2X, 2H99)
324 104 FORMAT(1H , 10X, 16HKARI KATA GOKEI , F7.0)
325 105 FORMAT(1H , 10X, 19HKASHI KATA DATURAKU)
326 106 FORMAT(1H , 15X, F7.0, 2X, 2H99, 2X, A2)
327 107 FORMAT(1H , 10X, 16HKASHI KATA GOKEI, F7.0)
330 301 FORMAT(1H0, 6X, 18HTANJUN TORIHIKI NO/ , 7X,
              16HKINGAKU DATURAKU, 2X, A2
331 1, 2X, A2)
332 201 FORMAT(1H , 22HKANJO CODE NO MACHIGAI)
333 202 FORMAT(1H , 10X, 8HKARIKATA/(1H , 15X, F7.0, 2X, A2, 3X, L1))
334 203 FORMAT(1H , 10X, 10HKASHI KATA/(1H , 15X, F7.0, 2X, A2, 3X, L1))
335 END
336 END OF SOURCE*
```

（図表6-1）決算を含む電子計算機複式簿記のプログラム（その6）完

	ZENKI KURI KOSHI	KONKI TORIHIKI	ZANDAKA	SEIRI KINYU	P/L	B/S				
GENKIN	0	189000	154000	35000	0	0	35000	0		
URIKAKE	0	40000	39000	1000	0	0	0	1000	0	
BIHIN	0	68000	0	68000	0	0	0	68000	0	
INVT	0	0	0	0	46250	0	0	46250	0	
KAIKAKE	0	58000	60000	0	2000	0	0	0	2000	
KDHIKAT	0	0	0	0	0	10	0	0	10	
SHIHON	0	0	135000	0	135000	0	0	0	135000	
URIAGE	0	0	95000	0	95000	0	0	95000	0	
SHIRE	0	120000	0	120000	0	46250	73750	0	0	
KYURYO	0	5000	0	5000	0	0	5000	0	0	
EIGYOHI	0	3000	0	3000	0	0	3000	0	0	
K. DAORE	0	0	0	0	10	0	10	0	0	
	0	0	0	0	0	0	0	0	0	
NET P/L	0	0	0	0	0	13240	0	0	13240	
TOTAL	0	483000	483000	232000	232000	46260	95000	95000	150250	150250

(付録1) (図表6-1) に示したプログラムで決算を行なった結果