(12) UK Patent (19) GB (11) 2 222 476 (13) B

(54) Title of Invention

A data input device for a data processing device

(51) INT CL[5]; G06K 9/70

(21) Application No
8919522.6

(22) Date of filing
29.08.1989

(30) Priority Data

(31) 63215572

(32) 30.08.1988

(33) JP

(43) Application published
07.03.1990

(45) Patent published
21.07.1993

(72) Inventor(s)
Kimiyoshi Yoshida
Akio Sakano
Hirofumi Tamori
Masayuki Chatani

(73) Proprietor(s)
Sony Corporation

(Incorporated in Japan)

7-35 Kitashinagawa 6-chome
Shinagawa-ku
Tokyo
Japan

(74) Agent and/or
Address for Service
J A Kemp & Co
14 South Square
Gray's Inn
London
WC1R 5LX
United Kingdom

(52) Domestic classification
(Edition L)
G4R RRP R1C R1X R11D
R11E R11F R3G R8A R9B

(56) Documents cited
EP0279158 A2

(58) Field of search

As for published application
2222476 A *viz:*
UK CL(Edition J) G4R RRP
INT CL[4] G06K
updated as appropriate

2222476

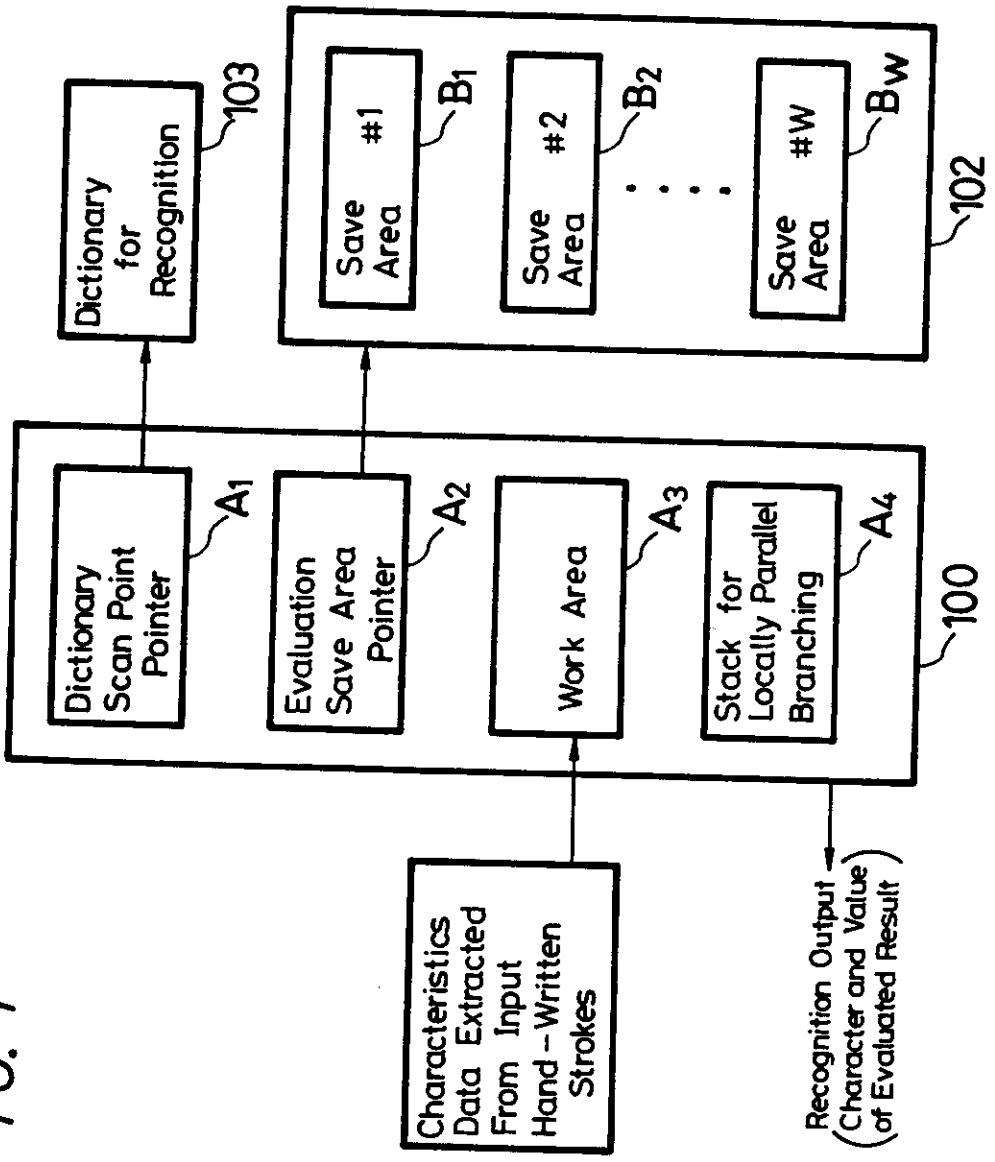


FIG. 1

2222476

FIG. 2

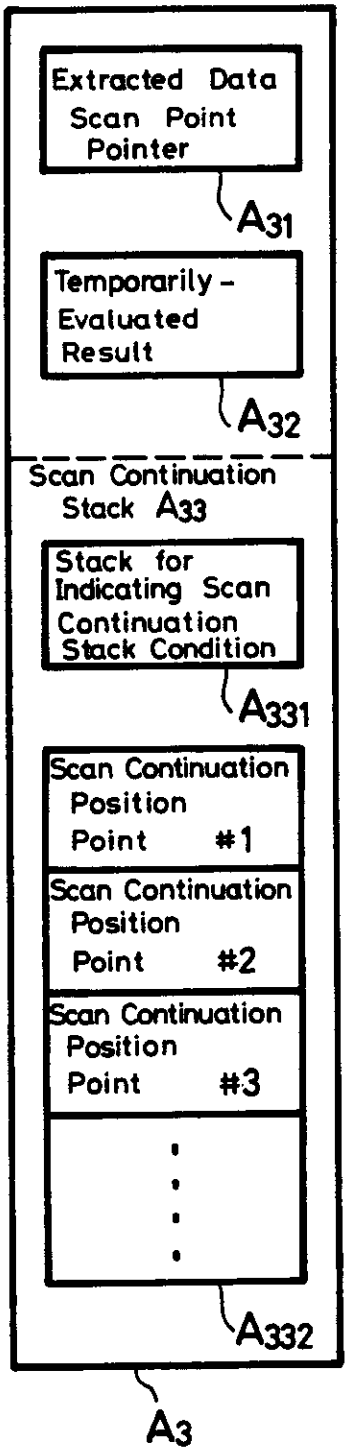


FIG. 3

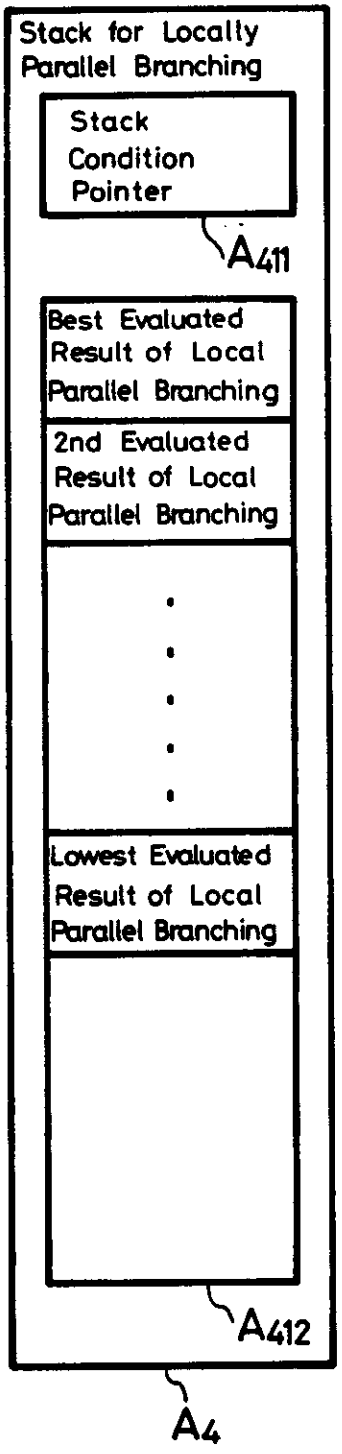


FIG. 4

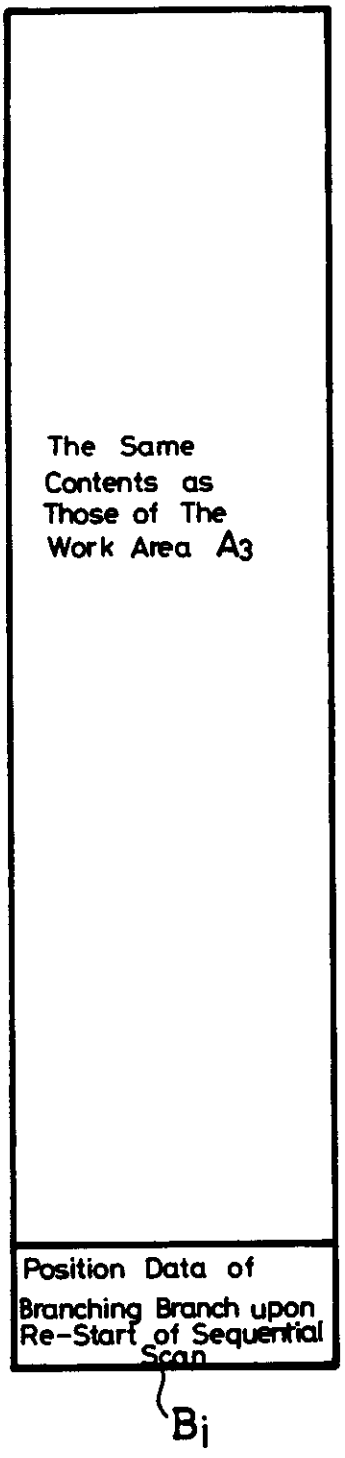


FIG. 5

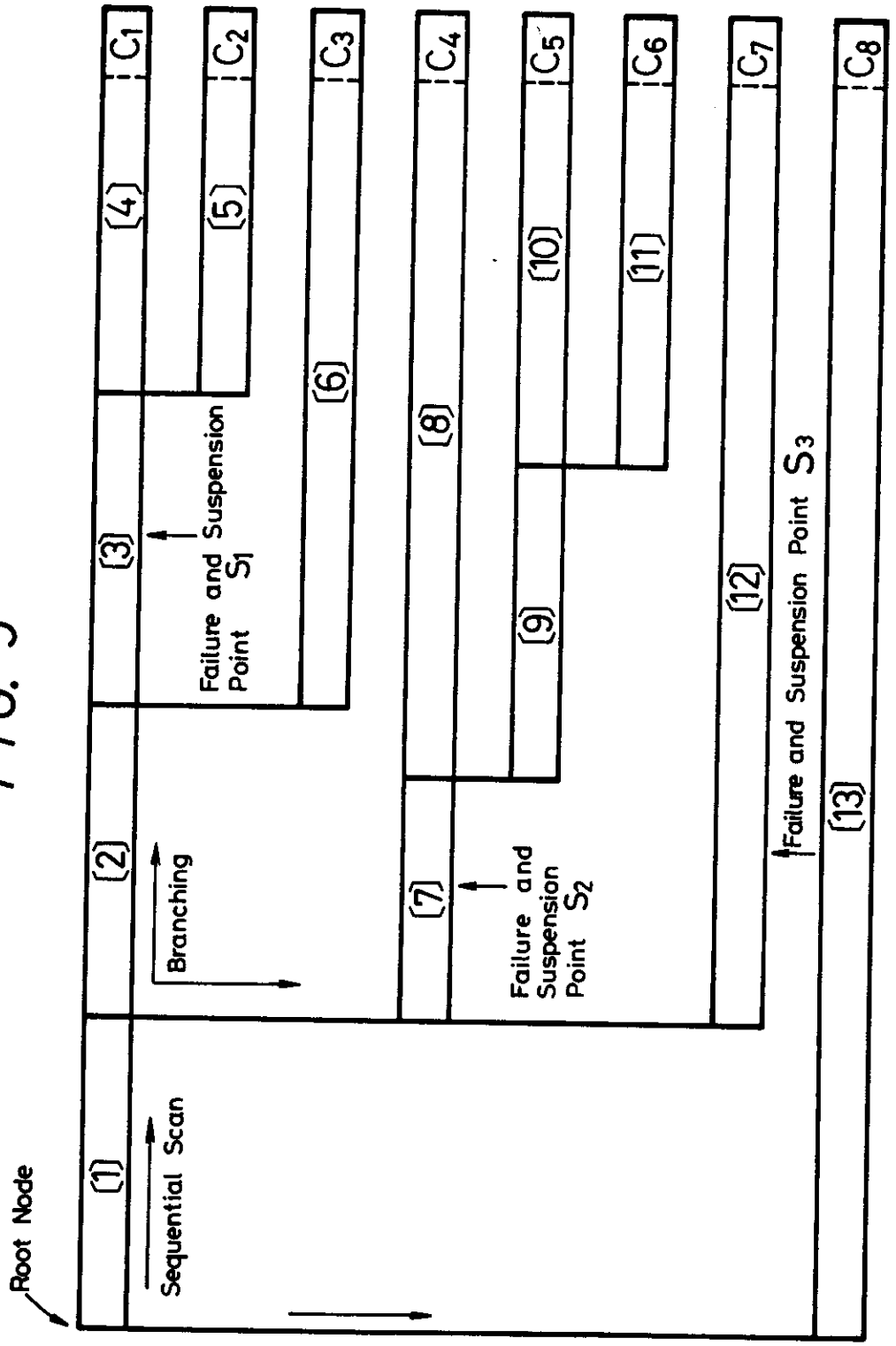

Root Node
Sequential Scan
Branching
Failure and Suspension Point S1
Failure and Suspension Point S2
Failure and Suspension Point S3
C1
C2
C3
C4
C5
C6
C7
C8
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)

