

(19)



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) Publication number:

**0 345 088 B1**

(12)

## EUROPEAN PATENT SPECIFICATION

(45) Date of publication of patent specification: **10.11.93** (51) Int. Cl.<sup>5</sup>: **B41J 29/36**

(21) Application number: **89305597.0**

(22) Date of filing: **02.06.89**

(54) **Character processing apparatus.**

(30) Priority: **03.06.88 JP 136786/88**

(43) Date of publication of application:  
**06.12.89 Bulletin 89/49**

(45) Publication of the grant of the patent:  
**10.11.93 Bulletin 93/45**

(84) Designated Contracting States:  
**DE FR GB IT**

(56) References cited:  
**EP-A- 0 228 292**  
**EP-A- 0 256 633**  
**GB-A- 2 156 559**

(73) Proprietor: **CANON KABUSHIKI KAISHA**  
**30-2, 3-chome, Shimomaruko,**  
**Ohta-ku**  
**Tokyo(JP)**

(72) Inventor: **Kawanabe, Tetsuya**  
**15-3-308, Kawaiyuku-cho**  
**Asahi-ku**  
**Yokohama-shi Kanagawa-ken(JP)**

(74) Representative: **Beresford, Keith Denis Lewis**  
**et al**  
**BERESFORD & Co.**  
**2-5 Warwick Court**  
**High Holborn**  
**London WC1R 5DJ (GB)**

**EP 0 345 088 B1**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid (Art. 99(1) European patent convention).

## Description

### BACKGROUND OF THE INVENTION

#### Field of the Invention

The present invention relates to a character processing apparatus and, more particularly, to an electronic typewriter in which recorded characters or the like can be deleted on a word unit basis.

#### Related Background Art

A conventional electronic typewriter has a printing apparatus and a keyboard input apparatus. The keyboard input apparatus has various kinds of function keys such as character input keys, space key and back space key to feed a printer head to the right and left in the horizontal direction, return key, index key, and reverse index key to feed a recording paper upward and downward in the vertical direction, delete key to delete recorded characters or the like when they were erroneously input or the like, etc.

On the other hand, in recent years, the electronic typewriter has been electronized more and more. There has been provided an electronic typewriter having the back trace function in which by storing recorded characters or the like into memory means such as memory or the like, a printer head is returned to a plurality of preceding lines of the recorded characters or the like on a character or word unit basis. Further, there has been provided an electronic typewriter having the word delete function in which characters or the like recorded in the same line are deleted on a word unit basis.

In such an electronic typewriter, the operator makes a desired document or the like by operating the keyboard.

For instance, to make a sentence as shown in Fig. 13A, the key input operations are executed in accordance with an order as shown in Fig. 13B.

In Fig. 13B, SP denotes a space key and RTN indicates a return key. By inputting the return key, a recording paper is fed by a predetermined amount and the printer head is moved to the position of y in Fig. 13A.

It is now assumed that the operator becomes aware of a mistake of the input of the words shown by a region x in the diagram and tries to correct "TYPEWRTER." to "TYPEWRITER."

The first method of correction is shown in Fig. 14A. First, the printer head is moved from y to y' in Fig. 13A by returning the recording paper by a predetermined amount by performing the reverse index key input (RIX). Next, by executing the space key input (SP) a predetermined number of times, the printer head is moved from y' to z in Fig. 13 By

executing the delete key input (DEL) four times, the recorded characters are deleted in accordance with the order of ".", "R", "E", and "T". The character key input is then executed in accordance with the order of "I", "T", "E", "R", and "." so as to obtain the correct word spelling. In this manner, the correction is executed.

The second method is shown in Fig. 14B and is applied to the case of the apparatus having the back trace function mentioned above. First, the printer head is moved from y to z in Fig. 13A by performing the back trace key input (BTR). The recorded characters ".", "R", "E", and "T" are deleted by executing the delete key input (DEL) four times in a manner similar to the above. Then, characters "I", "T", "E", "R", and "." are retyped. In this manner, the correction is executed.

The third method is shown in Fig. 14C and is applied to the case of the apparatus having the word delete function mentioned above. In a manner similar to the above, the printer head is moved to z in Fig. 13A by performing the back trace key input (BTR). "TYPEWRTER." are all deleted by executing the word delete key input (WDEL) and "TYPEWRITER." are retyped. In this manner, the correction is executed.

However, when comparing the number of key input times which are required to correct the sentence shown in the example, there is a drawback such that in the case of the first method, its value becomes maximum and the operations are complicated. According to the second method using the back trace function as the improvement of the first method, although the operation to move the printer head to the deletion start position z is simplified, the number of key input times which are required to delete is the same as that in the first method. In addition, as in the example, in order to correct "TYPEWRTER." to "TYPEWRITER.", the operator must count the position where he wants to input "I", so that there is a drawback such that an erroneous operation such that the recorded characters are erroneously deleted too much or the like can easily occur.

The third method using the word delete function has been considered as the improvement of such a drawback. However, although the operability with respect to the deletion of the recorded characters was improved, all of the words are deleted by the single operation. Therefore, in the example, the portion of "TYPEWR" in "TYPEWRTER." is additionally deleted, so that "TYPEWR" must be again input and there is a drawback such that the number of key input times which are required for correction is larger than that in the second method.

On the other hand, according to the third method, since the number of deleting times of the recorded characters is large, the operating time

which is required for deletion, a use amount of a deleting ribbon, a use amount of a recording ribbon when retyping, and the like increase. Thus, there are drawbacks such that the correcting time (throughput) increases and the running cost also increases.

In addition, even in any of the conventional methods mentioned above, in spite of the fact that the operator wants to correct the words which have just previously been recorded, he needs to execute other operations of the reverse index key (RIX), space key (SP), back track key (BTR), and the like which are not directly concerned with the deleting operation. Consequently, there is a drawback such that the correcting operation cannot be easily understood.

GB-A-2156559 discloses an electronic typewriter having a movable print point. It has a word delete function such that if the word delete function is operated while the print point is not at the position of a printed character, the typewriter determines whether there are any printed characters in the current print line to the left of the position of the print point. If there are, the print point is moved leftward to the nearest such character, which is then deleted together with all other characters in the same word. If the word delete function is operated when the print point is at the position of a printed character, that character is deleted together with all other characters in the same word, whether to the right or the left of the character which is at the position of the print point when the word delete function is operated. If the word delete function is operated a plurality of times, the same number of words will be deleted from the current print line.

#### SUMMARY OF THE INVENTION

The invention provides apparatus for receiving, outputting and deleting characters as set out in claim 1. The remaining claims set out optional features.

An embodiment of the invention provides a character processing apparatus in which the reduction in running cost and the improvement in throughput in the word deleting operation can be realized and the operability regarding the word deletion is improved.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is an external view of a typewriter according to an embodiment;

Fig. 2 is a control block diagram of the typewriter of the embodiment;

Fig. 3 is a diagram for explaining a control logic circuit;

Fig. 4 is a diagram for explaining a keyboard of the typewriter of the embodiment;

Figs. 5A, 5B, 5C, 6A, 6B, 7A, and 7B are diagrams for explaining print character memory means;

Fig. 8 is a flowchart for the word deleting process of the embodiment-1;

Figs. 9A and 9B are diagrams for explaining the key input in the word deleting operation;

Fig. 10 is a diagram for explaining the problems in the embodiment-1;

Fig. 11 is a flowchart for the word deleting process of the embodiment-2;

Fig. 12 is a flowchart for the key input process in the embodiment-2; and

Figs. 13A, 13B, 14A, 14B, and 14C are diagrams for explaining conventional examples.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

An embodiment of the present invention will be described in detail hereinbelow with reference to the drawings.

[First embodiment]

Fig. 1 is an external view of an electronic typewriter as a character processing apparatus to which the invention can be applied. In the diagram, reference numeral 1 denotes a keyboard on which character input keys and various kinds of function keys are arranged. Print data and various kinds of instruction data are input by the keyboard 1.

Reference numeral 2 denotes what is called a daisy wheel printer mechanism. A carriage 4 on which a printer head 3 is mounted can be moved in the longitudinal direction of a platen by a carrier motor (not shown). The carriage 4 has a wheel motor (not shown) which can select a desired type by rotating a type wheel 5 and a feed motor (not shown) to take up a print ribbon and a correction ribbon by predetermined amounts. The printer head 3 has a hammer (not shown) to print onto a paper through an ink ribbon by hitting the back surface of a desired type. When deleting a desired printed character, the carriage 4 is moved to the position of the printed character and, thereafter, the type of the type wheel 5 corresponding to the printed character to be deleted is selected by the wheel motor. The correction tape is set to the height of the type surface and the back surface of the type is hit. Thus, the printed character is peeled off or the white ink is coated onto the printed character, thereby deleting the character. Reference numeral 6 denotes a platen around which a paper 7 is wound and which is rotated by a paper feed motor (not shown) or the like and

feeds the paper. The platen 6 also functions as a recording base to receive the shock of the type by the hammer.

The printer mechanism 2 in the embodiment is operated by a control circuit, which will be explained hereinafter.

The control circuit of the above apparatus will now be described.

Fig. 2 is a control block diagram and Fig. 3 shows a detailed construction of a control logic circuit 11 in Fig. 2.

The control logic circuit 11 comprises an MPU (microprocessing unit) 30, an ROM (read only memory) 31, an RAM (random access memory) 32, a timer 36, and the like. These components are connected by an MPU bus.

With the above circuit construction, the MPU 30 executes the arithmetic operation control in accordance with micro instructions which have previously been stored in the ROM 31, receives the key input information from an input control circuit 9, and executes the input/output control with a print control circuit 13. Under the control of the MPU 30, the timer 36 generates reference time information, measures the elapsed time, generates an interruption request (INT) according to the time control condition, or controls the control logic circuit 11 in a real-time manner. Each of control flowcharts, which will be explained hereinafter, for instance, programs shown in Figs. 8, 11, and 12 are stored in the ROM 31. The RAM 32 has work areas such as area for a key buffer 34 to temporarily store the key information input from the keyboard 1, area for a correction buffer 33 to store the characters or the like which were displayed and/or recorded, area for a buffer 35 of a pointer 37, a space amount 38, an E-flag, and R-flag, and a register which are necessary for the program control, and the like.

Contact devices which are interlocked with the keys which are operated by the operator are arranged on the keyboard 1 in a matrix form. The input control circuit 9 retrieves the input state of the keyboard 1 every predetermined period (5 msec) through a keyboard control line 8 by using the timer 36 and the like. When a key input is detected, the position on the matrix where the input key exists is examined and is converted into the microcode or the like. The key input data is supplied to the MPU 30 through an input interface 10. The MPU 30 analyzes the key input data and stores into the key buffer 34 in the RAM 32.

Since such an operation is executed by interrupting means or the like, it is certainly performed even during the execution of the print control, which will be explained hereinafter. Even if the key input is executed at a high speed, no key input data will be lost. On the other hand, the key buffer 34 has what is called a ring buffer structure and

has the FIFO function.

