

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2015-179370

(P2015-179370A)

(43) 公開日 平成27年10月8日(2015.10.8)

(51) Int.Cl.	F 1	テーマコード (参考)
G06F 15/02 (2006.01)	G06F 15/02 315L	5B019
G09G 3/00 (2006.01)	G09G 3/00 R	5C080

審査請求 未請求 請求項の数 9 O L (全 19 頁)

(21) 出願番号 特願2014-56121 (P2014-56121)
 (22) 出願日 平成26年3月19日 (2014.3.19)

(71) 出願人 000001443
 カシオ計算機株式会社
 東京都渋谷区本町1丁目6番2号
 (74) 代理人 110001254
 特許業務法人光陽国際特許事務所
 (72) 発明者 村木 暁子
 東京都羽村市栄町3丁目2番1号 カシオ
 計算機株式会社 羽村技術センター内
 Fターム(参考) 5B019 HE04 HE09 HE19 HF07
 5C080 DD13 EE01

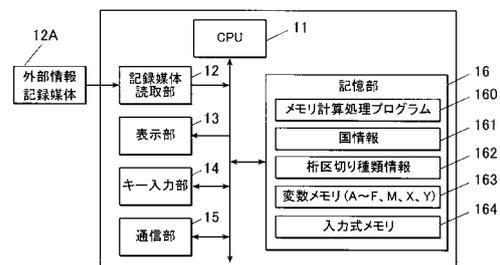
(54) 【発明の名称】 変数値表示制御装置及びプログラム

(57) 【要約】 (修正有)

【課題】ユーザが、数式の入力中に、全ての変数にそれぞれ設定されている変数値を一括して確認することが可能な変数値表示制御装置を提供する。

【解決手段】変数値表示制御装置は、複数の変数 A ~ F、M、X、Y について変数値を記憶するための変数値記憶手段 163 と、ユーザ操作に応じて、複数の変数のうちのいずれかの変数を指定して変数値記憶手段 163 に指定の変数の変数値を設定する変数値設定手段と、ユーザ操作に応じて、変数値記憶手段 163 に記憶された各変数の変数値を一覧表示させる変数値一覧表示手段と、ユーザ操作に応じて、一覧表示された変数のうちのいずれかの変数が指定されると、当該変数を入力する一覧変数入力手段とを備える。

【選択図】 図 2



【特許請求の範囲】**【請求項 1】**

複数の変数について変数値を記憶するための変数値記憶手段と、
ユーザ操作に応じて、前記複数の変数のうちのいずれかの変数を指定して前記変数値記憶手段に指定の変数の変数値を設定する変数値設定手段と、
ユーザ操作に応じて、前記変数値記憶手段に記憶された各変数の変数値を表示部に一覧表示させる変数値一覧表示手段と、
ユーザ操作に応じて、前記一覧表示された変数のうちのいずれかの変数が指定されると、当該変数を入力する一覧変数入力手段と、
を備えることを特徴とする変数値表示制御装置。

10

【請求項 2】

ユーザ操作に応じて入力された数式を表示する入力数式表示制御手段を備え、
前記入力数式表示制御手段により入力された数式を表示された状態で、前記変数値一覧表示手段により各変数の変数値を一覧表示された後に、一覧変数入力手段によりいずれかの変数が指定されると、前記入力された数式に当該変数を追加入力することを特徴とする請求項 1 に記載の変数値表示制御装置。

【請求項 3】

ユーザ操作に応じて入力された、前記追加入力された変数を含む数式を計算して計算結果を表示させる計算結果表示手段を備える請求項 2 に記載の変数値表示制御装置。

【請求項 4】

前記計算結果又は前記変数値を桁区切り有りで表示するか、桁区切り無しで表示するかをユーザが設定すると、それに応じて前記計算結果又は前記変数値を桁区切り有り又は桁区切り無しで表示させることを設定する桁区切り有無設定手段を備え、
前記変数値一覧表示手段は、前記桁区切り有無設定手段により設定された桁区切り有無の設定に応じて、各変数の変数値を一覧表示させることを特徴とする請求項 1 から請求項 3 のいずれか一項に記載の変数値表示制御装置。

20

【請求項 5】

前記計算結果又は前記変数値を桁区切り有りで表示させる際の桁区切りの文字又は桁区切りの位置を桁区切り種類として設定する桁区切り種類設定手段と、
国ごとに桁区切り種類を記憶した国別桁区切り種類記憶手段と、
ユーザ操作に応じて、いずれかの国を設定する国設定手段と、
を備え、
前記変数値一覧表示手段は、前記桁区切り有無設定手段により桁区切り有りが設定されている場合に、前記国設定手段により設定された国に対応する桁区切り種類で、各変数の変数値を一覧表示させることを特徴とする請求項 4 に記載の変数値表示制御装置。

30

【請求項 6】

前記計算結果又は前記変数値を桁区切り有りで表示させる際の桁区切りの文字又は桁区切りの位置を桁区切り種類として設定する桁区切り種類設定手段を備え、
前記桁区切り種類設定手段は、ユーザ操作により桁区切り種類を設定することを特徴とする請求項 4 に記載の変数値表示制御装置。

40

【請求項 7】

前記変数値一覧表示手段は、前記いずれかの変数値に複素数形式、分数形式、又は平方根形式の数値が含まれる場合に、各変数の変数値を一覧表示させる際に、前記桁区切り有無設定手段により設定された桁区切り有無の設定にかかわらず、複素数形式、分数形式、又は平方根形式の数値については桁区切り無しで当該変数を表示させることを特徴とする請求項 4 から請求項 6 のいずれか一項に記載の変数値表示制御装置。

【請求項 8】

前記表示部と、キー群を備え、前記キー群に対して前記ユーザ操作を行うことを特徴とする請求項 1 から請求項 7 のいずれか一項に記載の変数値表示制御装置。

【請求項 9】

50

複数の変数について変数値を記憶するための変数値記憶手段を備えるコンピュータに、ユーザ操作に応じて、前記複数の変数のうちのいずれかの変数を指定して前記変数値記憶手段に指定の変数の変数値を設定する変数値設定機能と、

ユーザ操作に応じて、前記変数値記憶手段に記憶された各変数の変数値を表示部に一覧表示させる変数値一覧表示機能と、

ユーザ操作に応じて、前記一覧表示された変数のうちのいずれかの変数が指定されると、当該変数を入力する一覧変数入力機能と、
を実現させることを特徴とするプログラム。

【発明の詳細な説明】

【技術分野】

10

【0001】

本発明は、少なくとも変数の数値を表示可能な変数値表示制御装置及びプログラムに関する。

【背景技術】

【0002】

従来、数式を入力して計算を行わせることが可能な関数電卓等の変数値表示制御装置においては、例えば、各変数の関係式を数理モデルとして入力する際に、ユーザが各変数の意味を確認するために、各変数の意味を一覧表示させることができるように構成された変数値表示制御装置が知られている（例えば特許文献1参照）。

【0003】

20

このような変数値表示制御装置では、例えば、変数の意味として駆動力や質量、弾性係数等の語句やルピーレート（すなわちインドルピーのレート）や元レート（すなわち中国元のレート）等の語句が画面上に一覧表示されるようになっている。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開2000-331036号公報

【発明の概要】

【発明が解決しようとする課題】

【0005】

30

しかしながら、このような変数値表示制御装置では、ユーザが一覧表示を見て各変数の意味を確認することはできるが、変数の具体的な数値を確認することができず、一覧表示では各変数の意味しか分からなかった。そして、各変数にどのような数値が設定されているかを確認するためには、ユーザが、変数を選択して数値を呼び出す処理を各変数について繰り返し行う必要があり、数値入力装置がユーザにとって必ずしも便利なものとなっていなかった。

【0006】

本発明は、上記の問題点を鑑みてなされたものであり、ユーザが、全ての変数にそれぞれ設定されている変数値を一括して確認することが可能な変数値表示制御装置及びプログラムを提供することを目的とする。