FIG. 6A

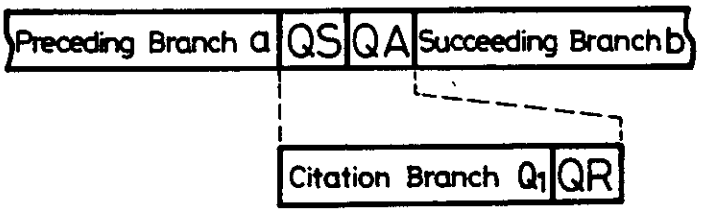


FIG. 6B



FIG. 7A

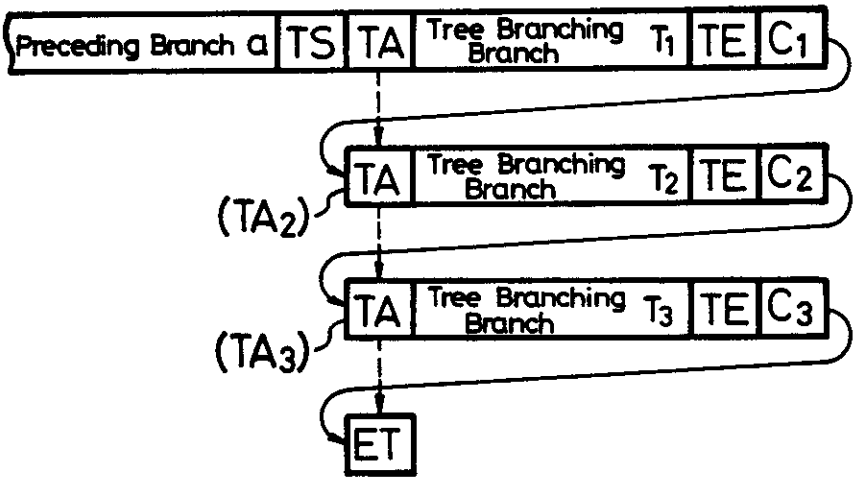
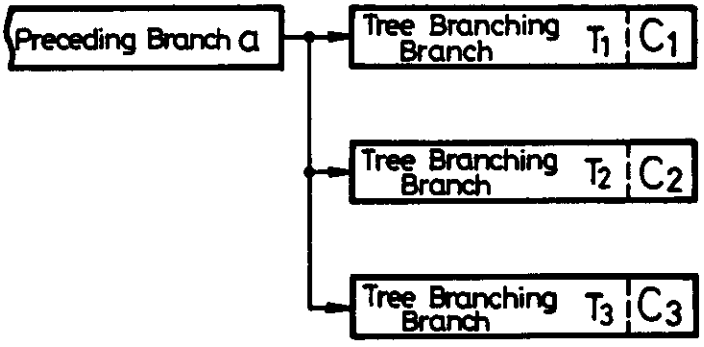


FIG. 7B



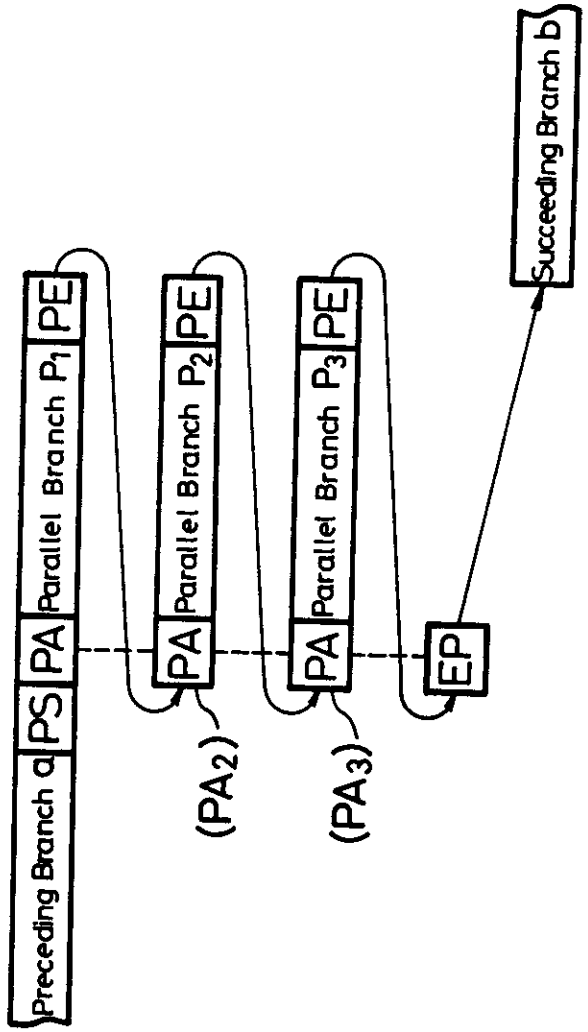

Preceding Branch a
PS
PA
Parallel Branch P1
PE
PA
Parallel Branch P2
PE
PA
Parallel Branch P3
PE
EP
Succeeding Branch b
(PA2)
(PA3)

FIG. 8A

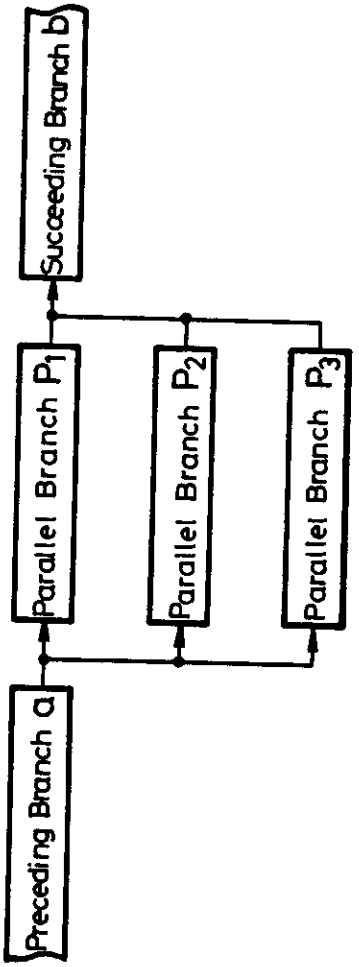

Preceding Branch a
Parallel Branch P1
Parallel Branch P2
Parallel Branch P3
Succeeding Branch b