When the key input data is stored in the key buffer 34, the MPU 30 sequentially reads out the key input data and discriminates whether the input key is the character key or one of the various kinds of function keys. If the input key indicates the printing operation or the like, the MPU 30 controls the operation of actuators such as carrier motor, wheel motor, and the like mentioned above through the print control circuit 13 comprising a motor driver and the like and makes the printer mechanism 2 operative, thereby executing the print control such as printing of a desired character, deletion of the character, or the like. To enable the characters to be automatically deleted by executing the delete key input or word delete key input, the printed characters and the like are stored into the correction buffer 33.

The keyboard 1 will now be described with reference to Fig. 4.

The keyboard 1 comprises character keys 40 and various kinds of function keys and the like. By pressing the character key 40, the character written on the upper surface of the key depressed can be input and printed.

Reference numeral 41 denotes a space key to input a blank between words or to instruct to shift the carriage 4 to the right; 42 indicates a back space key to instruct to shift the carriage 4 to the left; 43 a delete key to instruct to erase the nearest printed character; 44 a word delete key to indicate to delete the nearest printed characters on a word unit basis; and 45 a code key to instruct to extend the function. By inputting other function key or character key with the code key 45 depressed, the depressed function key or character key functions as a sole function key. Therefore, many functions can be added without increasing the number of keys to the typewriter keyboard. Reference numeral 47 denotes a return key to instruct the carriage return and 46 indicates a shift key to instruct the switching between the capital letter and the small letter and between the numeral and the symbol. On the other hand, the shift key 46 can further instruct many functions in association with the code key 45.

The printed character storage control will now be described with reference to Figs. 5A to 5C.

The printed character storage control is executed by using memory means such as correction buffer 33, buffer 35, and the like in the RAM 32. The buffer 35 has memory areas for the pointer 37 and space amount 38.

Fig. 5A is a diagram for explaining the states of the correction buffer 33 and pointer 37 upon initialization by the power-on of the apparatus or the like. In the diagram, n indicates a head address in the correction buffer 33 in the RAM 32 and m repre-

sents an area size in the correction buffer 33. First, a buffer start code (BG) indicative of the head in the correction buffer 33 is stored into address n. A buffer end code (BE) indicative of the end in the correction buffer 33 is stored into address  $n+(m-1)$ , that is, the end address in the correction buffer 33. Then, n is set into the pointer 37 to make the data from address  $n+1$  to address  $n+(m-2)$  invalid. In the embodiment, the data from the address indicated by the pointer 37 to the buffer end code (BE) is handled as invalid data.

Fig. 5B shows the result in which "I", "T", "space key", ... were sequentially input by the keyboard 1 of the apparatus and printed and output. Fig. 5C shows the states of the correction buffer 33 and pointer 37 at that time. The character printing and the storage of the printed characters will now be described.

First, the correction buffer 33 and pointer 37 are set into the foregoing states. The space amount 38 is set to "0". The printer head 3 is located at a in Fig. 5B. When "I" is input by the keyboard 1, the printer mechanism 2 is made operative and "I" is printed at the position a in Fig. 5B on the recording paper 7. Then, the printer head 3 is shifted by a predetermined amount and is located at the position b in Fig. 5B. At this time, the pointer 37 is advanced by (+1). The character code "I" is stored in address  $n+1$  in the correction buffer 33. A feed amount of the printer head 3 is stored into the space amount 38. When "T" is then input, in a manner similar to the above, "T" is printed, the printer head is fed, the pointer 37 is advanced to  $n+2$ , the character code "T" is stored into address  $n+2$  in the correction buffer 33, and the feed amount of the printer head 3 is set into the space amount 38. When "space" is then input, the printer head 3 is shifted by a predetermined amount and this amount is added to the space amount 38 and the new amount is stored. When "I" is subsequently input, "I" is printed at the position c in the diagram and the printer head is moved. Since the space amount 38 is larger than the feed amount by one digit, the pointer 37 is advanced to  $n+3$ . The space amount code (SP) is stored into address  $n+3$ . At this time, the space amount indicates the distance from "T" to "I". The pointer 37 is advanced to  $n+4$ . The character code "I" is stored into address  $n+4$ . The feed amount of the printer head 3 after "I" was printed is set into the space amount 38. The control similar to the above is executed in accordance with the order of "S", "space", "T", "H", "E", "space", "K", .... When the "return key" is input, the carriage return is executed and the printer head 3 is moved to the position d in the diagram. At this time, the pointer 37 is advanced to  $n+33$ . In order to obtain the printing position (e in the diagram) of "." (period)

stored in address  $n+32$  as an absolute position of the printer head 3, the space amount 38 is subtracted from the absolute position information of the printer head 3. The resultant value is stored as a position code (CP) into address  $n+33$ . The pointer 37 is advanced to  $n+34$ . The line end code is stored into address  $n+34$ . The space amount is set to "0".

As mentioned above, in the embodiment, the printed characters are stored, thereby making the result of the printing shown in Fig. 5B correspond to the correction buffer 33 shown in Fig. 5C.

The word delete control will now be described.

When the word delete key 44 is input by the keyboard 1 of the apparatus in the state described in Fig. 5C, that is, under the control state in which the printer head 3 exists at the position d in Fig. 5B, the pointer 37 indicates address  $n+34$  in the correction buffer 33, and the line end code (DL) is stored in address  $n+34$ , the content in address  $n+34$  in the correction buffer 33 which is indicated by the pointer 37 is first examined, thereby checking that the input data indicates the line end code (DL) instead of the character code. Since the input data is the line end code (DL) in this case, the recording paper 7 is returned in the vertical direction by a predetermined amount and the printer head 3 is set to the vertical position of the first line. The pointer 37 is moved back by (-1) and the content in address  $n+33$  in the correction buffer 33 is examined. Since it indicates the position code (CP), the printer head 3 is moved to the absolute position indicated by the data of the position code. The space amount 38 is set to "0". The pointer 37 is moved back to indicate address  $n+32$ . By the above operation, the printer head 3 is moved to the position e corresponding to "." (period) shown in Fig. 6A. On the other hand, the correction buffer 33 and pointer 37 in such a state are as shown in Fig. 6B. Due to this, in the embodiment, when the word delete key 44 is input, if the printer head 3 does not exist at the position over the printer character, the printer head 3 is located to the nearest printed character. Therefore, even after the line feed was executed by the return key 47 or the like, the printer head 3 can be easily located to the final printed character or the end of the word. On the other hand, since the printer head 3 is ordinarily located at the position next to the final printed character, an inconvenience such that when the word delete key 44 is erroneously input, the word which does not need to be deleted is deleted is prevented.

When the word delete key 44 is input in the above-mentioned state, that is, the state in which the printer head 3 exists at the position e in Fig. 6A, the pointer 37 indicates address  $n+32$ , the content of the correction buffer 33 is as shown in

Fig. 6B, and the space amount 38 is set to "0", the pointer 37 indicates address  $n+32$  and the character code "." (period) is set in address  $n+32$  in the correction buffer 33. Therefore, the word deleting operation is executed. In other words, to delete "." (period), the type "." (period) of the type wheel 5 is selected and hit through the correction tape. The printed character "." (period) on the recording paper 7 is peeled off or the white ink is coated onto the printed character, thereby deleting the printed character.

The pointer 37 is moved back by (-1) and address  $n+31$  is indicated. The distance from the present printer head position (e in the diagram) to "N" stored in address  $n+31$  is stored into the space amount 38. Since address  $n+31$  in the correction buffer 33 indicated by the pointer 37 represents the character code "N", the printer head 3 is moved to the position over the printed character "N" on the basis of the movement distance information of the space amount 38. Then, the space amount 38 is set to "0". In a manner similar to the above, "N" is deleted, the pointer 37 is moved back by (-1) to indicate address  $n+30$ , and the distance from the printer head position to "I" in address  $n+30$  is stored into the space amount. When "J", "A", and "M" are deleted in a manner similar to the above, the pointer 37 indicates address  $n+26$  in the correction buffer 33. The content in address  $n+26$  indicates the space code (SP) instead of the character code. Thus, the word deletion is finished and a predetermined feed amount indicated by the space code (SP) is added to the space amount 38 and the resultant data is stored. After completion of the above operations, the recording paper 7 has a form as shown in Fig. 7A. The contents of the correction buffer 33 and pointer 37 are as shown in Fig. 7B. The printer head 3 is located at g in Fig. 7A. The distance from the present printer head position to the printed character "Y" is stored in the space amount 38.

When the word delete key 44 is again input, the pointer 37 indicates address  $n+25$  in the correction buffer 33. The content in address  $n+25$  is the character code "Y". However, the space amount 38 is not set to "0" and indicates that the printer head 3 does not exist over the printed character. Therefore, the word deletion is not executed. The printer head 3 is moved to the position (h in the diagram) over the printed character "Y" on the basis of the movement distance information of the space 38. The space amount 38 is set to "0" and the operation is finished. Consequently, according to the embodiment, even when the printer head 3 is located on the same line as the printed character, if the printer head 32 is not located over the printed character, the head 3 is moved to the nearest printed character.

That is, even in any case, if the printer head 3 is not located over the printed character when the word delete key 44 is input, the printer head 3 can be located to the position over the nearest printed character. Therefore, even in such a case, the word can be deleted by inputting the word delete key 44 twice. Thus, the word deletion can be executed without performing the complicated operations as in the conventional example.

The word deleting process which is executed by inputting the word delete key 44 mentioned above will now be described.

Fig. 8 is a flowchart for the word deletion processing program stored in the ROM 31 in the control logic circuit 11. This program is started by inputting a word delete instruction by the word delete key 44.

In the first step S1, a check is made to see if the content in a predetermined address in the correction buffer 33 indicated by the pointer 37 is the character code or not. If YES, step S2 follows and a check is made to see if the space amount 38 is set to "0" or not to thereby discriminate whether the printer head 3 exists over the printed character or not. If the head 3 is located over the printed character (space amount 38 = "0"), step S3 follows and the word deletion is started. In step S3, the printer mechanism 2 is activated and the printed character on the recording paper 7 is deleted on the basis of the printed character data stored in predetermined addresses in the correction buffer 33 indicated by the pointer 37. In step S4, the pointer 37 is moved back to delete the data of the erased printed character from the correction buffer 33. In step S5, the distance from the position of the printer head 3 to the stored character or the like indicated by the pointer 37 is obtained and stored into the space amount 38.