40

【課題を解決するための手段】

【0007】

以上の課題を解決するため、本発明の変数値表示制御装置は、
複数の変数について変数値を記憶するための変数値記憶手段と、
ユーザ操作に応じて、前記複数の変数のうちのいずれかの変数を指定して前記変数値記憶手段に指定の変数の変数値を設定する変数値設定手段と、

ユーザ操作に応じて、前記変数値記憶手段に記憶された各変数の変数値を表示部に一覧表示させる変数値一覧表示手段と、

ユーザ操作に応じて、前記一覧表示された変数のうちのいずれかの変数が指定されると、当該変数を入力する一覧変数入力手段と、

50

を備えることを特徴とする。

【発明の効果】

【0008】

本発明によれば、ユーザが、数式の入力中に、全ての変数にそれぞれ設定されている変数値を一括して容易かつ的確に確認することが可能となる。

【図面の簡単な説明】

【0009】

【図1】変数値表示制御装置の一例としての関数電卓の概観を示す平面図である。

【図2】関数電卓の内部構成を示すブロック図である。

【図3】関数電卓における処理の流れを表すフローチャートである。

10

【図4】操作及びディスプレイ上の表示内容を示す図である。

【図5】(A)桁区切り種類情報や(B)国情報を表す図であり、(C)桁区切り無し、(D)桁区切り有りに表示する状態、及び(E)複素数形式等の数値が含まれる場合には桁区切り無しで表示させることを表す図である。

【図6】操作及びディスプレイ上の表示内容を示す図である。

【図7】操作及びディスプレイ上の表示内容を示す図である。

【発明を実施するための形態】

【0010】

以下、図面を参照して、本発明に係る変数値表示制御装置を関数電卓に適用した場合の実施形態について詳細に説明する。ただし、発明の範囲は、図示例に限定されない。

20

【0011】

[外観構成]

図1は、本実施形態に係る変数値表示制御装置としての関数電卓1の平面図である。図1に示すように、関数電卓1は、各種キー群を有する入力キー群2と、ディスプレイ10とを備えている。

【0012】

入力キー群2は、ユーザから数値や計算記号等の数式構成要素の入力操作を受けたり、各種処理の指示操作を受けたりするためのキー群であり、それぞれ固有の機能を割り当てられた複数のキーを備えている。本実施形態においては、入力キー群2は、テンキー20やカーソルキー21、MODEキー22、SHIFTキー23、記号入力キー24、ALPHAキー25、変数キー26、STOキー27等を備えている。

30

【0013】

このうち、テンキー20は数値の入力操作を受けるキーであり、カーソルキー21は、ディスプレイ10内で編集対象位置や選択対象位置を示すカーソルを所定の方向に移動させる場合等に押下されるキーであり、本実施形態では、上下左右の4方向について入力可能に構成されている。

【0014】

MODEキー22は、ユーザが種々のモードを設定したり変更したりする際に押下されるキーであり、これについては後で説明する。SHIFTキー23は、各キーの機能を当該キーの面上に記載された機能からその上側に記載された機能に変更する(或いはその逆)際に押下されるキーである。また、記号入力キー24は、加減乗除の記号のほか、logや等の数式の入力に必要な記号や式等を入力するためのキーである。

40

【0015】

ALPHAキー25は、A~F、M、X、Yのアルファベットで表される各変数を入力する際に押下されるキーであり、本実施形態ではA~F、M、X、Yのアルファベットで表される各変数を入力する際に、ALPHAキー25を押下して当該アルファベットに対応する変数キー26を押下することで、入力中の数式に変数としての当該アルファベットを入力することができるようになっている。

【0016】

なお、本実施形態では、MODEキー22を押下した後、所定のテンキー20を押下すると

50

、計算モードを、マトリクスモード（図1の「MATRIX」参照）やベクトルモード（「VECTOR」参照）、複素数モード（「CMLPX」参照）等に変えることができるようになっている。

【0017】

SHIFTキー24は、当該SHIFTキー23の押下後に入力キー2が押下されると、当該入力キー2の上に表示されている内容が入力されるようになっている。例えば、SHIFTキー24を押下された後に「sin」キーが押下されると、「sin」キーの上に表示されている「sin⁻¹」が入力される。

【0018】

STOキー27は、ユーザがある変数について変数値を設定する際に押下されるキーであり、例えば、ユーザが「1」、「0」、「STO」、「A」と各キーを押下すると、ディスプレイ10上に「10 A」と表示されて、変数Aに10が設定される。なお、本実施形態では、STOキー27の後に変数キー26の1つである（-）キーを押下すると、自動的に変数「A」を入力することができるようになっているが、STOキー27の後にALPHAキー25を押下してから（-）キーを押下して変数「A」を入力するように構成することも可能である。

【0019】

また、STOキー27の上側に記載されている「RCL」（Recall）機能は、SHIFTキー23を押下された後にSTOキー27が押下されると実行され、記憶されている各変数の変数値を呼び出す機能（Recall機能。図1参照）である。Recall機能を行うことで、後述する記憶部16に記憶された各変数（A～F、M、X、Y）の変数値がディスプレイ10上に一覧表示されるようになっているが、この点については後で説明する。

【0020】

「=」キーである実行キー28は、ユーザが数式の入力後に計算処理の実行を指示する際に押下されるキーである。

【0021】

ディスプレイ10は、LCD（Liquid Crystal Display）やELD（Electronic Luminescent Display）等により構成されており、入力キー群2などの操作に応じた文字や符号、記号、式、計算結果等の各種データを複数のドットにより表示できるようになっている。なお、ディスプレイ10に、タッチパネルを、例えば表示画面全面に亘って一体的に設けることも可能である。

【0022】

[内部構成]

続いて、関数電卓1の内部構造について説明する。図2は、関数電卓1の内部構成を示すブロック図である。図2に示すように、関数電卓1は、記録媒体読取部12と、表示部13と、キー入力部14と、通信部15と、記憶部16と、CPU（Central Processing Unit）11とを備えて構成されている。

【0023】

記録媒体読取部12は、着脱自在に装着されるUSBメモリ等の外部情報記録媒体13Aから情報を読み取るものである。

【0024】

表示部13は、上述のディスプレイ10を備えており、CPU11からの表示信号に従って各種情報をディスプレイ10に表示できるようになっている。

【0025】

キー入力部14は、前述した入力キー群2（前述したようにタッチパネルを設ける場合には入力キー群2とタッチパネル）を備えており、ユーザ操作により入力されたキーに対応するキー入力信号をCPU11に出力する。CPU11は、ユーザ操作により入力されたキーに対応するキー入力信号を受け付けて対応する数式を表示部に表示したり、計算実行する。

【0026】

10

20

30

40

50

通信部 15 は、図示しないネットワークに接続可能とされており、これにより、ネットワークに接続される外部機器（例えばサーバやコンピュータ等）との通信が可能となっている。

【0027】

記憶部 16 は、関数電卓 1 の各種機能を実現するためのプログラムやデータ等を記憶するとともに、CPU 11 の作業領域として機能するメモリである。本実施形態では、記憶部 16 は、本発明に係るプログラムであるメモリ計算処理プログラム 160 の記憶領域を備えている。メモリ計算処理プログラム 160 は、後述のメモリ計算処理（図 3 参照）を CPU 11 に実行させるためのプログラムである。

【0028】

また、記憶部 16 は、国情報 161 の記憶領域と、桁区切り種類情報 162 の記憶領域等も備えている。これらについては後で説明する。記憶部 16 は、さらに、変数 A ~ F、M、X、Y を記憶する記憶領域（すなわち変数メモリ 163）や、入力された数式を記憶する記憶領域（すなわち入力式メモリ 164）も備えている。すなわち、本実施形態では、記憶部 16 が、複数の変数について変数値を記憶するための変数値記憶手段として機能するようになっている。

【0029】

CPU 11 は、関数電卓 1 の各部を中央制御する。具体的には、CPU 11 は、記憶部 16 の各記憶領域に記憶されているシステムプログラムや各種アプリケーションプログラムの中から指定されたプログラムを読み出して記憶部 16 の作業領域に展開し、記憶部 16 に展開されたプログラムとの協働で、各種処理を実行するようになっている。また、CPU 11 は、表示部 13 を制御して、ディスプレイ 10 上に必要な表示を行わせるようになっている。