FIG. 8B

2222476

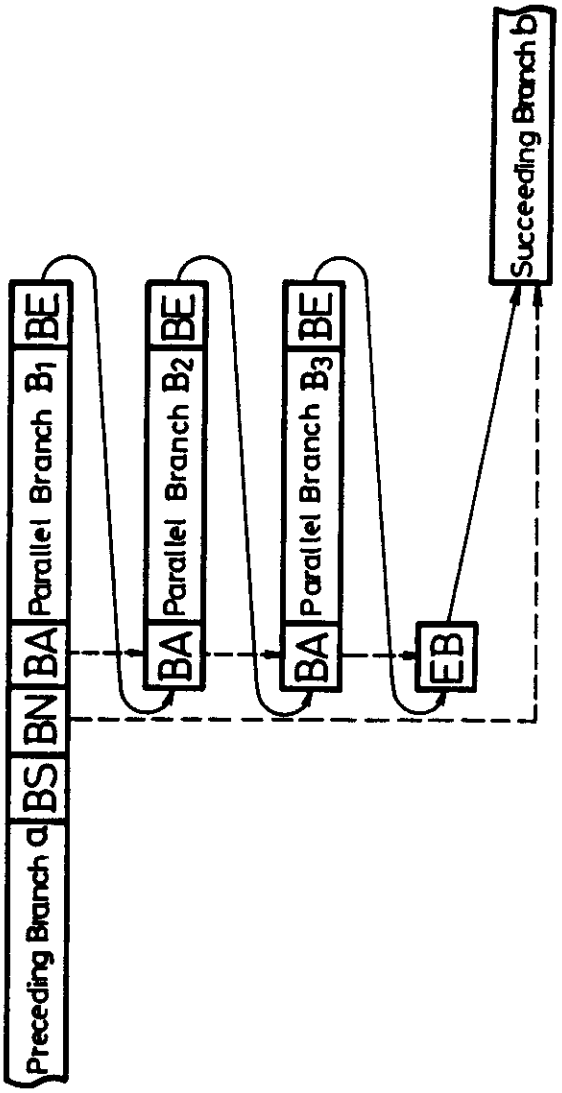


FIG. 9A

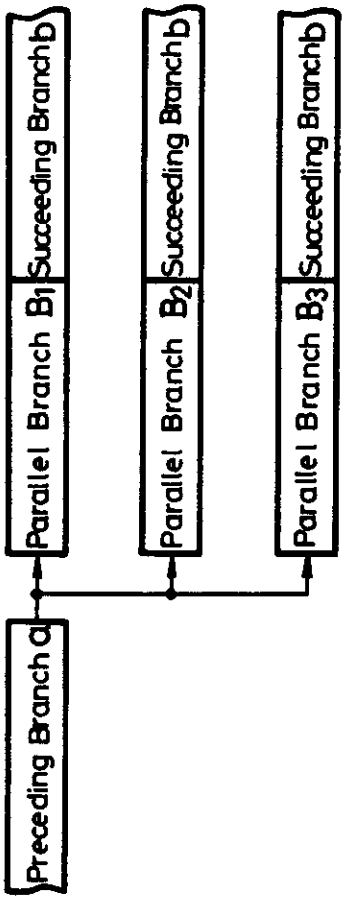
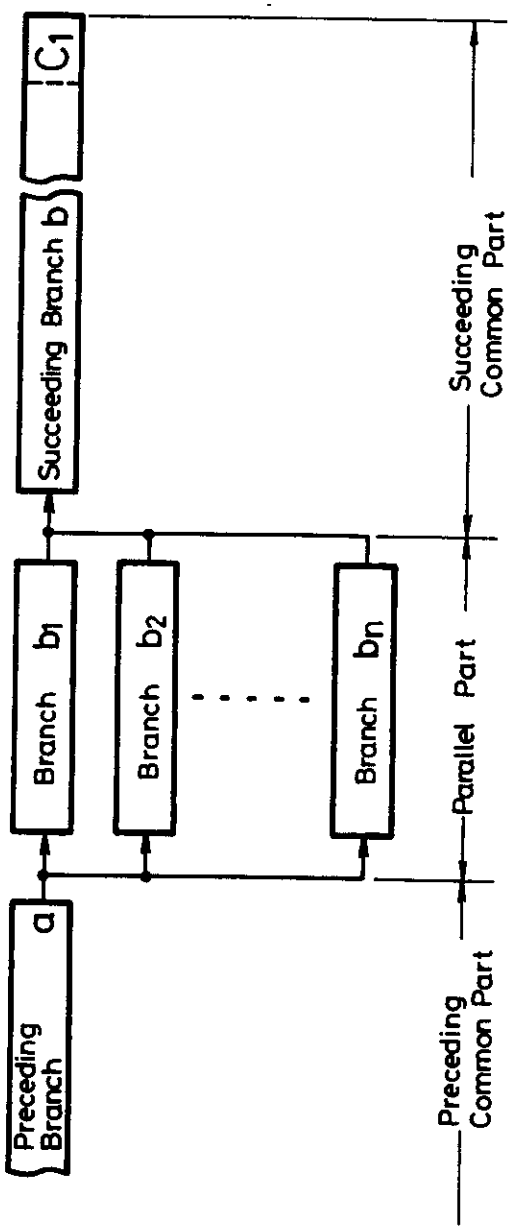


FIG. 9B

2222476

FIG. 10



2222476

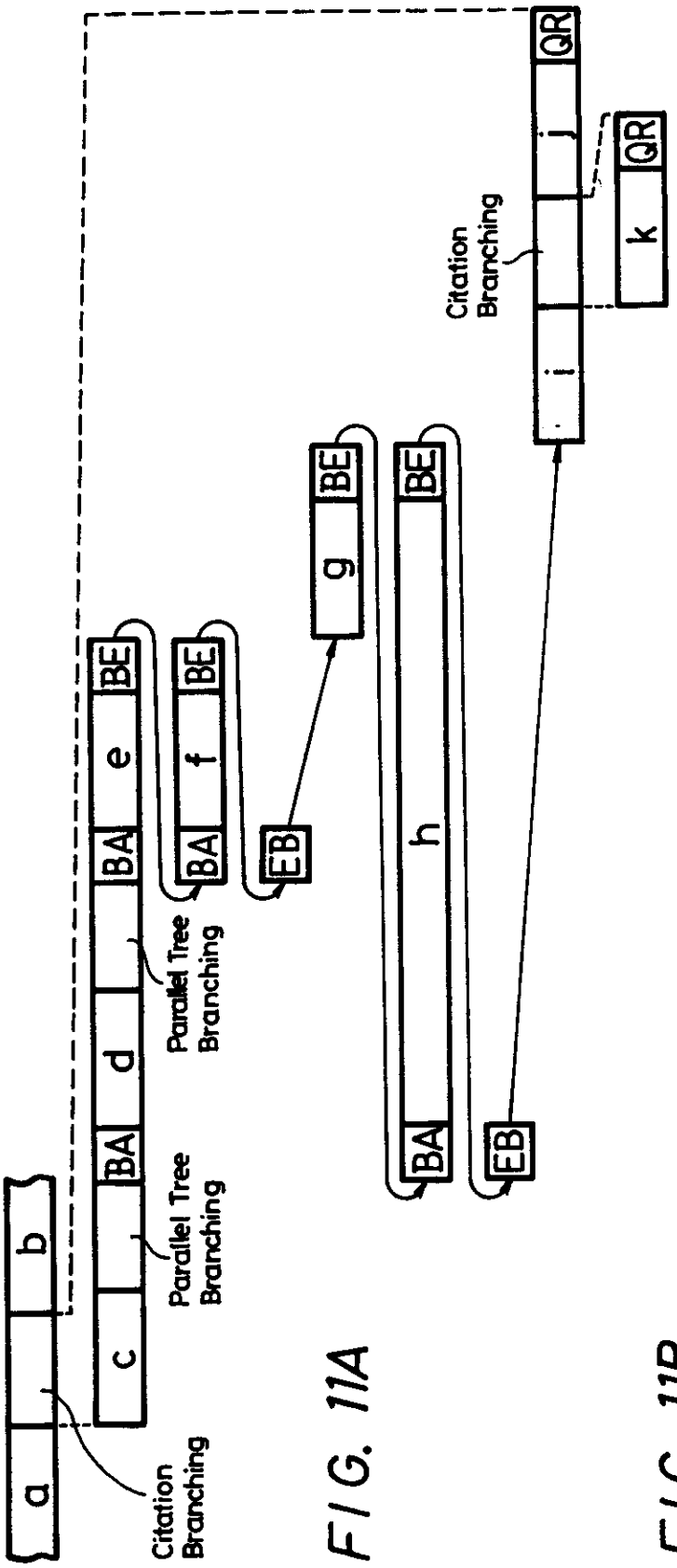


FIG. 11A

FIG. 11B

A DATA INPUT DEVICE FOR A DATA PROCESSING DEVICE

The present invention relates generally to an apparatus for inputting data to a data processing device, i.e. an on-line apparatus, and, more particularly, to one using an evaluation tree scan control apparatus.