In the next step S6, if the stored data indicated by the pointer 37 is the character code, step S7 follows and the printer head 3 is moved back by only the distance stored in the space amount 38 in order to move the printer head 3 to the position over the printed character of the stored data indicated by the pointer 37. In step S8, the space amount 38 is reset to "0". The processing routine is again returned to step S3 and a desired character is deleted. The processes in steps S3 to S8 are repeated until the control code is detected in step S6. When the control code is detected in step S6, step S9 follows. If the pointer does not indicate the space code (SP), this means that the buffer start code (BG) or line end code (DL) has been detected. Therefore, the word deleting process is finished. On the other hand, if the pointer indicates the space code (SP), step S10 follows and the shift amount is added to the space amount 38 and the resultant value is stored. The space code (SP) is

deleted in step S11 and the word deleting process is finished.

On the other hand, if the space amount 38 is not "0" in step S2, this means that the printer head 3 does not exist over the printed character. Therefore, step S12 follows and the printer head 3 is moved back by only the distance stored in the space amount 38 and is located over the printed character. The space amount 38 is reset to "0" in step S17 and the word deleting process is finished. If the memory content of the correction buffer 33 indicated by the pointer 37 is the control code in step S1, step S13 follows. If the control code is not the line end code (DL) in step S13, the control code is the buffer start code (BG). Therefore, this means that no printed character exists, so that the word deleting process is finished. Since such a state relates to the case where the word delete key 44 was input when the character to be deleted does not exist, this state is regarded as an abnormal operation and a warning may be informed to the operator by a buzzer or the like.

On the other hand, if the control code indicates the line end code (DL) in step S13, step S14 follows and the platen 6 of the printer mechanism 2 is rotated to locate the printer head to the previous line. In step S15, the pointer 37 is moved back and the line end code (DL) is deleted. Step S16 then follows and the printer head 3 is moved to the absolute position shown by the position code (CP) indicated by the pointer 37. The printer head 3 is located over the printed character. In step S17, the space amount 38 is reset to "0" and the word deleting process is finished.

#### [Second embodiment]

In the foregoing apparatus of the embodiment, even if a correction word is input during the execution of the word deletion, the key data input by the keyboard 1 is temporarily stored into the key buffer 34, so that a problem such that the input key data is lost does occur. The key data stored in the key buffer 34 is not read out so long as the word deleting process is not finished.

The storage control of the key input will now be described with reference to Figs. 9A and 9B.

Fig. 9A is a diagram showing a state during the execution of the word deletion of a word "TYPEWRITER." (shown in a region i in the diagram) as a printed character string having the wrong spelling. The characters (ER.) shown as broken lines in the diagram denote that they have already been deleted. The printer head 3 is located at the position j in the diagram.

Fig. 9B shows printer mechanism control timings which are required for the word deleting operations such as deletion (D1) of ".", deletion (D2) of

"R", deletion (D3) of "E", ... by inputting the word delete key (WDEL) 44 from the keyboard 1. The operating times differ in dependence on the characters to be deleted or the like. On the other hand, the key input is retrieved at every predetermined period by using counting means in the timer 36 of the apparatus. When the key input is detected, the input key data is immediately stored into the key buffer 34.

Therefore, even during the deleting operation, the correction word such as "T" input (K1), "Y" input (K2), ... can be also input. Since the input characters or the like are stored in the key buffer 34, when the word deleting operation is finished, the characters stored in the key buffer 34 are sequentially read out in accordance with the storing order and a desired correction word "TYPEWRITER." is printed and output. In this manner, the input waiting time is reduced, the throughput is raised, and the smooth operation is provided.

However, when the word delete key 44 is input to delete a character string l "TYPEWRITER." in a state in which a sentence shown in Fig. 10 has been printed and the printer head 3 is located at P in Fig. 10, a character string k "ELECTRO...WRTER." is deleted because the operator forgot to input the space at the position Q in Fig. 10. Consequently, there is a problem such that a correct printed character string m "ELECTRONIC" is also unwillingly deleted. On the other hand, when the operator inputs "TYPEWRITER." after the word delete key 44 was input in order to correct "TYPEWRITER." to "TYPEWRITER." without becoming aware of the absence of the space at point Q, the printed result becomes an undesirable sentence such as "THIS\_IS\_TYPEWRITER.". In order to obtain a desired sentence of the operator, "TYPEWRITER." must be again deleted and "ELECTRONIC\_TYPEWRITER." must be input. There is a fear such that the operations become very complicated.

The above construction of the apparatus and the like are similar to the foregoing embodiment. The flag and register buffer area 35 in the RAM 32 has an E-flag 351 (not shown) and an R-flag 352 (not shown). Both of the E-flag 351 and the R-flag 352 are reset upon initialization of the power-on of the apparatus or the like.

Fig. 11 shows a flowchart for the word deletion processing program stored in the ROM 31 of the control logic circuit 11. This program is started when a word deletion instruction is input by the word delete key 44.

The contents of the processes in steps S1 to S17 in the flowchart of Fig. 11 are the same as those shown in Fig. 8 described in the foregoing embodiment.

If the pointer 37 indicates the character code in step S1, step S2 follows. If the printer head 3 exists over the printed character (space amount 38 = "0") in step S2, step S20 follows. In step S20, the E-flag 351 is set to "1" to make the operation to stop the word deleting operation valid by a key input process, which will be explained hereinafter. The following word deleting operation is started. In step S3, a desired printed character is deleted and step S4 follows and the pointer 37 is moved back and the deleted character data is deleted from the correction buffer 33. In step S5, the distance until the character to be deleted next is stored into the space amount 38. Then, step S6 follows. If the pointer 37 indicates the character code in step S6, a check is made in step S21 to see if the stop operation has been instructed or not by checking the R-flag 352. If NO, step S7 follows and the printer head 3 is moved to the position over the character to be deleted. The space amount 38 is reset in step S8. The processes in steps S3 to S8 are repeated until the control code (delimiter of a word) is detected in step S6.

On the other hand, if the R-flag 352 has been set in step S21, this means that the stop of the operation has been requested. Therefore, step S22 follows and the E-flag 351 and R-flag 352 are reset and the input of the stop operation is inhibited and the word deleting process is stopped. Since the stop discriminating step S21 is executed every deletion of the character, the operation can be stopped at an arbitrary position during the word deleting operation.

When the operation stop is not requested, the control code (delimiter of a word) is detected in step S6. The processing routine advances to step S9. If the pointer does not indicate the space code (SP) in step S9, the E-flag 351 and R-flag 352 are reset in step S22 and the word deleting process is finished. If the pointer indicates the space code (SP) in step S9, the space amount 38 is updated in step S10. Step S11 then follows and the pointer 37 is moved back and the space code (SP) is deleted. In step S22, the E-flag 351 and R-flag 352 are reset and the word deleting process is finished.

On the other hand, if the space amount 38 is not set to "0" in step S2, the printer head 3 is moved to the nearest printed character in step S12. Step S17 follows and the space amount 38 is reset. The word deleting process is finished.

On the other hand, if the pointer indicates the control code in step S1, step S13 follows. If the control code does not indicate the line end code (DL) in step S13, the word deleting process is finished. If the line end code (DL) has been indicated in step S13, the recording paper 7 is returned to the previous line in step S14. In step S15, the pointer 37 is moved back and the line end

code (DL) is deleted. Step S16 then follows and the printer head 3 is moved to the position indicated by the position code (CP). In step S17, the space amount 38 is reset. The word deleting process is finished.

The key input process will now be described.

Fig. 12 is a flowchart for a key input processing program stored in the ROM 31 in the control logic circuit 11. The state of the keyboard 1 is retrieved at every predetermined time by the interrupting process or the like on the basis of an output (INT2) of the counting means of the timer 36. When the key input is detected, the execution of the above program is started.

First, if the word deletion stop operation is invalid in step S30, that is, if the E-flag 351 is not set in step S20 in Fig. 11, step S31 follows. If a blank area exists in the key buffer 34, the input key data is stored into the key buffer 34 in step S32 and the key input process is finished. If the key buffer 34 is filled with data, the key input process is finished. At this time, the overflow of the key buffer can be also warned by a buzzer sound or the like.

On the other hand, if the E-flag 351 has been set in step S30, step S33 follows and a check is made to see if the word deletion stop operation is executed or not. The word deletion stop operation can be also executed by inputting an independent function key. However, in such a case, the number of keys on the keyboard 1 increases and there is a fear such that the cost of the apparatus rises. Therefore, the stop operation is executed by inputting a combination of the existing keys on the keyboard 1, for instance, by inputting the word delete key 44 with the code key 45 depressed.

If the stop operation is executed in step S33, step S34 follows and the R-flag 352 is set such that the stop is determined in the discriminating step S21 in Fig. 11. In the next step S35, the key buffer 34 is reset to delete the key data which was input during the word deleting operation and the key input process is finished. The problem in the foregoing embodiment can be solved by such a process. On the other hand, even if the key buffer 34 is filled with data, the key data of the stop operation is processed without being stored into the key buffer 34. Therefore, in the case of the slow operation such as a deleting operation, even if the key input operations had been executed at a high speed and the key buffer 34 over-flowed, the word deleting operation can be stopped.

According to the embodiment, in the case where one of the keys arranged on the same keyboard in a matrix form was solely input, the key input data is stored into the FIFO buffer. When the key input operations were performed by a combination of special keys, the key input data is not



stored into the FIFO buffer but the special instructing function can be provided.

On the contrary, if the stop operation is not indicated in step S33, this means that the key input such as a correction word or the like has been performed during the word deleting operation. Therefore, step S31 follows and if a blank area exists in the key buffer 34, the key input data is stored into the key buffer 34 in step S32 and the key input process is finished. Thus, a correction word or the like can be also input during the word deleting operation.

By the above operation control, the purpose of the embodiment such that the word deleting operation is stopped at an arbitrary position can be accomplished.

The effects shown in the foregoing embodiment are not obviously lost.

In the above two embodiments, the word delete key 44 has been used to instruct the word deletion. However, for instance, the word deletion can be also instructed by inputting the delete key 43 with the code 45 depressed.

On the other hand, if the foregoing special key combination input is applied to, for instance, the operation of the temporary print stop function, the hood-open sensor provided in the conventional recording apparatus such as a typewriter or the like can be omitted. The low-cost apparatus can be provided without reducing the functions of such an apparatus.

Although the embodiment has been described with respect to the word deletion, the invention can be also similarly applied to the deletion of one line. In such a case, it is sufficient that the deleting operation is finished by using the line end code (DL) and buffer start code (BG).

As described above, if the word deletion instruction is input when the printer head does not exist over the printed character, the printer head can be easily moved to the position over the nearest printed character to be deleted. On the other hand, even if the printed characters to be deleted exist on a plurality of lines rather than one line, the printer head can be easily moved to the position of a desired printed character to be deleted without operating a plurality of keys as in the conventional apparatus.

Further, there is an advantage such that a correction word or the like can be input even during the word deleting operation, so that the working time to correct a sentence can be reduced.

On the other hand, since the word deleting operation can be stopped at an arbitrary position, the deletion is not performed in vain in the case of the erroneous operation, so that the high efficient typing work can be provided.

Further, although what is called a daisy wheel printer has been used as a printing apparatus in the embodiments, the similar effect can be also obtained by applying the invention to other recording apparatuses such as a thermal copy transfer printer and the like.