【0030】

[動作]

次に、本実施形態に係る変数値表示制御装置である関数電卓 1 における動作について、図 3 に示すフローチャートに基づいて他の図面を参照しつつ説明する。また、本実施形態に係る変数値表示制御装置である関数電卓 1 の作用についてもあわせて説明する。

【0031】

なお、以下で説明する変数値表示制御装置（関数電卓 1）の動作は、図 2 に示したメモリ計算処理プログラム 160 に従って行われるため、以下の説明は、本発明に係るプログラムであるメモリ計算処理プログラム 160 についての説明にもなっている。

【0032】

また、図 3 のフローチャートでは説明を省略するが、公知の関数電卓と同様に、本実施形態においても、ユーザが前述したカーソルキー 21 を操作することで、ディスプレイ 10 上のカーソルの位置を適宜移動させることができるようになっている（例えば後述する図 4（G）参照）。また、ユーザ操作により入力された数式は、随時、記憶部 16 の入力された数式を記憶する記憶領域（すなわち入力式メモリ 164。図 2 参照）に記憶されるようになっている。

【0033】

本発明に特徴的な変数 Recall 一括表示処理、すなわち Recall 操作以降の処理（ステップ S1 ~ S7）については後で説明することとし、まず、公知の関数電卓でも行われる一般的な操作や処理について説明する。なお、ユーザにより、以下で説明する操作以外の操作が行われた場合には（ステップ S23；YES）、関数電卓 1 の CPU 11 は、当該操作に対応する他の処理に移行し、図 3 のフローチャートに示したルーチンを終了する。

【0034】

また、以下では、図 4（A）~（J）に示す複素数モードでの具体例に沿って説明する。この場合、前述したように、本実施形態では、ユーザは、MODE キー 22（図 1 参照）を押下した後、「2」のテンキー 20 を押下することで、数式の入力モードを複素数入力（図 1 の「CMPLX」参照）に変える。

10

20

30

40

50

【 0 0 3 5 】

[データ入力処理]

関数電卓 1 の CPU 1 1 は、テンキー 2 0 や記号入力キー 2 4 が押下されたり、SHIFT キー 2 3 が押下された状態で記号入力キー 2 4 が押下されたり、或いは ALPHA キー 2 5 が押下された後で変数キー 2 6 がされる（すなわち A ~ F、M、X、Y のいずれかの変数が入力される）等して文字（変数）や符号、記号、式等の各種データが入力されると（ステップ S 1 3 ; Y E S）、数式入力中か否かを判断する（ステップ S 1 4）。

【 0 0 3 6 】

そして、数式入力中であれば（ステップ S 1 4 ; Y E S）、入力されたデータ（文字、符号、記号、式等）を、入力中の数式に追加して表示させる処理を行う（ステップ S 1 5）。また、数式入力中でなければ（ステップ S 1 4 ; N O）、入力されたデータを新規入力データとしてディスプレイ 1 0 上に表示させる処理を行う（ステップ S 1 6）。

10

【 0 0 3 7 】

なお、その際、例えば図 4 (G) に示すように、ユーザによりディスプレイ 1 0 上のカーソルの位置が移動された場合には、関数電卓 1 の CPU 1 1 は、入力されたデータを、入力中の数式の、ディスプレイ 1 0 上に表示されているカーソルの位置に追加して表示させる（ステップ S 1 5）。このようにして、入力した数式の修正等を適宜行うことができるようになっている。

【 0 0 3 8 】

[実行処理]

また、関数電卓 1 の CPU 1 1 は、「 = 」キーである実行キー 2 8 が押下されて実行操作が行われると（ステップ S 1 7 ; Y E S）、入力された数式の計算を実行する処理を行う（ステップ S 1 8）。そして、計算結果をディスプレイ 1 0 上に表示させる処理を行う（ステップ S 1 9）。

20

【 0 0 3 9 】

すなわち、本実施形態では、CPU 1 1 は、ユーザ操作に応じて入力された、前記変数を含む数式を計算して計算結果を表示させる計算結果表示手段として機能するようになっている。なお、本実施形態では、CPU 1 1 は、「 = 」キーである実行キー 2 8 が押下されて実行操作が行われると、例えば図 4 (E) や図 4 (H) に示すように、ディスプレイ 1 0 の右下の領域に計算結果を表示させるようになっている。

30

【 0 0 4 0 】

[Store 処理]

また、関数電卓 1 の CPU 1 1 は、Store 操作がなされると（ステップ S 8 ; Y E S）、次に、変数キー 2 6 が押下されたか否かを判断し（ステップ S 9）、変数キー 2 6 が押下されずに（ステップ S 9 ; N O）、他のキーが押下された場合には当該キーに対応する処理を行う。

【 0 0 4 1 】

また、Store 操作がなされ、変数キー 2 6 が押下された場合（ステップ S 9 ; Y E S）、数式入力中でなければ（ステップ S 1 0 ; N O）、CPU 1 1 は、計算結果の数値を指定された変数 X の変数値として設定し、ディスプレイ 1 0 上に「 A n s X 」と結果数値とを表示させる（ステップ S 1 2）。

40

【 0 0 4 2 】

具体的には、例えば図 4 (A) に示すように、「 2 」のテンキー 2 0 が押下された後、STO キー 2 7 が押下されて [A] の変数キー 2 6 が押下されると、CPU 1 1 は、入力された数値「 2 」を指定された変数「 A 」に変数値として変数値記憶手段に設定する。なお、図 4 (A) に示すように、本実施形態では、設定された変数値がディスプレイ 1 0 の右下の領域に表示されるようになっている。また、同様に、「 2 」のテンキー 2 0 が押下された後、STO キー 2 7 が押下されて [B] の変数キー 2 6 が押下されると、CPU 1 1 は入力された数値「 2 」を指定された変数「 B 」に変数値として変数値記憶手段に設定する。

50

【 0 0 4 3 】

また、同様に、図 4 (B) に示すように、「 3 」のテンキー 2 0 が押下された後、STO キー 2 7 が押下されて [C] の変数キー 2 6 が押下されると、CPU 1 1 は、入力された数値「 3 」を指定された変数「 C 」に変数値として変数値記憶手段に設定する。そして、例えば、以上のように設定することで、図 4 (C) に示すように、変数 A、B、C にはそれぞれ変数値 2、2、3 が設定される。なお、図 4 (C) 等に示す一覧表示については後で説明する。

【 0 0 4 4 】

そして、Store操作がなされ、変数キー 2 6 が押下された場合 (ステップ S 9 ; Y E S)、数式入力中でなく、数式の入力後であれば (ステップ S 1 0 ; N O)、計算結果の数値を指定された変数の変数値として設定する処理を行うようになっている (ステップ S 1 2)。

10

【 0 0 4 5 】

具体的には、例えば図 4 (D) に示すようにある数式が入力され、図 4 (E) に示すように「 = 」キーである実行キー 2 8 が押下されて実行操作が行われると (ステップ S 1 7 ; Y E S)、CPU 1 1 は、変数値記憶手段から数式中の各変数の変数値を呼び出して数式の計算を実行し (ステップ S 1 8)、計算結果をディスプレイ 1 0 上に表示させる (ステップ S 1 9)。

【 0 0 4 6 】

そして、STO キー 2 7 が押下されて (ステップ S 8 ; Y E S)、[X] の変数キー 2 6 が押下されると (ステップ S 9 ; Y E S)、CPU 1 1 は、数式は実行操作済みであり、入力中ではないから (ステップ S 1 0 ; N O)、計算結果の数値を指定された変数 X の変数値として設定するとともに、図 4 (F) に示すように、ディスプレイ 1 0 の左上の領域に「 A n s X 」と表示させ、ディスプレイ 1 0 の右下の領域に設定された変数値を表示させる (ステップ S 1 2)。