In order to understand the present invention more clearly, let us first explain the general principles of stroke order of hand-written Japanese characters. Though there are inevitably a number of exceptions, most Japanese characters are written according to the established principles of stroke order. Generally, hand-written characters are recognized in the following two steps.

1st step: To extract various characteristics from strokes (dot data) of a hand-written character data obtained from a data tablet by a tablet pen.

2nd step: To evaluate the characteristics extracted in the 1st step by comparing them with the characteristics of each character described in a recognition dictionary data base, thereby detecting the character having the best evaluated result.

The recognition dictionary data base used in the 2nd step describes therein characteristics of respective characters as follows.

For example, the Japanese character (KANJI in Japanese) "一" describes a characteristic which means a horizontal stroke "→." Further, the Japanese

character " 二 " describes a characteristic which means two strokes but in the cursive-style or in the semi-cursive style. This symbol describes a characteristic of "→," " ↓ " and "↘." In this case, the broken arrow characteristic " ↓ " indicates the movement of a pen between the two hand-written strokes. Furthermore, a character " 十 " is written by two strokes but it really describes characteristics of "→," " ↖ " and "↓." Also in this example, the broken arrow characteristic represents the movement of the pen between two hand-written sub-strokes. In this case, the description of "↙" is included in the above-mentioned two Kanji characters " 二 " and " 十 ."

This is also true for Kanji characters having a common portion. Though the above-mentioned Kanji characters " 二 " and " 十 " are simpler and are essentially considered to be single element characters, most Kanji comprise two or more elements. In almost all cases, there is one key element, known as the radical, which indicates the general nature of the character. This radical combines with one or more other elements which give more specific information. For example, the Kanji character "木木" (meaning "forest" in English) is the combination of the radical "木" (meaning "tree" and "wood" in English) and the Kanji character " 木 " (meaning "tree" in English) to provide the meaning of "forest" in English. Further, the Kanji character "松" is the combination of radical " 木 " and another element " 公 " (meaning "public," "fair" or "lord" in English) to provide the meaning of "pine" in English.

As described above, though the Kanji characters "林" and "松" have the common radical " 木 ," they are separately described in the dictionary. Furthermore,

there are Kanji characters which allow a plurality of stroke orders when hand-written. Let us consider Kanji when hand-written. For example, let us consider the Kanji character, "右" (meaning "right" in English). The Kanji character "右" is the combination of the element " 十 " and the element of " 口 " ("mouth" in English) to provide the meaning of "right" in English. In this case, in order to write the Kanji "右," there are two stroke orders. One describes characteristics of "→," "↑ ," "↵" and the other describes characteristics "←," " ↑ " and "→." In addition, the characteristics of the Kanji character " 口 " are again described therein.

As described above, though the characteristics described in the prior art dictionary have the common portion, the prior-art dictionary describes the whole characteristics of different Kanji characters in parallel. As a result, since the prior-art dictionary for recognition is constructed as described above, the dictionary has many overlapping portions for Kanji characters and thus, the dictionary is inevitably made large in size.

Further, during evaluation, the common characteristic portion is scanned in a multiplexed fashion, i.e. is scanned a multiple number of times, which increases the time required for recognition.

According to the present invention, there is provided a data input apparatus for a data processing device, the apparatus being operative to respond to hand-written script on a tablet to produce electronic data corresponding to the textual content thereof and comprising:

a tablet and a tablet pen for generating sequential characteristic data from stroke data of a hand-written character;

a scan control central unit for receiving said sequential characteristic data;

a recognition dictionary memory organized in a tree and branch structure for electronically storing character data describing characteristics of characters, the tree structure of the dictionary means including at least citation branching, parallel branching and tree branching, the parallel branching being classified into parallel tree branching and locally parallel branching;

evaluation scan control means for scanning the tree and branches of the dictionary memory to search for character data corresponding to the hand-written pattern information data; and

save means for storing the conditions of the evaluation scan control means, wherein at least the conditions of the scanning existing at the beginning of each branch are saved in the save means upon branching in order to allow recursive use of branches of the character data stored in the dictionary means.

These and other objects, features and advantages of the present invention will be apparent in the following detailed description of a preferred embodiment when read in conjunction with the accompanying drawings, in which like reference numerals are used to identify the same or similar parts in the several views.

Fig. 1 is a block diagram schematically showing a functional block arrangement illustrating the

operation of an apparatus according to an embodiment of the present invention, although it is to be understood that the operations are actually carried out by a programmed computer.

Fig. 2 is a schematic diagram showing a work area used in the apparatus and method of the present invention;

Fig. 3 is a schematic diagram showing a locally parallel stack used in the apparatus and method of the present invention;

Fig. 4 is a schematic diagram showing a save area used in the apparatus and method of the present invention;

Fig. 5 is a schematic representation used to explain the outline of the evaluation applied to incoming strokes of a hand-written character by using a dictionary;

Figs. 6A and 6B are, respectively, schematic representations of a logic structure of a dictionary used in the present invention, and which are useful for explaining the citing description of the present invention;

Figs. 7A and 7B are, respectively, schematic representations used to explain tree branching processing of the present invention;

Figs. 8A and 8B are, respectively, schematic representations used to explain locally parallel branching of the present invention;

Figs. 9A and 9B are, respectively, schematic diagrams used to explain parallel tree branching of the present invention;

Fig. 10 is a schematic diagram to which reference is made in explaining parallel branching of the present invention; and

Figs. 11A and 11B are schematic representations to which reference is made in explaining the logic structure of the present invention, respectively.

An embodiment of an apparatus according to the present invention will now be described in detail with reference to the drawings.

Fig. 1 shows an overall functional block arrangement of an apparatus according to an embodiment of the present invention. It will be understood that, for the most part, the apparatus is a computer programmed to carry out the functions now to be discussed.

Referring to Fig. 1, it will be seen that the hand-written, on-line character recognition apparatus and method of the present invention is comprised of a scan control central unit 100, a scan condition save memory unit 102 and a recognition dictionary data base 103.

The scan control central unit 100 receives sequential characteristic data which are extracted from the stroke data of a hand-written character obtained from a tablet pen (not shown). Then the scan control central unit evaluates the thus obtained handwritten character data on the basis of the characteristics of each character derived from the recognition dictionary data base 103.

In order to execute the above-mentioned processing, the scan control central unit 100 includes a dictionary scan point pointer A_1, an evaluation save area pointer A_2, a work area A_3 and a stack area A_4 for locally parallel branching.

In this embodiment, the dictionary scan point pointer A_1 indicates a present position at which the dictionary data base 103 is to be read out. The pointer A_1 might be, for example, an address register, and an evaluation command is derived from the dictionary data base 103 at the point indicated by the pointer A_1. The evaluation save area pointer A_2 indicates an area for saving the various conditions of the central unit 100 when branching occurs during the evaluation-process, for example, when the content of the dictionary at the point indicated by the dictionary scan point pointer A_1 means a tree branch on a parallel tree branch. Each time branching occurs, the evaluation save area pointer A_2 indicates one save area within the scan condition save memory unit 102.

Fig. 2 shows the contents of the work area A_3 in the memory of the central unit 100 more fully. Referring to Fig. 2, it will be seen that the work area A_3 is provided with an extracted data reading point pointer A_{31} for the hand-written strokes necessary to sequentially execute the recognition evaluation process, a temporarily-evaluated result memory (partial coincidence or the like) A_{32} and a scan continuation stack A_{33} for the recognition dictionary data base 103. In order to allow a recursive citation, the stack area A_{33} is provided with a stack area A_{332} for saving a plurality of scan continuation points and A_{331} for indicating the stack condition such as the number of scan continuation points saved in the stack area A_{332} (the position of the latest scan continuation point saved in the stack area A_{332}). The stack A_{33} generally operates in a first-in-last-out (FILO) fashion.