## Claims

1. Apparatus for receiving character inputs, storing input character data in a memory means (32), and providing character outputs, the apparatus having a next character output position such that during normal operation if a character is input it will be output at the next character output position, and the apparatus having a delete function for deleting at least one output character, such that if the delete function is operated when the next character output position is at the position of a character which has already been output, that character at least is deleted, whereas if following a character input operation the delete function is operated when the next character output position is not at the position of a character which has already been output, the next character output position is moved to the position of the nearest preceding character which has already been output,
  - characterised in that
  - the nearest preceding character is not deleted when the delete function is operated when the next character output position is not at the position of a character which has already been output.
2. Apparatus according to claim 1, in which space amount data, indicating the distance between the next character output position and the position at which a character has been output, is stored in the memory means (32).
3. Apparatus according to claim 2, in which during a delete operation, the said space amount data is read and if it is non-zero it is determined that the next character output position is not at the position of an output character.
4. Apparatus according to any one of claims 1 to 3, in which input character data is stored in the memory means (32) as a data string interspersed with control data.
5. Apparatus according to claim 4 when dependent on claim 2 or claim 3, in which when space data is input, the space amount data is updated, and when character data is input following input of space data, control data (SP) is

stored in the data string representing the value of the space amount data.

6. Apparatus according to claim 4 or claim 5, in which the said control data comprises absolute position data. 5
7. Apparatus according to claim 6, in which when new line data is input, absolute position data (CP) is stored in the data string, representing the absolute position of the end of the existing line, and line shift data (DL) is stored in the data string. 10
8. Apparatus according to any one of claims 4 to 7, in which pointer data is stored in the memory means (32), pointing to a current location in the data string. 15
9. Apparatus according to claim 8 in which, during a delete operation, if the data at the location in the data string pointed to by the pointer data is control data representing movement of the next character output position without output of a character, it is determined that the next character output position is not at the position of an output character. 20 25
10. Apparatus according to any one of claims 4 to 9, in which a control code representing movement of the next character output position without output of a character defines the end of a word in the data string. 30
11. Apparatus according to any preceding claim, comprising impact type print means (3, 4, 5) for providing said character outputs. 35
12. Apparatus according to claim 11, wherein the impact type print means (3, 4, 5) comprises print wheel means (5). 40
13. Apparatus according to claim 11 or 12, wherein the impact type print means (3, 4, 5) operates using a printing ribbon to print and a deleting ribbon to delete a printed character. 45
14. Apparatus according to any preceding claim, wherein said character inputs and said character outputs comprise English characters. 50
15. Apparatus according to any preceding claim, wherein if the next character output position is at the position of a character which has already been output when the delete function is operated, that character and preceding characters are deleted until the beginning of a word is reached. 55

## Patentansprüche

1. Vorrichtung zum Empfangen von Zeicheneingaben, zum Speichern von eingegebenen Zeichendaten in einer Speichereinrichtung (32), und zum Ausgeben von Zeichen, mit einer nächsten Zeichen-Ausgabeposition dergestalt, daß ein während Normalbetrieb eingegebenes Zeichen an der nächsten Zeichen-Ausgabeposition ausgegeben wird, und mit einer Löschfunktion zum Löschen mindestens eines Ausgabezeichens derart, daß zumindest dieses Zeichen gelöscht wird, falls die Löschfunktion ausgeführt wird, wenn die nächste Zeichen-Ausgabeposition an der Position eines bereits ausgegebenen Zeichens ist, wogegen die nächste Zeichen-Ausgabeposition an die Stelle des nächstliegenden vorangehenden, bereits ausgegebenen Zeichens verschoben wird, falls die Löschfunktion auf eine Zeicheneingabe folgend ausgeführt wird, wenn die nächste Zeichen-Ausgabeposition nicht an der Position eines bereits ausgegebenen Zeichens ist, dadurch gekennzeichnet, daß das nächstliegende vorangehende Zeichen nicht gelöscht wird, falls die Löschfunktion ausgeführt wird, wenn die nächste Zeichen-Ausgabeposition nicht an der Position eines bereits ausgegebenen Zeichens ist.
2. Vorrichtung nach Anspruch 1, dadurch gekennzeichnet, daß den Abstand zwischen der nächsten Zeichen-Ausgabeposition und der Position, an der ein Zeichen ausgegeben wurde, kennzeichnende Leerraummengen-Daten in der Speichereinrichtung (32) gespeichert werden.
3. Vorrichtung nach Anspruch 2, dadurch gekennzeichnet, daß während eines Löschvorgangs die Leerraummengen-Daten gelesen werden und, falls diese nicht Null sind, bestimmt wird, daß die nächste Zeichen-Ausgabeposition nicht an der Position eines ausgegebenen Zeichens ist.
4. Vorrichtung nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß eingegebene Zeichendaten in der Speichereinrichtung (32) als Datenkette mit verstreut eingefügten Steuerdaten gespeichert werden.
5. Vorrichtung nach Anspruch 4 im Falle dessen Abhängigkeit von Anspruch 2 oder 3, dadurch gekennzeichnet, daß die Leerraummengen-Daten aktualisiert werden, wenn Leerraum-Daten eingegeben werden, und Steuerdaten (SP) in der den Wert der Leerraummengen-Daten re-

präsentierenden Datenkette gespeichert werden, wenn Zeichendaten auf die Eingabe von Leerraum-Daten folgend eingegeben werden.

6. Vorrichtung nach Anspruch 4 oder 5, dadurch gekennzeichnet, daß die Steuerdaten Absolutpositions-Daten beinhalten. 5
7. Vorrichtung nach Anspruch 6, dadurch gekennzeichnet, daß bei der Eingabe von Zeilenumbruch-Daten Absolutpositions-Daten (CP) in der Datenkette gespeichert werden, die die Absolutposition des Endes der bestehenden Zeile repräsentieren, und Zeilenvorschub-Daten (DL) in der Datenkette gespeichert werden. 10
8. Vorrichtung nach einem der Ansprüche 4 bis 7, dadurch gekennzeichnet, daß Zeigerdaten in der Speichereinrichtung (32) gespeichert werden, die auf eine momentane Stelle in der Datenkette zeigen. 20
9. Vorrichtung nach Anspruch 8, dadurch gekennzeichnet, daß während einer Löschoperation bestimmt wird, daß die nächste Zeichen-Ausgabeposition nicht an der Position eines ausgegebenen Zeichens ist, falls die Daten an der Stelle in der Datenkette, auf die mittels der Zeigerdaten gezeigt wird, Steuerdaten sind, die eine Verschiebung der nächsten Zeichen-Ausgabeposition ohne Ausgabe eines Zeichens repräsentieren. 25
10. Vorrichtung nach einem der Ansprüche 4 bis 9, dadurch gekennzeichnet, daß ein Steuercode, der die Verschiebung der nächsten Zeichen-Ausgabeposition ohne Ausgabe eines Zeichens repräsentiert, das Ende eines Wortes in der Datenkette definiert. 30
11. Vorrichtung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß eine Druckeinrichtung nach dem Anschlagverfahren (3, 4, 5) zur Zeichenausgabe vorgesehen ist. 40
12. Vorrichtung nach Anspruch 11, dadurch gekennzeichnet, daß die Druckeinrichtung nach dem Anschlagverfahren (3, 4, 5) eine Druckrad-Einrichtung (5) aufweist. 45
13. Vorrichtung nach Anspruch 11 oder 12, dadurch gekennzeichnet, daß die Druckeinrichtung nach dem Anschlagverfahren (3, 4, 5) unter Verwendung eines Druckbandes zum Drucken und eines Löschbandes zum Löschen eines gedruckten Zeichens arbeitet. 50

14. Vorrichtung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß die Zeicheneingaben und die Zeichenausgaben englische Zeichen umfassen.

15. Vorrichtung nach einem der vorstehenden Ansprüche, dadurch gekennzeichnet, daß, falls bei Ausführen der Löschfunktion die nächste Zeichen-Ausgabeposition an der Stelle eines Zeichens, das bereits ausgegeben wurde, ist, dieses Zeichen und vorangehende Zeichen bis zum Erreichen des Beginns eines Wortes gelöscht werden.

## Revendications

1. Appareil pour recevoir des entrées de caractères, stocker des données de caractères d'entrée dans un moyen à mémoire (32), et fournir des sorties de caractères, l'appareil ayant une position de sortie du caractère suivant telle qu'en fonctionnement normal, si un caractère est introduit, il sera sorti à la position de sortie du caractère suivant, et l'appareil ayant une fonction d'effacement pour effacer au moins un caractère de sortie, de telle sorte que si la fonction d'effacement est utilisée lorsque la position de sortie du caractère suivant se trouve à la position d'un caractère ayant déjà été sorti, ce caractère est au moins effacé, alors que si, après l'opération d'entrée d'un caractère, la fonction d'effacement est utilisée lorsque la position de sortie du caractère suivant ne se trouve pas à la position d'un caractère ayant déjà été sorti, la position de sortie du caractère suivant est décalée vers la position du caractère précédent le plus proche ayant déjà été sorti, caractérisé en ce que le caractère précédent le plus proche n'est pas effacé lorsque la fonction d'effacement est utilisée alors que la position de sortie du caractère suivant ne se trouve pas à la position d'un caractère ayant déjà été sorti.
2. Appareil selon la revendication 1, dans lequel une donnée de quantité d'espaces indiquant la distance entre la position de sortie du caractère suivant et la position à laquelle un caractère a été sorti, est stockée dans le moyen à mémoire (32).
3. Appareil selon la revendication 2, dans lequel, lors d'une opération d'effacement, ladite donnée de quantité d'espaces est lue et, si elle n'est pas nulle, il est déterminé que la position de sortie du caractère suivant ne se trouve pas à la position d'un caractère sorti.