20

【 0 0 4 7 】

図 4 (G) ~ (I) に示すように、数式が修正され (図 4 (G) 参照)、実行操作 (図 4 (H) 参照) が行われた場合も同様に、その後、STO キー 2 7 が押下されて [Y] の変数キー 2 6 が押下されると、CPU 1 1 は、修正された数式の計算結果の数値を指定された変数 Y の変数値として設定する (ステップ S 1 2) とともに、図 4 (I) に示すように、ディスプレイ 1 0 の左上の領域に「 A n s Y 」と表示させ、ディスプレイ 1 0 の右下の領域に設定された変数値を表示させる。

30

【 0 0 4 8 】

このように設定処理が行われることで、図 4 (J) に示すように、変数 X、Y にはそれぞれ図 4 (D) の数式や図 4 (G) の修正された数式の計算結果がそれぞれ変数値として設定される。なお、この場合、変数 A、B、C にはそれぞれ変数値 2、2、3 が既に設定されている。また、図 4 (J) 及び後述する図 5 (E) 中の「 / 」は、除算を意味する記号であり、 / と同じ意味で用いられている。

【 0 0 4 9 】

このように、本実施形態では、CPU 1 1 は、ユーザ操作に応じて、複数の変数のうちのいずれかの変数を指定して、変数値記憶手段である記憶部 1 6 の変数を記憶する記憶領域 (変数メモリ 1 6 3。図 2 参照) に、指定された変数の変数値を設定する変数値設定手段として機能するようになっている。

40

【 0 0 5 0 】

一方、Store操作がなされ (ステップ S 8 ; Y E S)、変数キー 2 6 が押下された場合 (ステップ S 9 ; Y E S)、数式入力中であれば (ステップ S 1 0 ; Y E S)、CPU 1 1 は、入力された数式の計算を実行し、ディスプレイ 1 0 上に「入力中の数式 変数」と計算結果を表示させ、当該変数の変数値を記憶させるようになっている (ステップ S 1 1)。

【 0 0 5 1 】

50

具体的には、図示を省略するが、例えば図4(A)、(B)に示したように、変数A、B、Cの変数値として2、2、3が設定された状態で、例えば「A+B+C」の数式を入力した時点で、実行操作を行わずに、STOキー27が押下されてStore操作がなされ(ステップS8;YES)、例えば「X」の変数キー26が押下されると(ステップS9;YES)、CPU11は、この場合は、数式の入力中であるから(ステップS10;YES)、変数値記憶手段(すなわち変数メモリ163)から変数A、B、Cの各変数値2、2、3を呼び出して数式の計算を実行し、「A+B+C X」を表示させて、計算結果である「7」をディスプレイ10の例えば右下の領域に表示させるとともに、変数Xに結果数値7を変数値として記憶させる(ステップS11)。

【0052】

[変数Recall一括表示処理]

次に、本発明に特徴的な変数Recall一括表示処理について説明する。

【0053】

上記のようにして、ユーザ操作により、数式が入力されたり、変数の変数値が設定される等している最中に、例えば図4(C)に示すように、SHIFTキー23が押下されてSTOキー27が押下されると、すなわちRecall操作が行われると(ステップS1;YES)、関数電卓1のCPU11は、変数Recall一括表示処理を行うようになっている。

【0054】

すなわち、CPU11は、Recall操作が行われると、変数値記憶手段である記憶部16の変数メモリ163(図2参照)から設定されている各変数A~F、M、X、Yの変数値を呼び出して、図4(C)に示すように、ディスプレイ10上に、各変数の変数値を一覧表示させるようになっている(ステップS2)。

【0055】

このように、本実施形態では、CPU11は、数式の入力中に、ユーザ操作に応じて、変数値記憶手段(すなわち記憶部16の変数メモリ163)に記憶された各変数の変数値を一覧表示させる変数値一覧表示手段として機能するようになっている。なお、図4(C)では図4(A)、(B)に示した操作が行われた後の各変数の変数値が一覧表示される場合が示されている。すなわち、変数A~Cについては2、2、3が設定されているが、変数D~F、M、X、Yについては変数値が設定されていないため、それぞれ0が設定された状態になっている。

【0056】

このように、数式の入力中や変数の変数値の設定中に、各変数の変数値を一覧表示させることができるように構成することで、ユーザは、数式の入力や変数の変数値の設定の最中に、各変数の変数値を1つ1つ確認する処理を繰り返したり、画面をスクロールしたり、或いは画面送り等をすることなく、各変数の変数値の一覧表示を見て各変数にどのような数値が入力されているかを一度に確認することが可能となる。

【0057】

このように、本実施形態に係る変数値表示制御装置(関数電卓1)によれば、ユーザは、全ての変数にそれぞれ設定されている変数値を、一括して、容易かつ的確に確認することが可能となる。

【0058】

一方、このように各変数の変数値がディスプレイ10上に一覧表示された状態で、ユーザが、変数キー26を押下せずに(ステップS3;NO)、別の操作を行った場合には、ユーザが、各変数の変数値の一覧表示を見て所望の変数の変数値を確認することができたものと考えられるため、本実施形態では、CPU11は、ディスプレイ10上の表示を上記のRecall操作が行われる前の表示状態に戻すようになっている(ステップS4)。

【0059】

また、各変数の変数値がディスプレイ10上に一覧表示された状態で、ユーザが、続けて、変数キー26を押下した場合(ステップS3;YES)、各変数の変数値をディスプレイ10上に一覧表示させる直前の状態が数式の入力中の状態であった場合には(ステッ

10

20

30

40

50

プ S 5 ; Y E S)、所望の変数の変数値を確認したユーザが、引き続き数式の入力を行うために当該変数を指定したものと考えられる。

【 0 0 6 0 】

そのため、本実施形態では、この場合 (ステップ S 5 ; Y E S)、CPU 1 1 は、変数キー 2 6 が押下されることにより指定された変数を入力中の数式に追加入力する処理を行って (ステップ S 6)、数式の入力処理を継続するようになっている。このように、本実施形態では、CPU 1 1 は、変数値一覧表示手段 (CPU 1 1 自身) により表示された変数のうちのいずれかの変数が指定されると、当該変数を、入力中の数式に追加入力する一覧変数入力手段として機能するようになっている。

【 0 0 6 1 】

また、各変数の変数値がディスプレイ 1 0 上に一覧表示された状態で、ユーザが、続けて、変数キー 2 6 を押下した場合 (ステップ S 3 ; Y E S)、各変数の変数値をディスプレイ 1 0 上に一覧表示させる直前の状態が、数式の入力終了の状態、すなわち数式の実行操作が行われた状態である場合には (ステップ S 5 ; N O)、ユーザは、計算結果をある変数の変数値として設定したいが、どの変数に設定すればよいかを確認するために各変数の変数値の一覧表示を見たものと考えられる。

【 0 0 6 2 】

そのため、本実施形態では、この場合 (ステップ S 5 ; N O)、CPU 1 1 は、計算結果の数値を指定された変数の変数値として、変数値記憶手段である記憶部 1 6 の変数を記憶する記憶領域 (変数メモリ 1 6 3。図 2 参照) に設定するとともに、変数値を表示する処理を行うようになっている (ステップ S 7)。

【 0 0 6 3 】

このように構成することで、ユーザの意図に沿って、数式の入力操作が継続されたり、計算結果が所定の変数の変数値として設定されたりするようになるため、本実施形態における変数値表示制御装置である関数電卓 1 が、ユーザにとって使い勝手が良いものとなる。

【 0 0 6 4 】

[効果]

以上のように、本実施形態に係る関数電卓 1 (変数値表示制御装置) やプログラムによれば、CPU 1 1 を、ユーザ操作に応じて、複数の変数 A ~ F、M、X、Y のうちのいずれかの変数を指定して、記憶部 1 6 (変数値記憶手段) に指定の変数の変数値を設定し (変数値設定手段としての機能)、ユーザ操作に応じて入力された、変数を含む数式を計算して計算結果をディスプレイ 1 0 上に表示させるとともに (計算結果表示手段としての機能)、数式の入力中に、ユーザ操作に応じて、変数値記憶手段に記憶された各変数の変数値を一覧表示させ (変数値一覧表示手段としての機能)、表示された変数のうちのいずれかの変数が指定されると、当該変数を、入力中の数式に追加入力する (一覧変数入力手段としての機能) ように構成した。