The arrangement of the stack A_4 for locally parallel branching will be described in more detail

with reference to Fig. 3. As shown in Fig. 3, the stack A_4 is provided with a stack condition pointer A_{411} for indicating a multiplicity of locally parallel branches and a plurality of evaluation result store memories A_{412} for storing the temporarily-evaluated results of parallel branching.

Referring back to Fig. 1, the scan condition save memory unit 102 is comprised of W save areas B_1, B_2, ... B_W where W is a value larger than the maximal stages of the branches in the recognition dictionary data base 103. In this embodiment, the i-th save area B_i (i = any one of 1 to W) is a memory which stores the various conditions of the central unit 100 at a time point when the branching occurs in the evaluation-process of the recognition dictionary data base 103 and the starting position of the next branch.

The outline of the evaluation tree scan processing which is executed on the incoming hand-written character by using the recognition dictionary data base 103 will be described next with reference to Fig. 5. In Fig. 5, reference numerals [1] to [13] denote the sequential order of the evaluation scan when a suspension of the scan does not occur. Suspension points S_1 to S_3 are points at which the degree of inconsistency becomes worse and then exceeds a predetermined limit value when executing the sequential evaluation scan of strokes of the hand-written character from the root or start node. When the evaluation scan reaches the character code C_n (n = any one of 1 to 8) of the terminal of the branch without being suspended, the recognized result of the above character code C_n is generated as a candidate character together with the evaluation result. Further, in the illustrated example, the number of branches is selected to be four in the sequential

scan, i.e. the root node, and node points between branches [1] and [2], [2] and [3], and [3] and [4].

Sequential scan

The sequential scan will now be described. In the sequential scan, an evaluation command is sequentially derived from the character dictionary data base 103 in accordance with the branch of the content (evaluation tree) of the character dictionary data base 103. Thereafter, characteristic data of the input hand-written stroke (this is extracted from the hand-written stroke) is evaluated on the basis of the evaluation command described in each branch.

Branching scan

Each time when the evaluation scan reaches the branching node of the evaluation tree in the sequential scan, the condition of the central unit 100 at that time point is saved in the save area B_i and then the sequential scan is continued along with the first branch (in the case of Fig. 5, branch [2], [3] and [4], etc.)

Suspension of scan

When the value of the evaluated result exceeds a predetermined limit in the sequential scan (during the evaluation process), the scan to the succeeding branch is suspended and the evaluation is returned to the preceding branching node. Then, the condition of the central unit 100 is returned to the condition presented just before the branch, as saved in the save area B_i, and the sequential scan is re-started along with the other branch (branch [6] in the case of Fig. 5). The re-start point is the node nearest to the point at which the scan is suspended (the node between [2] and [3] in Fig. 5.).

Terminal end of scan

When the sequential scan reaches the terminal end of the tree, the character code Cn described at

the terminal end is used as the output character and the candidate character code is generated together with the evaluation point. After the candidate character code is generated, the sequential scan is re-started in accordance with another branch from the nearest branch node similarly to the suspension of scan. For example, if the scan is terminated at the branch [6], the scan is restarted from the branch [7].

The scan of the evaluation tree shown in Fig. 5 will now be explained. It will be seen in Fig. 5 that as to a certain hand-written stroke, the scan is carried out in the sequential order of the evaluation branches [1], [2] and [3]. If the degree of inconsistency, i.e. the extent to which the hand-written stroke does not match the character under consideration from the recognition dictionary data base, is increased at the evaluation branch [3], the scan is suspended at the suspension point S_1 and then the scan is executed for an evaluation branch [6]. After the character code C3 is generated together with the evaluation point, the scan of the evaluation branch [7] is started. In the evaluation branch [7], however, the scan is suspended at a suspension point S_2 so that the scan of the evaluation branch [12] is started next. Similarly, the scan of an evaluation branch [12] is suspended at the suspension point S_3 and this time, an evaluation branch [13] is scanned. Then, a character code C_8 is obtained together with the evaluation point. In this case, the character codes C_3 and C_8 are obtained as candidate characters. Such an evaluation scan is carried out for all the characters of the recognition dictionary and a character code having the best evaluation point is finally selected. If the suspension point S_1 is not provided in Fig. 5, then the character codes C_1 and C_2

are also to be obtained together with the evaluation point, respectively.

The logic structure of the recognition dictionary and the scan thereof, which are the most specific features of the present invention, will be described next.

Citation branching

A portion of the stokes of the Kanji character includes a portion which has a common characteristic relative to various Kanji characters such as radicals "Hen," "Tsukuri" or partial portions thereof. Therefore, these commonly usable characteristic evaluation commands (hereinafter, referred to as citing descriptions) are extracted and stored at one place in the recognition dictionary 3. Then, the citing description is called from a number of character evaluation descriptions.

Fig. 6A shows a part of the logic structure of the recognition dictionary 3. As shown in Fig. 6A, a preceding branch a is followed by a citation start mark QS and position data QA indicating the position of a citing description branch at a point necessary to the citation. A return code QR indicating the end of the citation is assigned at the end of the citing description branch Q_1.

The form of the position data QA indicating the address of the citing description might be an absolute address, a relative address or a citing description number which produces an address from a table.

When the citation start mark QS is determined during the sequential scan, the position data at the starting portion of the succeeding branch b is saved (or pushed) in the scan continuation stack A_{33} of the work area A_3 of the central unit 100. Then, a position information address of the citing

description branch Q_1 is obtained from the position data QA and is set in the dictionary scan point pointer A_1 of the central unit 100, whereby the sequential scan is continued along the citing description branch Q_1.

When the return code QR is obtained at the end of the citing description branch Q_1 while the sequential scan is executed along the citing description branch Q_1, the position information of the succeeding branch b previously saved (or pushed) in the scan continuation stack A_{33} of the central unit 100 is derived (or popped) and is set in the dictionary scan point pointer A_1. Thereafter, the sequential scan is continued along the succeeding branch b.

Thus, the scan in this case becomes effectively as shown in Fig. 6B, whereby the scan is executed in the sequential order of the preceding branch a, the citing description branch Q_1 and the succeeding branch b.

Tree branching

In various kinds of characters, there are some whose sub-strokes have a common characteristic such as a radical [Hen] and so on. In order to recognize the hand-written character, the input hand-written stroke is compared and evaluated with characteristics of all the characters stored in the dictionary and a character whose characteristic is sufficiently identical with the stroke of the hand-written character is selected therefrom. When the hand-written character is compared with many kinds of characters, if the common portion such as a radical is compared and evaluated only one time and the compared result is stored in the memory, the comparison of the common portion of the succeeding characters having the same common portion becomes

unnecessary and hence can be omitted. Thus, only the different portion need be evaluated.

Tree branching processing is based on the above-mentioned idea. Using tree branching processing, the save areas B_1 to B_W are provided in the unit 2 so as to save the partial evaluated results from the evaluation start to the node where an evaluation of non-common portion of a character occurs. Before starting the sequential scan of the branches [2], [3] and [4] shown in Fig. 5, the condition of the unit 100 immediately before the sequential scan and the succeeding scan re-start point branches (starting positions of the branches [7], [6] and [5]) are each saved as a set, thereby realizing a recursive multi-branching scan.