4. Appareil selon l'une quelconque des revendications 1 à 3, dans lequel la donnée de caractère d'entrée est stockée dans le moyen à mémoire (32) sous la forme d'une chaîne de données dans laquelle sont dispersées des données de commande. 5
5. Appareil selon la revendication 4, lorsqu'elle est considérée en association avec la revendication 2 ou 3, dans lequel, lorsqu'une donnée d'espace est introduite, la donnée de quantité d'espaces est mise à jour, et lorsqu'une donnée de caractère est introduite après une donnée d'espace, une donnée de commande (SP) est stockée dans la chaîne de données pour représenter la valeur de la donnée de quantité d'espaces. 10 15
6. Appareil selon la revendication 4 ou 5, dans lequel ladite donnée de commande comprend une donnée de position absolue. 20
7. Appareil selon la revendication 6, dans lequel, lorsqu'une donnée de passage à la ligne est introduite, une donnée de position absolue (CP) est stockée dans la chaîne de données, pour représenter la position absolue de la fin de la ligne en cours, et une donnée de décalage de ligne (DL) est stockée dans la chaîne de données. 25 30
8. Appareil selon l'une quelconque des revendications 4 à 7, dans lequel une donnée de pointeur est stockée dans le moyen à mémoire (32) pour pointer sur une position courante dans la chaîne de données. 35
9. Appareil selon la revendication 8, dans lequel, lors d'une opération d'effacement, si la donnée se trouvant dans la chaîne de données à l'emplacement pointé par la donnée de pointeur est une donnée de commande représentant un mouvement de la position de sortie du caractère suivant sans sortie de caractère, il est déterminé que la position de sortie du caractère suivant ne se trouve pas à la position d'un caractère sorti. 40 45
10. Appareil selon l'une quelconque des revendications 4 à 9, dans lequel un code de commande représentant un mouvement de la position de sortie du caractère suivant sans sortie de caractère, définit la fin d'un mot dans la chaîne de données. 50 55
11. Appareil selon l'une quelconque des revendications précédentes, comprenant un moyen d'impression du type à impact (3, 4, 5) pour produire lesdites sorties de caractères.
12. Appareil selon la revendication 11, dans lequel le moyen d'impression du type à impact (3, 4, 5) comprend un moyen à marguerite (5).
13. Appareil selon la revendication 11 ou 12, dans lequel le moyen d'impression du type à impact (3, 4, 5) fonctionne par utilisation d'un ruban d'impression pour imprimer et d'un ruban d'effacement pour effacer un caractère imprimé.
14. Appareil selon l'une quelconque des revendications précédentes, dans lequel lesdites entrées de caractères et lesdites sorties de caractères comprennent des caractères Anglais.
15. Appareil selon l'une quelconque des revendications précédentes, dans lequel, si la position de sortie du caractère suivant se trouve à la position d'un caractère ayant déjà été sorti lorsque la fonction d'effacement est utilisée, ce caractère et les caractères précédents sont effacés jusqu'à ce que le début d'un mot soit atteint.

FIG. 1

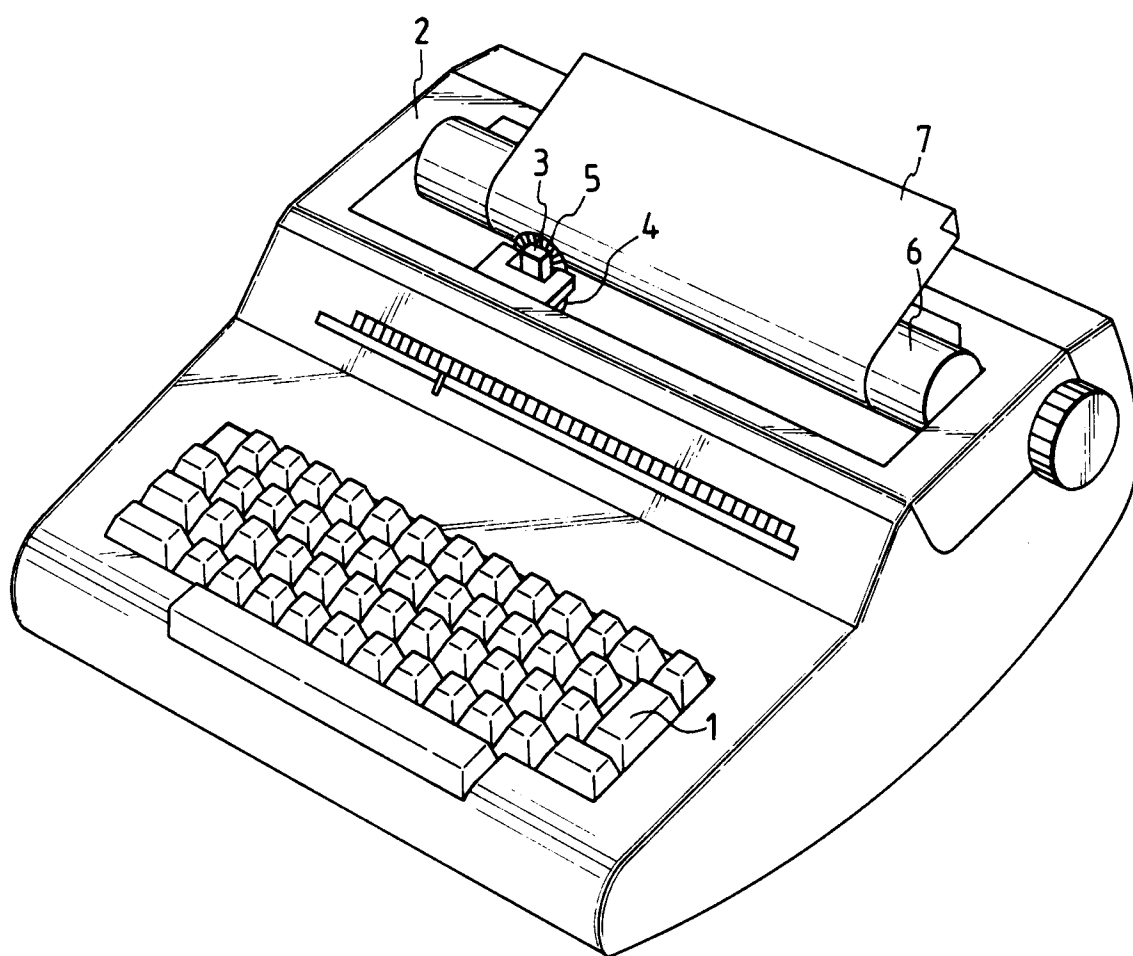


FIG. 2

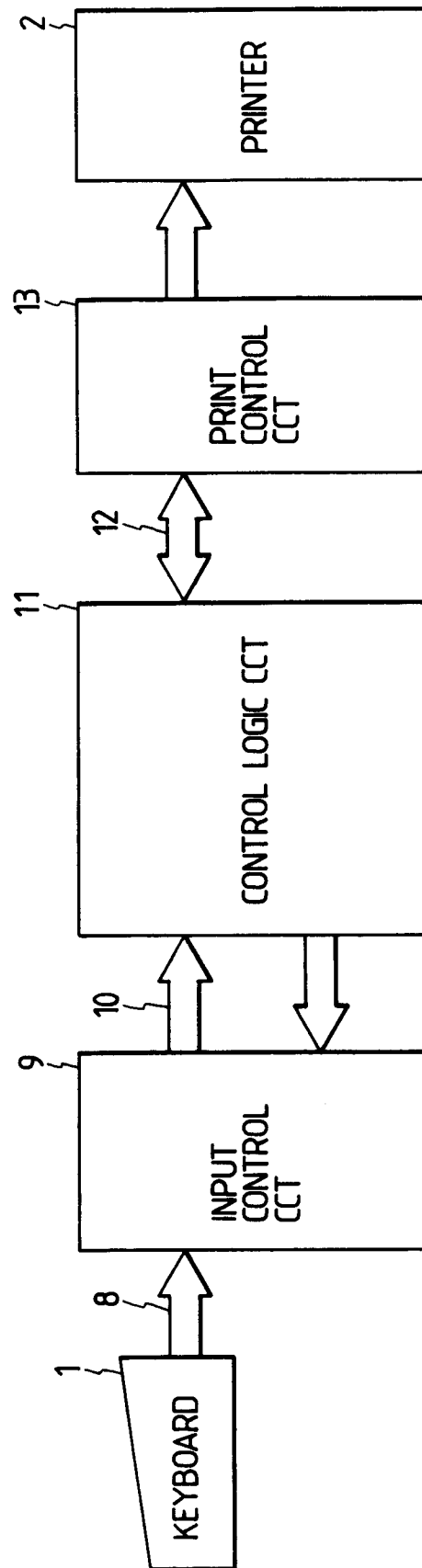
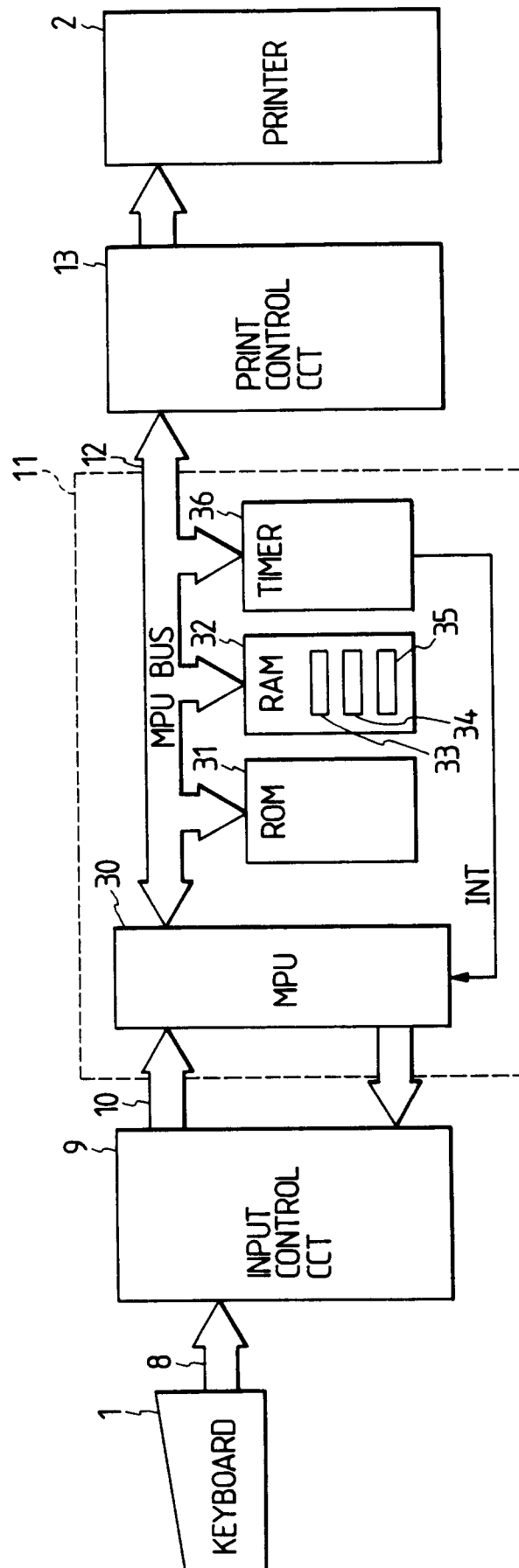