【 0 0 6 5 】

そのため、ユーザは、数式の入力や変数の変数値の設定の最中に、各変数の変数値を 1 つ 1 つ確認する処理を繰り返したり、画面をスクロールしたり、或いは画面送り等を行うことなく、各変数の変数値の一覧表示を見て各変数にどのような数値が入力されているかを一度に確認することが可能となり、全ての変数にそれぞれ設定されている変数値を、一括して、容易かつ的確に確認することが可能となる。

【 0 0 6 6 】

[桁区切り有無設定処理]

次に、桁区切り有無設定処理について説明する。

【 0 0 6 7 】

例えば、上記のように、ディスプレイ 1 0 上に、各変数の変数値を一覧表示させたり (ステップ S 2) 或いは計算結果を表示したり (ステップ S 1 9) する場合、表示される変数値や計算結果等の数値の桁数が大きいと、ユーザが、数値が何桁なのかを容易に把握す

10

20

30

40

50

ることができず、数値が読み取りづらくなる場合がある。

【0068】

そのため、本実施形態に係る関数電卓1では、そのような場合に、計算結果や変数値等の数値を、桁区切り有りで表示するか、或いは桁区切り無しで表示するかをユーザが設定することができるようになっている（ステップS20）。

【0069】

具体的には、本実施形態では、関数電卓1の記憶部16（図2参照）には、前述したように、国情報161と桁区切り種類情報162とがそれぞれ予め記憶されている。数値の桁を区切る際、国によって桁区切り種類（すなわち桁の区切り方）が異なる。そのため、本実施形態では、桁区切り種類情報162として、例えば図5（A）に示すように、桁をコンマ（Comma）で区切る区切り方（この場合、小数点はドットで表される。）や、ドット（Dot）で区切る区切り方（この場合、小数点は逆にコンマで表される。）、スペース（Space）を開けて区切る区切り方（この場合、小数点以下もスペースで区切る場合（Space(DN)参照）と小数点以下は区切らない場合（Space参照）がある。）が予め設定されている。

10

【0070】

また、インドでは、例えば「3,25,84,729」のように、百の位と千の位との間や、万の位と十万の位との間、百万の位と千万の位との間等にコンマを入れるようにして数値の桁が区切られるため、本実施形態では、桁区切り種類情報162として、図5（A）に示すように、インド（India）方式の桁区切り種類も予め設定されている。また、数値を桁区切り無しで表示することを選択するために、桁区切り種類情報162として、「区切りなし」の選択肢も設定されている。

20

【0071】

また、上記のように、数値の桁を区切る際、国によって桁区切り種類が異なるため、本実施形態では、図5（B）に示すように、国情報161として、国名と、桁区切り有りで表示することが設定された場合の当該国での桁区切り種類とを対応付ける情報が、例えばテーブル等の形で、予め記憶部16に設定されている。

【0072】

そして、例えば、上記のように、ディスプレイ10上に、各変数の変数値を一覧表示させる際に（ステップS2）、各変数の変数値を桁区切り無しで表示するように設定されている場合には、関数電卓1のCPU11は、例えば図5（C）に示すように、各変数A～F、M、X、Yの変数値をそれぞれ桁区切り無しでディスプレイ10上に表示させる。また、各変数の変数値を桁区切り有りで表示するように設定されており、例えば、その際の桁区切り種類としてインド（India）方式が設定されている場合には、例えば図5（D）に示すように、各変数A～F、M、X、Yの変数値をそれぞれ桁区切り無しでディスプレイ10上に表示させるようになっている。

30

【0073】

なお、この場合、一覧表示させる各変数の変数値のうちのいずれかに複素数形式や分数形式、平方根形式の数値が含まれる場合に、変数値の数値を桁区切りして表示すると、数値がかえって見づらくなる。そのため、本実施形態では、例えば図5（E）に示すように、一覧表示させる各変数の変数値のうちのいずれかに複素数形式や分数形式、平方根形式の数値が含まれる場合には、各変数の変数値を一覧表示させる際に、仮にユーザにより桁区切り有りで表示することが設定された場合であっても、少なくとも複素数形式や分数形式、平方根形式の数値については桁区切り無しで表示させるようになっている。

40

【0074】

すなわち、例えば図5（E）における複素数形式の変数Aの場合、仮にユーザにより桁区切り有りで表示することが設定された場合であっても、「 $A = 1, 23, 456 + 12, 345i$ 」のように桁区切り有りで表示させると見づらくなるため、図5（E）に示すように、「 $A = 123456 + 12345i$ 」のように桁区切り無しで表示させる。また、分数形式の数値の場合には、例えば「 $12, 345 / 1, 234$ 」ではなく「 1234

50

5 / 1 2 3 4 」と桁区切り無しで表示し、平方根形式の数値についても、例えば「 1 2 , 3 4 5 」ではなく「 1 2 3 4 5 」と桁区切り無しで表示させる。

【 0 0 7 5 】

以下、図 6 (A) ~ (L) に示す例に沿って具体的に説明する。なお、以下では、桁区切り種類としてインド (India) 方式が予め設定されている場合について説明するが、図 5 (A) に示した他の桁区切り種類に従って桁区切りを行う場合も同様に説明される。

【 0 0 7 6 】

図 6 (A) に示すように、例えば、ユーザが、テンキー 2 0 (図 1 参照) の「 2 」、「 5 」、「 0 」を押下して「 2 5 0 0 0 0 0 0 」を入力すると、関数電卓 1 の CPU 1 1 は、ディスプレイ 1 0 上に「 2 5 0 0 0 0 0 0 」と表示し (ステップ S 1 6)、STO キー 2 7 が押下されて Store 操作が行われ (ステップ S 8 ; Y E S)、変数キー 2 6 である「 A 」が押下されると (ステップ S 9 ; Y E S)、数式の入力中ではないから (ステップ S 1 0 ; N O)、CPU 1 1 は、入力された数値「 2 5 0 0 0 0 0 0 」を指定された変数 A の変数値として、変数メモリ 1 6 3 (図 2 参照) に設定するとともに、ディスプレイ 1 0 上に図 6 (A) に示すような表示を行う (ステップ S 1 2)。

【 0 0 7 7 】

そして、表示を見たユーザが、例えば、桁区切り有りで数値を表示させたい場合には、本実施形態では、桁区切り有無の設定操作を行うことができるようになっている (ステップ S 2 0)。桁区切り有無の設定操作は、ユーザが SHIFT キー 2 3 を押下した後 MODE キー 2 2 を押下すると (ステップ S 2 0 ; Y E S)、図 6 (B) に示すような SETUP 機能が一覽表示される。そして、図 6 (B) では図示を省略したカーソルを下に下げて、「 3 : S e p a r a t o r 」を選択すると、この場合は、図 6 (C) に示すように区切りモードのオン/オフが一覽表示されるため (ステップ S 2 1)、その中から「 1 : O n 」を選択することで、桁区切り有りの設定操作を行うことができるようになっている。

【 0 0 7 8 】

そして、CPU 1 1 は、上記のようにユーザ操作により選択されて設定された桁区切り有無の設定に応じて、各変数の変数値を桁区切り有り又は桁区切り無しで表示させるように設定する (ステップ S 2 2)。すなわち、上記のようにして、桁区切り有り (すなわち「 O n 」) が選択されて設定されると、桁区切り有りの設定に応じて、この場合は記憶部 1 6 (図 2 参照) に記憶されている国情報 1 6 1 (図 5 (B) 参照) として予め「 India 」が設定されているから、桁区切り種類情報 1 6 2 (図 5 (A) 参照) のうち国「 India 」に対応する桁区切り種類で、図 6 (A) のように桁区切り無しで表示していた変数 A の変数値「 2 5 0 0 0 0 0 0 」を桁区切り有りの「 2 , 5 0 , 0 0 , 0 0 0 」のように表示する (図 6 (D) 参照)。