Fig. 7A shows the logic structure of the recognition dictionary 3. As shown in Fig. 7A, following the preceding branch a, a tree branching start mark TS for indicating a tree branching node is located at a node where tree branching becomes necessary. Then, position data TA indicating the starting position of the next tree branch T_{j+1} (j= 1 to 3 in Fig. 7A), the tree branch T_j, the end code TE and the character code C_j are provided as a set and a plurality of these sets are provided. (In this embodiment three sets in Figs. 7A-7B).

The expression form of the position data TA is the length of the branch T_j, the absolute address of the succeeding branch T_{j+1} and so on. As shown in Fig. 7A, the character code C_3 of the last tree branch T_3 is followed by the end code ET which indicates the end of a tree branch group, and the position data TA_3 of the tree branch T3 is data indicating the position of the end code ET.

The end code ET indicates the completion of all evaluations about one character, and the character

code C_j is a type of code which can correspond to the evaluated result up to the tree branch C_j, for example, a JIS (Japanese Industrial Standards) code.

When the tree branching start code TS is discovered in the sequential scan, all information of the work area A_3 (evaluated results of the extracted data scan position pointer A_{31}, the value of the evaluated result A_{32} and the scan continuation stack A_{33}) and the position information of the succeeding branching calculated from the position data TA are transferred to and saved in the save area B_i pointed out by the evaluation save area pointer A_2 in the central unit 100, and then the sequential scan is continued.

The save area B_i, pointed by the evaluation save area pointer A_2 of the unit 100, proceeds to the adjacent save area each time the evaluated results are transferred and saved therein.

When the scan reaches the end code TE in the sequential scan, all of the value of the evaluated result in the save area B_i and the character code C_j following the end code TE are provided as a set, which is output as a new candidate character.

Further, when a character is evaluated or when a suspension of scan is carried out due to the degraded value of the evaluated result, of the latest contents saved in the save area B_i, the corresponding information in the work area A_3 is returned to the work area A_3 of the unit 100 and the re-start branch position information is returned to the dictionary scan point pointer A_1 of the unit 100. By this processing, the condition of the work area A_3 of the unit 100 is returned to the condition presented just before the branching starts, and the scan point is brought about at the beginning of the succeeding branch.

When the save area B_i is returned to the unit 100, the save area pointed out by the save area pointer A_2 is simultaneously shifted by one node to the root.

When the scan point reaches the next branching node, one of the following three processes is selected on the basis of the code of the dictionary 3 indicated by the dictionary scan point pointer A_1:

i. The code indicates position information of the succeeding branch (such as TA_2, TA_3):

Under this condition the same processing as the branching scan is executed. Namely the condition of the work area provided just before the branching and the next branching re-start position information are saved and then the sequential scan is executed.

ii. Data indicates the end of the tree branch group (TA = ET):

Under these circumstances the processing is executed from the start of the re-start scan. According to this processing, the node is returned by one step.

iii. The data indicates the end of the whole tree of the dictionary 3 (in the case of Fig. 5, the branch [13] is ended):

The evaluation dictionary scan is finished.

Of the evaluated characters outputted, only the character having, for example, the best evaluated result is used. That is, Fig. 7A becomes substantially as shown in Fig. 7B, wherein respective tree branches T_1, T_2, T_3 are scanned after the preceding branch a.

Parallel branching

When a characteristic of one portion of one character is described in any one of the branches b_1, b_2, ..., b_n as shown in Fig. 10, the characteristic description becomes a preceding common part, a

parallel part and a succeeding common part. Parallel branching is roughly classified to provide the following two cases depending on whether the stroke portions of the character described by the branches b_1, b_2, ..., b_n correspond to the evaluation of the same portion or not:

i. Locally parallel branching (Figs. 8A-8B)

When the branches b_1, ..., b_n relate to the same portion of the hand-written stroke, accordingly, when the extracted data scan points are invariable such as b_1, ..., b_n in the start of the succeeding branch, this is referred to as "locally parallel branching." The locally parallel branching may be included in a desired branch. Thus, locally parallel branching may be included in one parallel branch b_i which is one of the locally parallel branches, and may be subjected to recursive citation.

ii. Parallel tree branching (Figs. 9A-9B):

For the case when the branches b_1, ..., b_n describe different amounts of extracted data, such as when the branch b_1 describes the portion of two sub-strokes of the square style forming the real prototype of the Japanese character (this square style is known in Japanese as Kaisho) and b_2 describes the above two-strokes in the cursive or semi-cursive style, the extracted data scan point upon the start of the succeeding part is changed. This branching is referred to as "parallel tree branching," wherein the starting point of the parallel branching has to be processed as the tacit branching starting point.

The branch, which can include parallel tree branch, is restricted by a certain condition. This restriction is to be understood such that the "parallel branch of the locally parallel branch must not include a parallel tree branch from a semantics standpoint." This restricts the dictionary

description by a certain degree. As will be described later, if the dictionary 3 does not satisfy the above restrictions, it is necessary to have a function to detect that the dictionary 3 satisfies the afore-noted restrictions. Upon the detection thereof, the dictionary 3 is incorrect, and the scan is ended.

The locally parallel branching and parallel tree branching will now be described in detail.

Locally parallel branching

As, for example, shown by the logic structure of the dictionary 3 in Fig. 8A, following the preceding branch a, a locally parallel branching start mark PS is located at a point (the starting portion of the locally parallel branching) in which the locally parallel branching becomes necessary. Then, similarly to the data TA, T_j and TE in the case of tree branching, position data PA, parallel branch P_j and end code PE are grouped as a set and, in this case, there are provided a plurality of the above sets (three sets in Fig. 8A).

The end code PE succeeding to the last parallel branch P_3 is followed by the end code EP which indicates the end of the parallel branching group, and the position data PA_3 of the parallel branch P_3 is employed as data indicative of the position of the end code EP. Further, the end code EP is followed by the succeeding branch b.

Since the length of the parallel branch P_j is known from the position data PA, it may be possible that without providing the end code PE, the scan position pointer A_1 checks that the scan reaches the starting portion of the succeeding parallel branch.

When the locally parallel branching code PS is discovered in the sequential scan, the stack condition proceeds to the next stack condition

(incremented by one step). Then, the stack condition A_{411} of the locally parallel stack A_4 is shifted to the next one (incremented by one step), and data indicating the result of failure is temporarily saved in the result store memory A_{412} indicated by the stack condition A_{411}. Thereafter, processing similar to the scan in the tree branching (save and re-start point set) is executed and then the sequential scan is continued.

When in the sequential scan of the parallel branch the locally parallel branching terminating condition is established or the scan suspension condition is established due to the degraded value of the evaluated result, the temporarily-evaluated result saved in the result store memory A_{412} and the better evaluated result of the present result store memory A_{32} of the unit 100 are again saved in the result store memory A_{412}. Then, the save area B_i is copied to the unit 100 and recovered and the processing proceeds to the next parallel branch.

In this case, the processing is branched into the following two processes depending upon the code pointed out by the dictionary scan point pointer A_1 which was determined by the preceding process:

i. In the case of the succeeding parallel branch (in the case of PA_2 and PA_3):

The sequential scan is continued immediately after the position data PA.

ii. In the case of the completion of the parallel branch (in the case of PA = EP):

The content stored in the result store memory A_{412} is shifted to the extracted data scan point pointer A_{31}. Then, the condition of the stack condition pointer A_{411} is returned by one step and the following completion processing of the local parallel branch which will be described next is carried out.