FIG. 3



**FIG. 4**

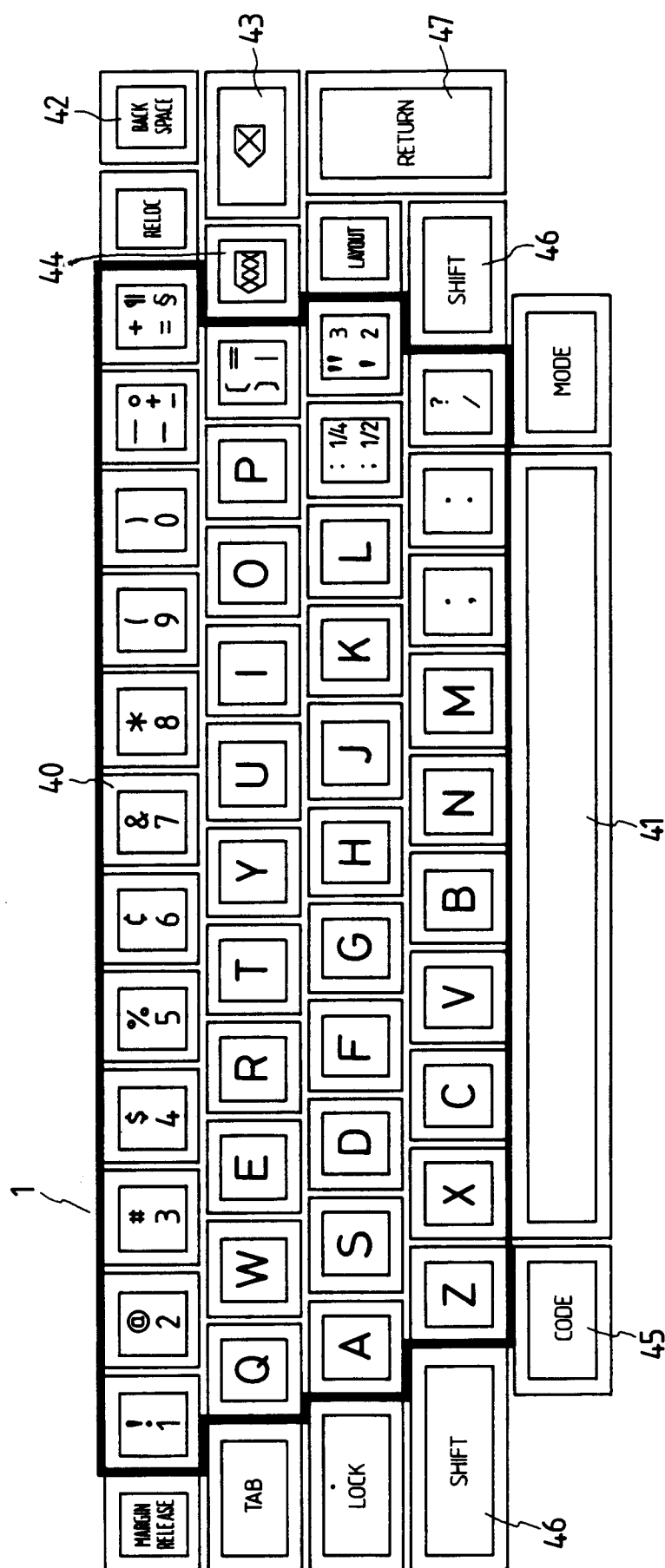








FIG. 6A

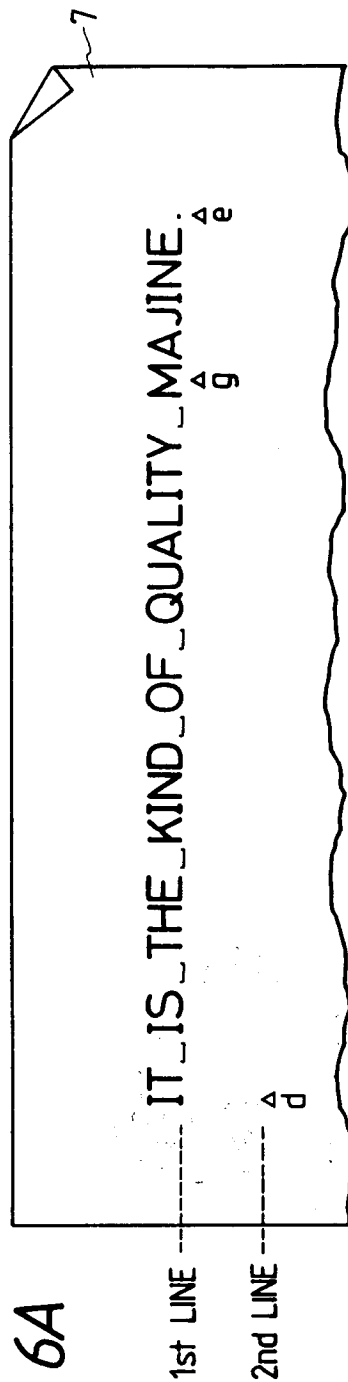


FIG. 6B

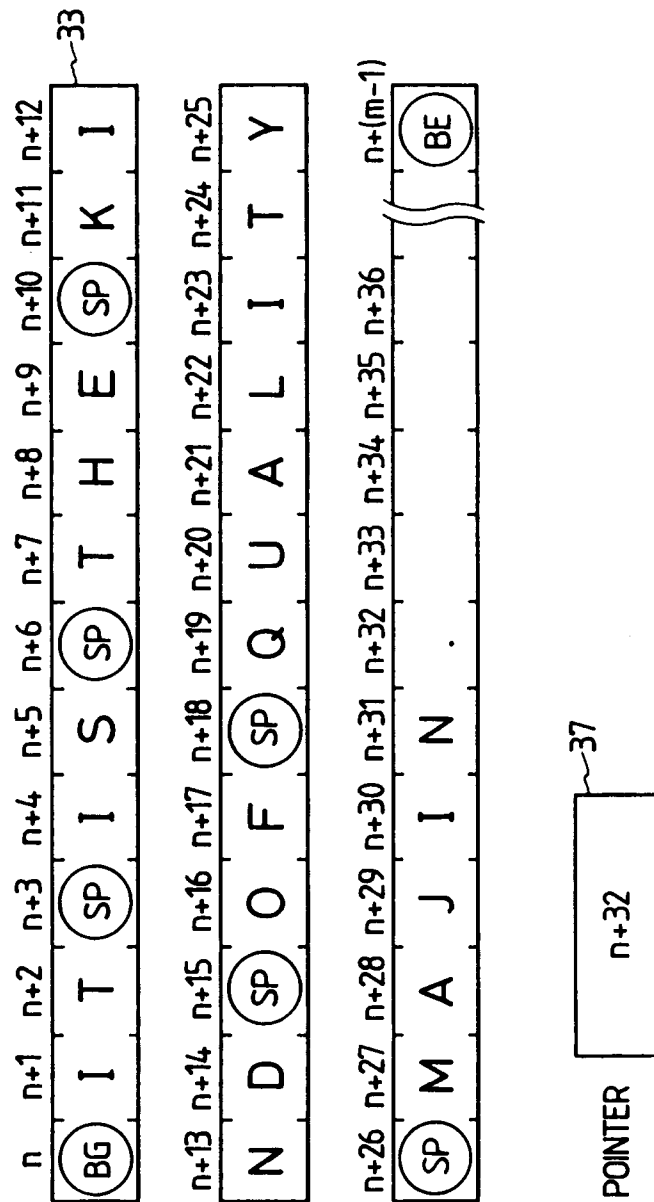


FIG. 7A

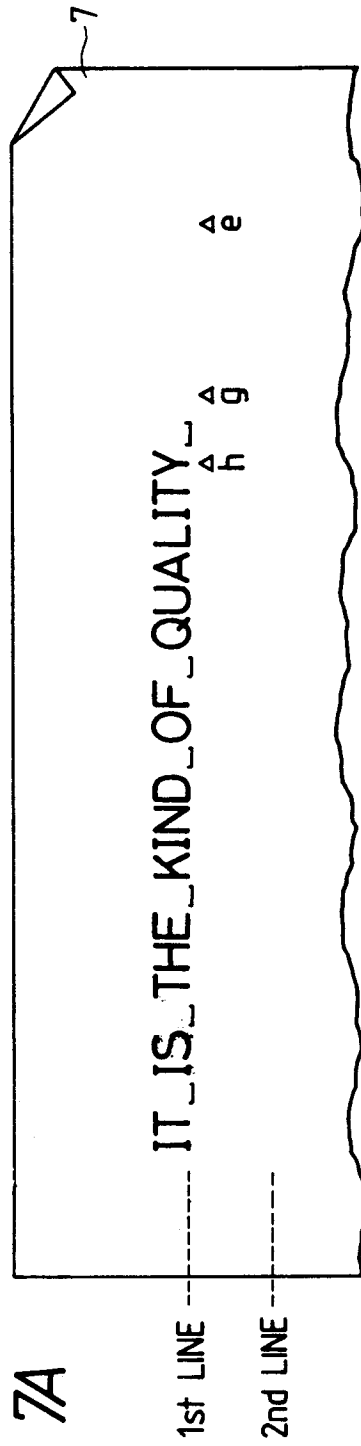


FIG. 7B

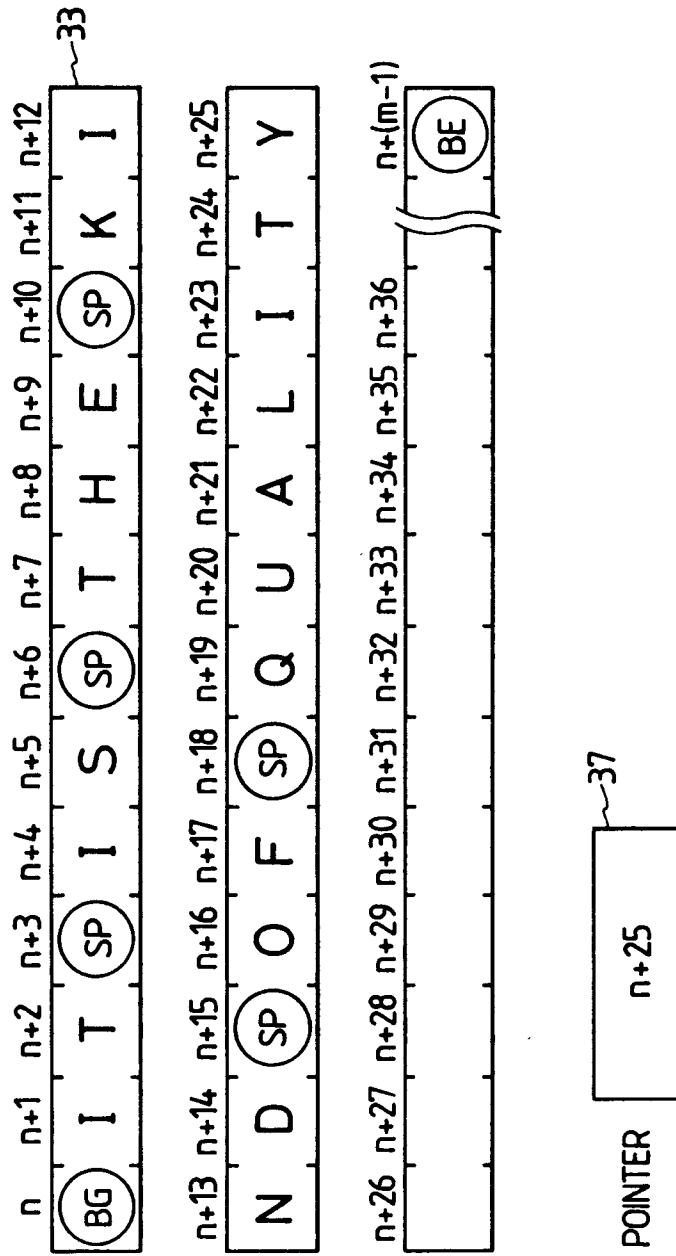


FIG. 8

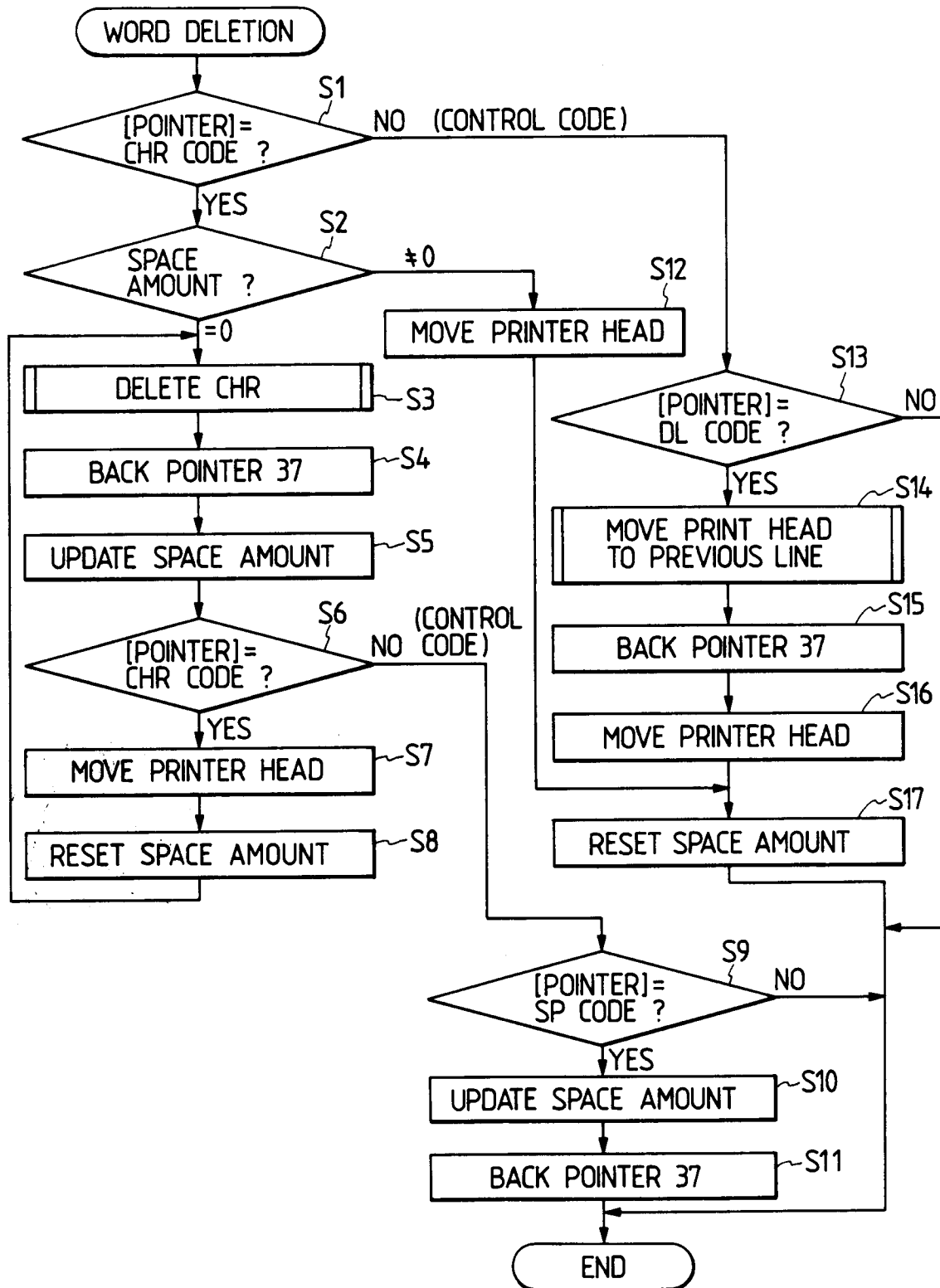