【 0 0 7 9 】

また、この状態で、ユーザが、図 6 (E) に示すように、テンキー 2 0 の「 1 」、「 2 」、「 3 」、「 4 」、「 0 」を押下して「 1 2 3 4 0 0 0 0 0 0 」を入力して変数 C の変数値とする Store 操作が行われると、CPU 1 1 は、入力された数値「 1 2 3 4 0 0 0 0 0 0 」を指定された変数 C の変数値として、変数メモリ 1 6 3 (図 2 参照) に設定するとともに、ディスプレイ 1 0 上に桁区切り有りの状態で「 1 , 2 3 , 4 0 , 0 0 , 0 0 0 」と表示する (ステップ S 1 2)。

【 0 0 8 0 】

また、例えばこの時点で、ユーザが変数 Recall 一括表示処理 (図 3 のステップ S 1 ~ S 7) を行って各変数の変数値を一覽表示させると、CPU 1 1 は、変数メモリ 1 6 3 から各変数の変数値を呼び出して、図 6 (F) に示すように、各変数の変数値を桁区切り有りの状態で一覽表示するようになっている。

【 0 0 8 1 】

一方、図 6 (K) に示すように、本実施形態では、ユーザが SHIFT キーを押下した後 MODE キー 2 2 を押下して SETUP 機能の表示をさせた一覽表示 (図 6 (B) 参照) の中から「 3 : S e p a r a t o r 」を選択し、CPU 1 1 が表示させた On / O f f の一覽表示 (

10

20

30

40

50

ステップS 2 1。図6 (C) 参照)の中から「2 : O f f」を選択することで、桁区切り無しの設定操作を行うことができるようになっている(ステップS 2 2)。

【0082】

すなわち、本実施形態では、ユーザが「O n」又は「O f f」を選択して設定することで、桁区切り有り又は桁区切り無しの設定操作を行うことができるようになっている。そして、上記のように桁区切り無し(すなわち「O f f」)の設定操作が行われると、CPU 1 1は、その設定に応じて、この場合は、図6 (J) のように桁区切り有りに表示していたC Yの計算結果であり変数Dの変数値である「2 , 0 5 , 1 6 , 4 8 , 4 0 0」を、図6 (K) に示すように、桁区切り無しの「2 0 5 1 6 4 8 4 0 0」のように表示する。

【0083】

また、この時点で、ユーザがSHIFT キー2 3を押下した後STOキー2 7を押下することにより変数Recall一括表示処理を行って各変数の変数値を一覧表示させると、CPU 1 1は、変数メモリ1 6 3から各変数の変数値を呼び出して、図6 (L) に示すように、各変数の変数値を桁区切り無しの状態で一覧表示するようになっている。

【0084】

このように、本実施形態では、CPU 1 1は、各変数の変数値や計算結果を桁区切り有りに表示するか、桁区切り無しで表示するかをユーザが設定すると、それに応じて変数値や計算結果を桁区切り有り又は桁区切り無しで表示させることを設定する桁区切り有無設定手段として機能するようになっている。そして、前述した変数値一覧表示手段としてのCPU 1 1は、桁区切り有無設定手段(すなわち自ら)により設定された桁区切り有無の設定に応じて、各変数の変数値を一覧表示させるようになっている(図6 (F)、(L) 参照)。

【0085】

このように構成することで、各変数の変数値や計算結果を、ユーザが桁区切り有りに表示させたいか桁区切り無しで表示させたいかに応じて、桁区切り有り又は桁区切り無しで表示することが可能となる。そのため、ユーザが各変数の変数値や計算結果を見やすくなる。また、ユーザが自由に桁区切り有無を設定することができるため、関数電卓1(変数値表示制御装置)を、ユーザにとって使い勝手がよいものとする事が可能となる。

【0086】

なお、図6 (F) に示したように、ユーザが変数Recall一括表示処理を行い(ステップS 1 ; Y E S)、CPU 1 1がディスプレイ1 0上に各変数の変数値を一覧表示した状態で(ステップS 2)、例えば図6 (G) に示すように、ユーザが、続けて、例えば変数Aに対応する変数キー2 6を押下すると(ステップS 3 ; Y E S)、変数Recall一括表示処理前の状態(図6 (E) 参照)では、数式の入力中ではないから(ステップS 5 ; N O)、CPU 1 1は、変数メモリ1 6 3から変数Aの変数値を呼び出し、桁区切り有りが設定されているため、ディスプレイ1 0上に変数Aの変数値(この場合は「2 5 0 0 0 0 0 0」)を桁区切り有りに表示する。

【0087】

そして、ユーザは、インドルピーの換算レートが変数Xとして設定されていたか変数Yとして設定されていたかを忘れてしまった場合は、再度、変数Recall一括表示処理を行って、図6 (H) のようにディスプレイ1 0上に各変数の変数値を一覧表示させることで、変数Yにルピーの換算レートが設定されていることを確認することができる。

【0088】

また、図6 (I) に示すように、続けてユーザが変数Yに対応する変数キー2 6を押下すると(ステップS 3 ; Y E S)、各変数の変数値の一覧表示前には(図6 (G) 参照)、変数Aを入力して数式の入力中であったから(ステップS 5 ; Y E S)、CPU 1 1は、変数Yを入力中の数式「A」に追加入力する処理を行って(ステップS 6)、数式の入力処理を継続する。

【0089】

そして、ユーザが数式「A Y」の計算結果を変数Bの変数値とするStore操作を行うと

10

20

30

40

50

、CPU11は、数式「AY」の計算結果「41565000」を変数Bの変数値として設定する。なお、この場合、図6(I)に示すように、ディスプレイ10上の表示は桁区切り有りに表示される。また、同様にして、図6(J)に示すように、数式「CY」の計算結果を変数Dの変数値として設定する操作が行われると、CPU11は、数式「CY」の計算結果「2051648400」を変数Dの変数値として設定するとともに、ディスプレイ10上に計算結果を桁区切り有りに表示する。

【0090】

[桁区切り有無設定処理の変形例1]

なお、上記の桁区切り有無設定処理では、例えば国情報161における国が例えばIndiaに予め設定されている場合について説明した。そして、例えば、関数電卓1(変数値表示制御装置)がインド国内のみで使用される場合にはそれで十分である。

10

【0091】

しかし、この関数電卓1(変数値表示制御装置)を様々な国に持って行って使用するような場合には、ユーザが、国を設定することができるように構成されている方が便利である。そして、そのような場合には、ユーザが国を選択すると、CPU11が、設定された国に応じて桁区切り種類を適切に設定するように構成することが好ましい。

【0092】

この場合、CPU11は、各変数の変数値や計算結果を桁区切り有りに表示させる際の桁区切りの文字又は桁区切りの位置を桁区切り種類として設定する桁区切り種類設定手段として機能するとともに、ユーザ操作に応じて、いずれかの国を設定する国設定手段として機能するように構成する。そして、記憶部16を、国ごとに桁区切り種類(すなわち本実施形態では桁区切り種類情報162)を記憶した国別桁区切り種類記憶手段として機能させ、変数値一覧表示手段であるCPU11は、前述したように桁区切り有りが設定されている場合に、国設定手段(すなわち自ら)により設定された国に対応する桁区切り種類で、各変数の変数値を一覧表示させるように構成することが可能である。

20

【0093】

この場合、図示を省略するが、CPU11は、例えば、ユーザがSHIFTキー23の押下後にMODEキー22を押下してSETUP機能を実行して、その機能中で国名を選択すると、記憶部16に記憶されている国情報161(図5(B)参照)に基づいて桁区切り種類を割り出し、桁区切り種類情報162(図5(A)参照)に基づいて当該選択された国に対応する区切り種類で、各変数の変数値を一覧表示したり計算結果を表示したりするように構成される。

30

【0094】

[桁区切り有無設定処理の変形例2]

一方、上記のように、ユーザが国名を選択して設定する代わりに、桁区切り種類を直接設定するように構成することも可能である。この場合も、CPU11は、ユーザ操作により、各変数の変数値や計算結果を桁区切り有りに表示させる際の桁区切りの文字又は桁区切りの位置を桁区切り種類として設定する桁区切り種類設定手段として機能する。以下、図7(A)~(I)を用いて具体的に説明する。