This completion processing is branched to provide the following two processes depending on the result stored in the result store memory A_{32}.

i. Suspension of scan.

A suspension of scan is executed when all of the parallel branches have failed and the evaluated result in the result store memory A_{32} indicates the failure, and the processing when the scan suspension condition is established is executed.

ii. Continuation of scan.

This processing is executed when there exists some of the parallel branches which reach the successful level and the evaluated result in the result store memory A_{32} indicates the successful result, and the sequential scan is continued just after the position data PA (= EP).

Accordingly, the evaluation scan in this case substantially becomes as shown in Fig. 8B, wherein after the preceding branch a, the parallel branches P_1, P_2, P_3 are scanned and the succeeding branch b is then scanned. With respect to the evaluated result, one of the best evaluated results of the branches P_1, P_2 and P_3 is saved.

Parallel tree branching

Fig. 9A shows the structure of the dictionary 3. As shown in Fig. 9A, next to the preceding branch a, a parallel tree branching mark BS indicating parallel tree branch and the position data BN indicating the starting position of the succeeding branch b are located at the points in which parallel tree branching occurs. Then, similarly to the locally parallel branching, the position data BA, the parallel branch B_j and the end code BE are provided as one set, and in this case, there are provided a plurality of (three sets in Fig. 9A) these sets. Then, the end code EB of parallel tree branch group

is located at the end and this end code EB is followed by the succeeding branch b.

When parallel tree branching mark BS is discovered in the sequential scan, the position data BN of the succeeding branch b is saved in the scan continuation stack A_{33} of the unit 100, and thereafter, processing similar to that of the aforementioned tree branching scan is executed.

Accordingly, in this case, the position data of the succeeding branch b is doubly saved both in the scan continuation stack A_{33} of the unit 100 and the continuation stack saved in the save area B_i.

In order to understand that the scan point reaches the end of the save area B_i of parallel tree branching in the sequential scan, it is proposed to prepare the end code BE in the dictionary description or to check whether the scan point pointer A_1 reaches the position of the position information of the re-start tree branching saved in the save area B_i or not.

When the end condition of parallel tree branching B_j is established in the sequential scan of parallel tree branching B_j, the same processing as the citation return scan is executed. In this case, however, since only the scan continuation stack A_{33} of the unit 100 is derived and the continuation point stack saved in the save area B_i is not changed, the position information of the succeeding branch b remains.

Therefore, the evaluation scan in this case substantially becomes as shown in Fig. 9B.

Fig. 11A shows an example of a logic structure of the dictionary 3 to which the present invention is applied, and Fig. 11B shows the substantial scan order thereof. In Figs. 11A and 11B, reference letters a to k designate branches.

According to the present invention, as shown in Figs. 11A and 11B, citation branching can be branched and also, the branching can be further branched. In order to realize the above-mentioned structure, the scan condition save unit is provided, wherein as shown, for example, in the scan [4] of Fig. 11B, even at the completion of citation branching, the scan is re-started in citation branching.

As will be apparent from Figs. 11A and 11B, the logic structure of the dictionary 3 can be made complex, whereby the dictionary 3 can be made compact in size and the recognition can be carried out at high speed.

As described above, according to the present invention, the size of the recognition dictionary 3 can be considerably compressed. That is, the recursive use of the citation description is made possible with the result that the dictionary 3 can be compressed. In the citation description, parallel tree branching can be used, thus allowing compression of the dictionary 3. Further, since parallel tree branching can be used again in other parallel tree branching, the dictionary 3 can be further compressed.

Furthermore, since common portions such as a radical (Hen) and the like in which the hand-writing is started is evaluated once, the time for recognition can be reduced. In other words, since locally parallel branching is used and the succeeding branch is evaluated once, the dictionary can be compressed in size and also the time for recognition can be reduced.

CLAIMS

1. A data input apparatus for a data processing device, the apparatus being operative to respond to hand-written script on a tablet to produce electronic data corresponding to the textual content thereof and comprising:

a tablet and a tablet pen for generating sequential characteristic data from stroke data of a hand-written character;

a scan control central unit for receiving said sequential characteristic data;

a recognition dictionary memory organized in a tree and branch structure for electronically storing character data describing characteristics of characters, the tree structure of the dictionary means including at least citation branching, parallel branching and tree branching, the parallel branching being classified into parallel tree branching and locally parallel branching;

evaluation scan control means for scanning the tree and branches of the dictionary memory to search for character data corresponding to the hand-written pattern information data; and

save means for storing the conditions of the evaluation scan control means, wherein at least the conditions of the scanning existing at the beginning of each branch are saved in the save means upon branching in order to allow recursive use of branches of the character data stored in the dictionary means.

2. A data input apparatus according to claim 1, wherein the

evaluation scan control means is adapted to scan a successive branch after saving the best evaluation result when scanning each branch during locally parallel branching.

3. A data input apparatus constructed and arranged to operate substantially as hereinbefore described with reference to and as illustrated in the accompanying drawings.

REGISTER ENTRY FOR GB2222476

Form 1 Application No GB8919522.6 filing date 29.08.1989

Priority claimed:
30.08.1988 in Japan - doc: 63215572

Title A HAND-WRITTEN, ON-LINE CHARACTER RECOGNITION APPARATUS AND METHOD

Applicant/Proprietor
SONY CORPORATION, Incorporated in Japan, 7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo 141, Japan [ADP No. 00732651001]

Inventors
KIMIYOSHI YOSHIDA, SONY CORPORATION, 7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo, Japan [ADP No. 02301299001]

AKIO SAKANO, SONY CORPORATION, 7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo, Japan [ADP No. 03143351001]

HIROFUMI TAMORI, SONY CORPORATION, 7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo 141, Japan [ADP No. 04347449001]

MASAYUKI CHATANI, SONY CORPORATION, 7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo 141, Japan [ADP No. 04347423001]

Classified to
G4R
G06K

Address for Service
J A KEMP & CO, 14 South Square, Gray's Inn, LONDON, WC1R 5LX, United Kingdom [ADP No. 00000026001]

Publication No GB2222476 dated 07.03.1990

Examination requested 29.08.1989

Patent Granted with effect from 21.07.1993 (Section 25(1)) with title A DATA INPUT DEVICE FOR A DATA PROCESSING DEVICE

10.05.1993 Application under Rule 110(4) for extension of period(s) prescribed by Rule(s) 34 filed on 19.02.1993
Entry Type 27.1 Staff ID. AGB1 Auth ID. F52

10.05.1993 Period prescribed by rule(s) 34 extended under Rule 110(6) in accordance with decision of the Comptroller dated 10.03.1993 and following the filing of F53/77 and the prescribed fee on 28.04.1993.
Entry Type 27.3 Staff ID. AGB1 Auth ID. F53

**** END OF REGISTER ENTRY ****

RENEWAL DETAILS

PUBLICATION NUMBER GB2222476

PROPRIETOR(S)

Sony Corporation, Incorporated in Japan, 7-35 Kitashinagawa 6-chome, Shinagawa-ku, Tokyo 141, Japan

DATE FILED 29.08.1989

DATE GRANTED 21.07.1993

DATE NEXT RENEWAL DUE 29.08.1995

DATE NOT IN FORCE

DATE OF LAST RENEWAL 19.08.1994

YEAR OF LAST RENEWAL 06

STATUS PATENT IN FORCE