FIG. 9A

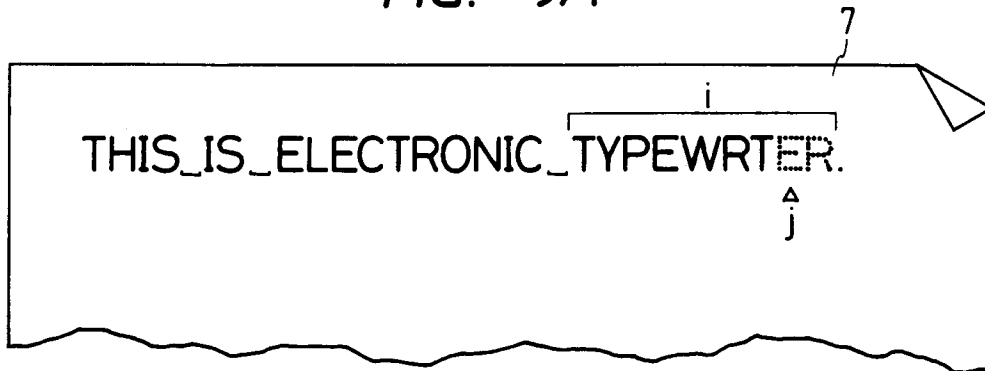


FIG. 9B

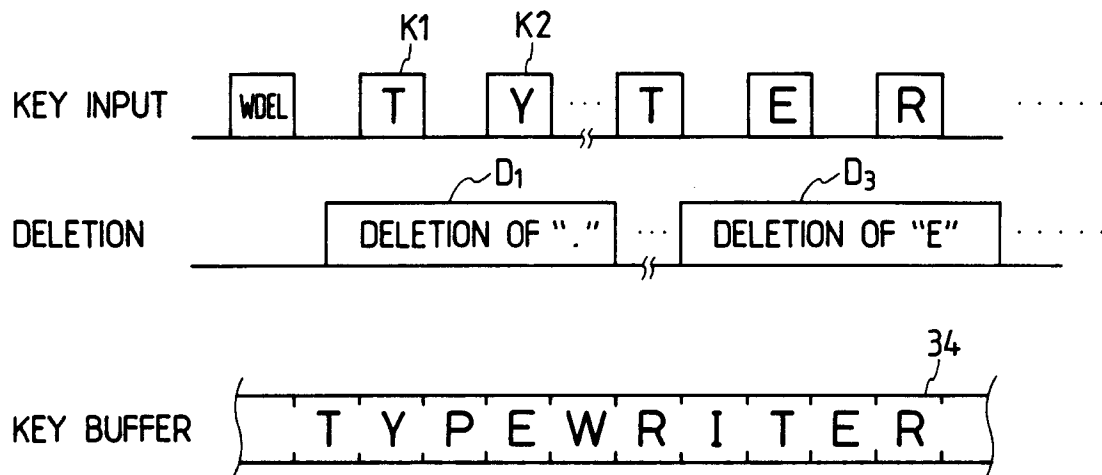


FIG. 10

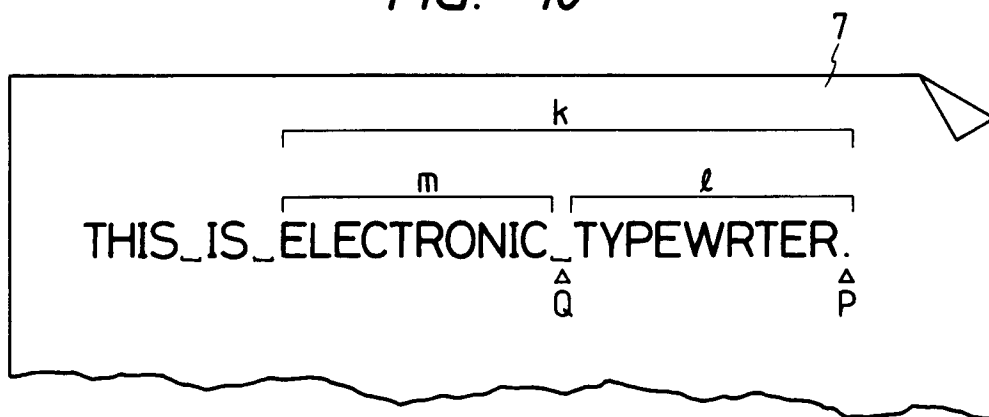


FIG. 11

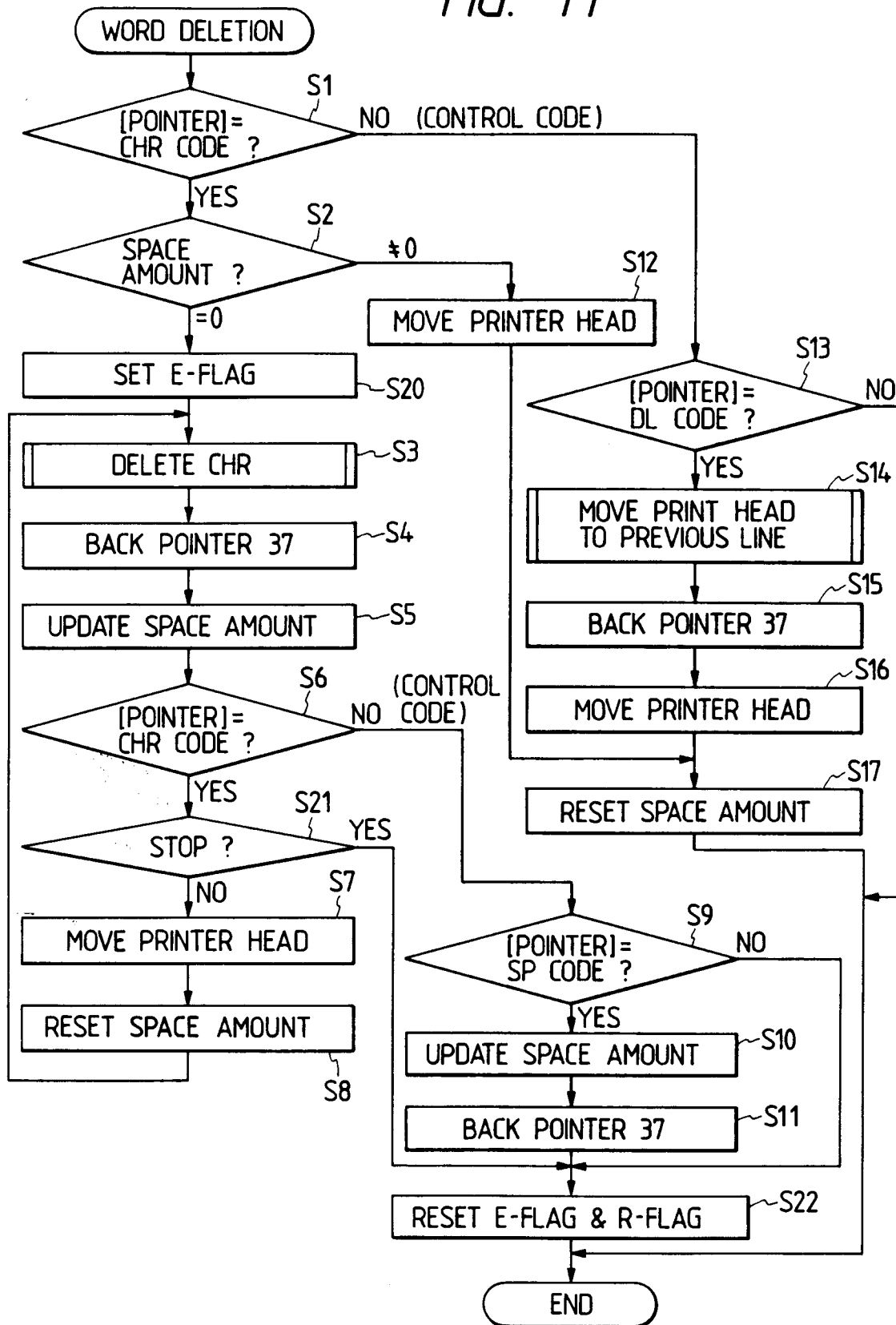
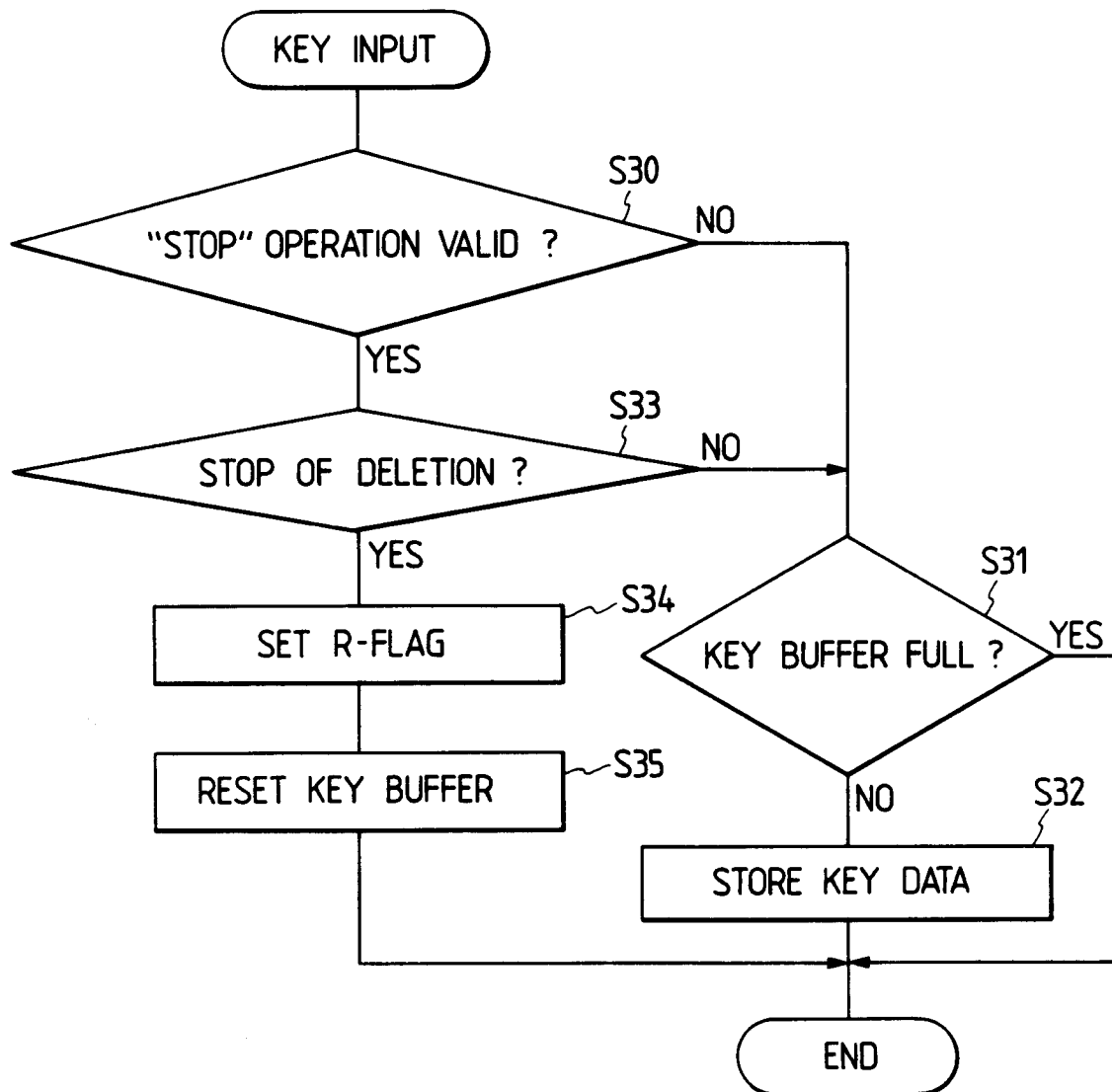


FIG. 12





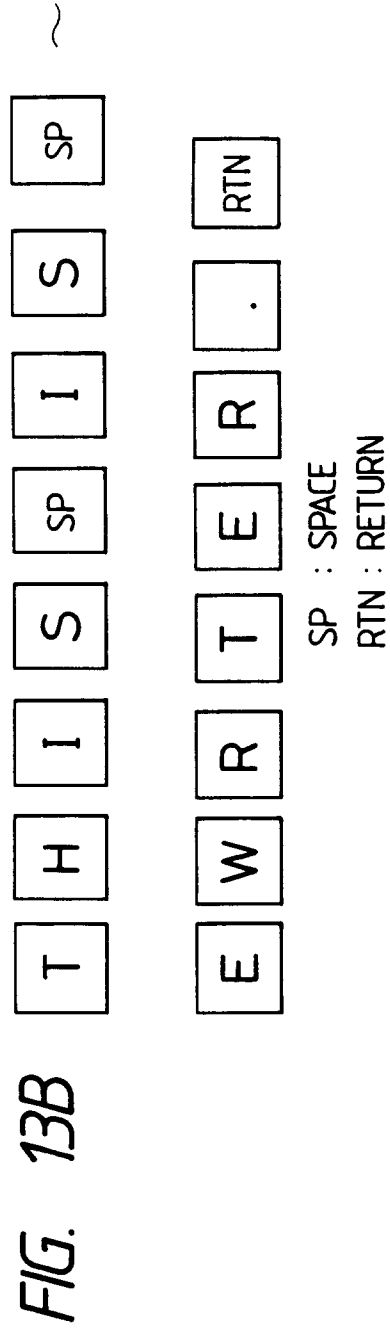