【0095】

図7(A)に示すように、例えば、ユーザが、テンキー20(図1参照)の「3」、「2」、「5」、「0」を押下して「3250000」を入力して、STOキー27を押下してStore操作を行い、変数キー26である「E」を押下すると、CPU11は、入力された数値「3250000」を指定された変数Eの変数値として変数メモリ163(図2参照)に設定するとともに、図7(A)に示すようにディスプレイ10上に表示する。

40

【0096】

そして、ユーザが、桁区切り有りで数値を表示させるためにSHIFTキー23の押下後にMODEキー22を押下してSETUP機能を実行すると、各種モードが一覧表示され(図6(B)参照)、その中から「3: Separator」を選択すると、この場合、CPU11は、図7(B)に示すように、桁区切り種類を一覧表示する。そして、ユーザが、その中か

50

ら例えば「3 : Space (DN)」を選択すると、CPU 11は、記憶部 16 (図 2 参照) に記憶されている桁区切り種類情報 162 (図 5 (A) 参照) を参照し、その中の「Space(DN)」の桁区切り種類に対応する桁区切りの仕方で、図 7 (C) に示すように、変数 E の変数値「3250000」を桁区切り有りの「3 250 000」のように表示する。

【0097】

また、この状態で、ユーザが、例えばSHIFT キー 23 を押下した後STOキー 27 を押下することにより変数Recall一括表示処理を行って、図 7 (D) に示すように各変数の変数値を一覧表示させると、CPU 11は、変数メモリ 163 から各変数の変数値を呼び出して、各変数の変数値を、「Space(DN)」の桁区切り種類に対応する桁区切り有りの状態で一覧表示する。

10

【0098】

そして、例えば、ユーザは、中国元の換算レートが変数 X として設定されていたか変数 Y として設定されていたかを忘れてしまった場合は、再度、変数Recall一括表示処理を行って、図 7 (F) のようにディスプレイ 10 上に各変数の変数値を一覧表示させることで、変数 X に元の換算レートが設定されていることを確認することができる。

【0099】

そして、ユーザが続けて変数 X に対応する変数キー 26 を押下すると、CPU 11は、図 7 (G) に示すように、変数 X を入力中の数式「E」に追加入力する処理を行う。そして、ユーザが数式「EX」の計算結果を変数 F の変数値とするStore操作を行うと、CPU 11は、数式「EX」の計算結果「54018250」を変数 F の変数値として設定するとともに、「Space(DN)」の桁区切り種類に対応する桁区切り有りの状態で表示する。

20

【0100】

また、ここで、ユーザが、桁区切り種類を変えて表示させたい場合には、図 7 (B)、(C) と同様に、SHIFT キーを押下した後MODEキー 22 を押下してSETUP機能の各種モードの一覧表示の中から「3 : Separator」を選択し、図 7 (B) に示した桁区切り種類の一覧表示の中から例えば「1 : Comma」を選択すると、CPU 11は、記憶部 16 (図 2 参照) に記憶されている桁区切り種類情報 162 (図 5 (A) 参照) を参照する。

【0101】

そして、その中の「Comma」の桁区切り種類に対応する桁区切りの仕方で、図 7 (H) に示すように、計算結果「54018250」を「Comma」の桁区切り種類に対応する桁区切り有りの状態で「54, 018, 250」のように表示する。また、この場合、変数Recall一括表示処理を行うと、図 7 (I) に示すように各変数の変数値が「Comma」の桁区切り種類に対応する桁区切り有りの状態で一覧表示される。

30

【0102】

この変形例 2 のように構成することで、ユーザは、各変数の変数値や計算結果を、所望の桁区切り種類で表示させることが可能となり、関数電卓 1 (変数値表示制御装置) がユーザにとって使い勝手がよいものとなる。

【0103】

以上、本発明の実施形態やいくつかの変形例を説明したが、本発明の範囲は、上述の実施形態等に限定するものではなく、特許請求の範囲に記載された発明の範囲とその均等の範囲を含む。

40

以下に、この出願の願書に最初に添付した特許請求の範囲に記載した発明を付記する。付記に記載した請求項の項番は、この出願の願書に最初に添付した特許請求の範囲の通りである。

〔付記〕

< 請求項 1 >

複数の変数について変数値を記憶するための変数値記憶手段と、

ユーザ操作に応じて、前記複数の変数のうちのいずれかの変数を指定して前記変数値記

50

憶手段に指定の変数の変数値を設定する変数値設定手段と、

ユーザ操作に応じて、前記変数値記憶手段に記憶された各変数の変数値を表示部に一覧表示させる変数値一覧表示手段と、

ユーザ操作に応じて、前記一覧表示された変数のうちのいずれかの変数が指定されると、当該変数を入力する一覧変数入力手段と、
を備えることを特徴とする変数値表示制御装置。

< 請求項 2 >

ユーザ操作に応じて入力された数式を表示する入力数式表示制御手段を備え、

前記入力数式表示制御手段により入力された数式を表示された状態で、前記変数値一覧表示手段により各変数の変数値を一覧表示された後に、一覧変数入力手段によりいずれかの変数が指定されると、前記入力された数式に当該変数を追加入力することを特徴とする請求項 1 に記載の変数値表示制御装置。

< 請求項 3 >

ユーザ操作に応じて入力された、前記追加入力された変数を含む数式を計算して計算結果を表示させる計算結果表示手段を備える請求項 2 に記載の変数値表示制御装置。

< 請求項 4 >

前記計算結果又は前記変数値を桁区切り有りに表示するか、桁区切り無しで表示するかをユーザが設定すると、それに応じて前記計算結果又は前記変数値を桁区切り有り又は桁区切り無しで表示させることを設定する桁区切り有無設定手段を備え、

前記変数値一覧表示手段は、前記桁区切り有無設定手段により設定された桁区切り有無の設定に応じて、各変数の変数値を一覧表示させることを特徴とする請求項 1 から請求項 3 のいずれか一項に記載の変数値表示制御装置。

< 請求項 5 >

前記計算結果又は前記変数値を桁区切り有りに表示させる際の桁区切りの文字又は桁区切りの位置を桁区切り種類として設定する桁区切り種類設定手段と、

国ごとに桁区切り種類を記憶した国別桁区切り種類記憶手段と、

ユーザ操作に応じて、いずれかの国を設定する国設定手段と、
を備え、

前記変数値一覧表示手段は、前記桁区切り有無設定手段により桁区切り有りが設定されている場合に、前記国設定手段により設定された国に対応する桁区切り種類で、各変数の変数値を一覧表示させることを特徴とする請求項 4 に記載の変数値表示制御装置。

< 請求項 6 >

前記計算結果又は前記変数値を桁区切り有りに表示させる際の桁区切りの文字又は桁区切りの位置を桁区切り種類として設定する桁区切り種類設定手段を備え、

前記桁区切り種類設定手段は、ユーザ操作により桁区切り種類を設定することを特徴とする請求項 4 に記載の変数値表示制御装置。

< 請求項 7 >

前記変数値一覧表示手段は、前記いずれかの変数値に複素数形式、分数形式、又は平方根形式の数値が含まれる場合に、各変数の変数値を一覧表示させる際に、前記桁区切り有無設定手段により設定された桁区切り有無の設定にかかわらず、複素数形式、分数形式、又は平方根形式の数値については桁区切り無しで当該変数を表示させることを特徴とする請求項 4 から請求項 6 のいずれか一項に記載の変数値表示制御装置。

< 請求項 8 >

前記表示部と、キー群を備え、前記キー群に対して前記ユーザ操作を行うことを特徴とする請求項 1 から請求項 7 のいずれか一項に記載の変数値表示制御装置。

< 請求項 9 >

複数の変数について変数値を記憶するための変数値記憶手段を備えるコンピュータに、

ユーザ操作に応じて、前記複数の変数のうちのいずれかの変数を指定して前記変数値記憶手段に指定の変数の変数値を設定する変数値設定機能と、

ユーザ操作に応じて、前記変数値記憶手段に記憶された各変数の変数値を表示部に一覧

10

20

30

40

50

表示させる変数値一覧表示機能と、

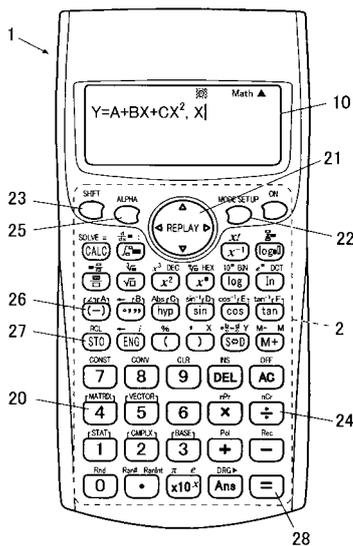
ユーザ操作に応じて、前記一覧表示された変数のうちのいずれかの変数が指定されると、当該変数を入力する一覧変数入力機能と、を実現させることを特徴とするプログラム。

【符号の説明】

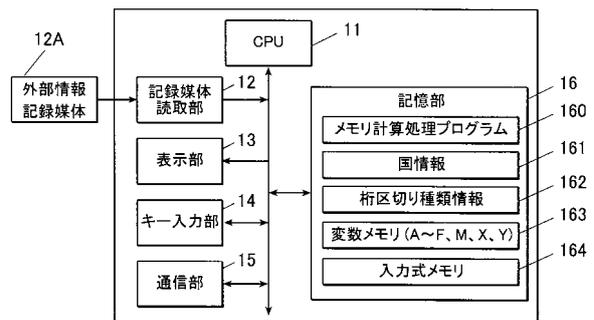
【0104】

- 1 関数電卓（変数値表示制御装置）
- 11 CPU（変数値設定手段、変数値一覧表示手段、一覧変数入力手段、入力数式表示制御手段、計算結果表示手段、桁区切り有無設定手段、桁区切り種類設定手段、国別桁区切り種類記憶手段、国設定手段）
- 16 記憶部（変数値記憶手段、国別桁区切り種類記憶手段）
- 160 メモリ計算処理プログラム（プログラム）
- A～F、M、X、Y 変数

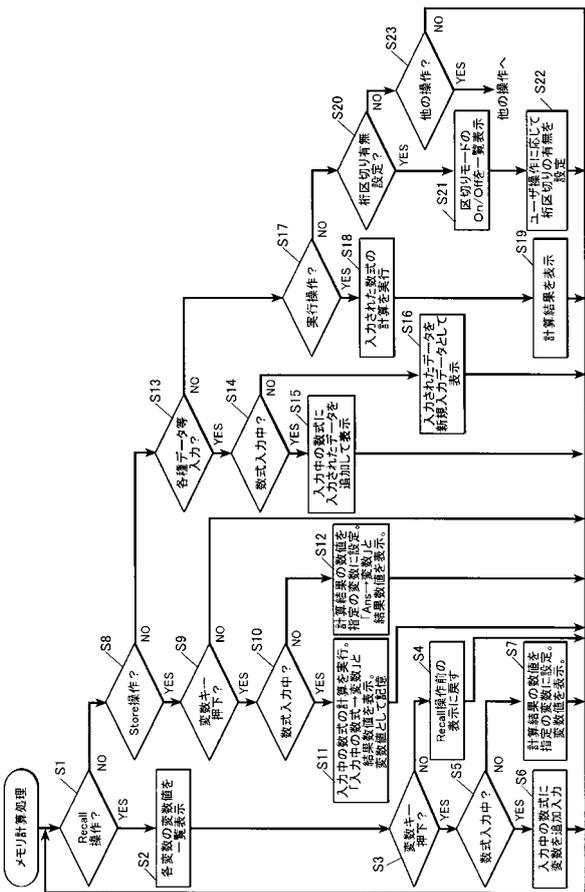
【図1】



【図2】



【 図 3 】



【 図 5 】

(A)

種類	type	小数部	表示
a : Comma	Comma	無し	-123,456.7891 × 10 ⁴⁵
b : Dot	Dot	無し	-123.456,7891 × 10 ⁴⁵
c : Space(DN)	Space	有り	-123 456.789 1 × 10 ⁴⁵
d : Space	Space	無し	-123 456.7891 × 10 ⁴⁵
e : India	India	無し	-1,23,456.7891 × 10 ⁴⁵
f : 区切りなし	Off	無し	-123456.7891 × 10 ⁴⁵

(B)

国	設定 1: On	設定 1: Off
China	d : Space	f : 区切りなし
India	e : India	f : 区切りなし
Japan	a : Comma	f : 区切りなし
...	..	f : 区切りなし

(C)

A=1234567890 C=1234567890 E=12345 M=12345 y=0	B=1234567890 D=1234567890 F=12345 x=12345
---	--

(D)

A=1,23,45,67,890 C=1,23,45,67,890 E=12,345 M=12,345 y=0	B=1,23,45,67,890 D=1,23,45,67,890 F=12,345 x=12,345
---	--

(E)

A=123456+12345i C=123456+12345i E=1+i M=1j+1j3i y=0	B=123456+12345i D=123456+12345i F=1+i x=1j+1j3i
---	--

【 図 4 】

(A) $2 \rightarrow A$

(B) $3 \rightarrow C$

(C) $A=2, C=3, E=0, M=0, y=0$

(D) 数式入力

(E) $=$

(F) $X \rightarrow X$

(G) 数式修正

(H) $=$

(I) $M \rightarrow Y$

(J) $RCL \rightarrow STO$

Math ▲ $-B + \sqrt{B^2 - 4AC}$
CMPLX $\frac{-B + \sqrt{B^2 - 4AC}}{2A}$

Math ▲ $1 - \frac{\sqrt{5}}{2}$
CMPLX $\frac{1 - \sqrt{5}}{2}$

Math ▲ $1 - \frac{\sqrt{5}}{2}$
CMPLX $\frac{1 - \sqrt{5}}{2}$

Math ▲ $A=2, D=0, E=0, M=0, x=-1, y=2 + \sqrt{5}, z=1 - \sqrt{5}$
CMPLX $\frac{-B + \sqrt{B^2 - 4AC}}{2A}$

【 図 6 】

(A) $25000000 \rightarrow A$

(B) 1: Recur Decimal, 2: Decimal Point, 3: Separator, 4: Simplify

(C) Separator 1: On, 2: Off

(D) $25000000 \rightarrow A$

(E) $1234000000 \rightarrow C$

(F) $A=2, C=1, E=0, M=0, x=16.621$

(G) $\alpha \rightarrow (-)$

(H) $RCL \rightarrow STO$

(I) $M \rightarrow Y$

(J) $\alpha \rightarrow \sin$

(K) MODE SETUP

(L) $RCL \rightarrow STO$

【 7 】

(A) 3250000 → E 3250000
 (STO) (cos) (E)

(B) Separator
 1: Comma 4: Space
 2: Dot 5: India
 3: Space(DN) 6: Off
 (SHIFT) (MODE SETUP) (3)

(C) 3250000 → E 3 250 000
 (3)

(D) A=25 000 000 B=41 565 000
 C=7 234 000 000 D=205 164 840
 E=3 250 000 F=J
 M=0 X=16.621
 V=1.6626
 (RCL) (SHIFT) (STO)

(E) E 3 250 000
 (ALPHA) (cos) (E)

(F) A=25 000 000 B=41 565 000
 C=7 234 000 000 D=205 164 840
 E=3 250 000 F=J
 M=0 X=16.621
 V=1.6626
 (RCL) (SHIFT) (STO)

(G) EX → F 54 018 250
 (ALPHA) (X) (F) (tan) (STO)

(H) EX → F 54,018,250
 (SHIFT) (MODE SETUP) (3) (1)

(I) A=25 000 000 B=41 565 000
 C=7 234 000 000 D=205 164 840
 E=3 250 000 F=54,018,250
 M=0 X=16.621
 V=1.6626
 (RCL) (SHIFT) (STO)