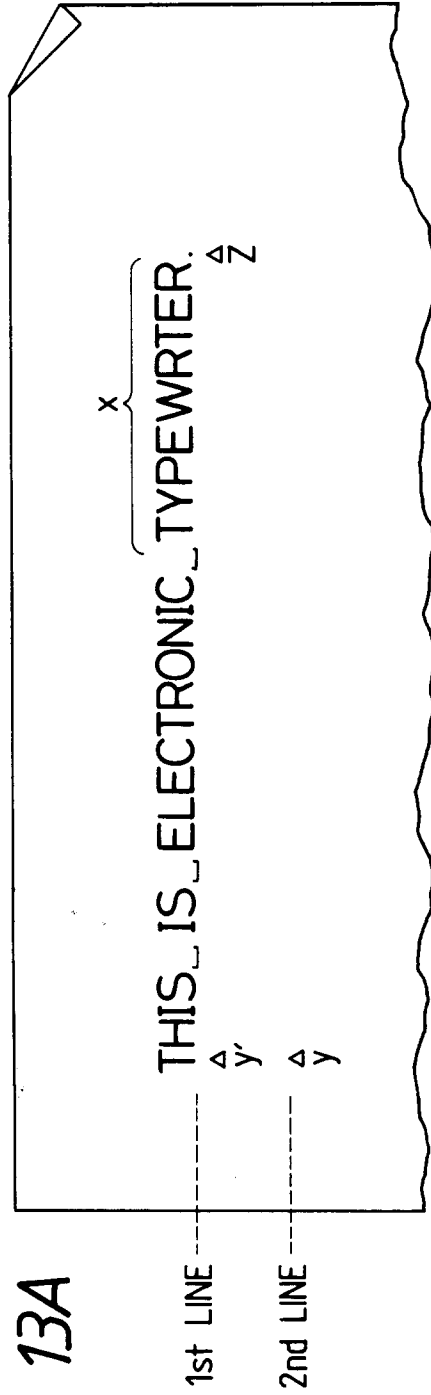


FIG. 14A

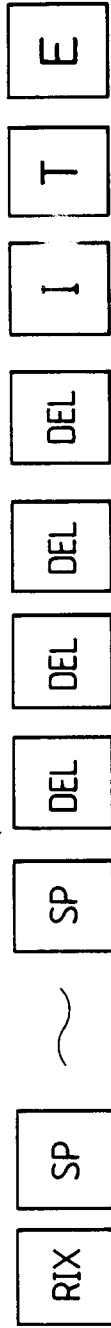


FIG. 14B

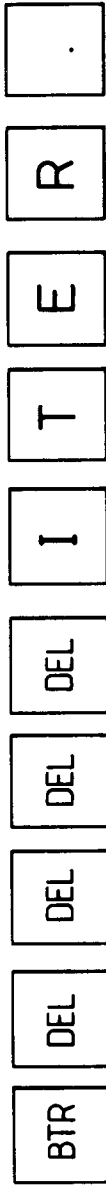
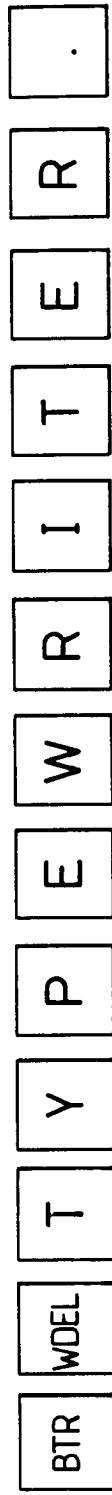


FIG. 14C



RIX : REVERSE INDEX  
 SP : SPACE  
 BTR : BACK TRACE  
 DEL : DELETE  
 WDEL : WORD DELETE