(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 206 330**
A2

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **86108655.1**

(22) Date of filing: **25.06.86**

(51) Int. Cl.⁴: **G 09 G 1/00**

(30) Priority: **26.06.85 JP 137728/85**

(43) Date of publication of application: **30.12.86 Bulletin 86/52**

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

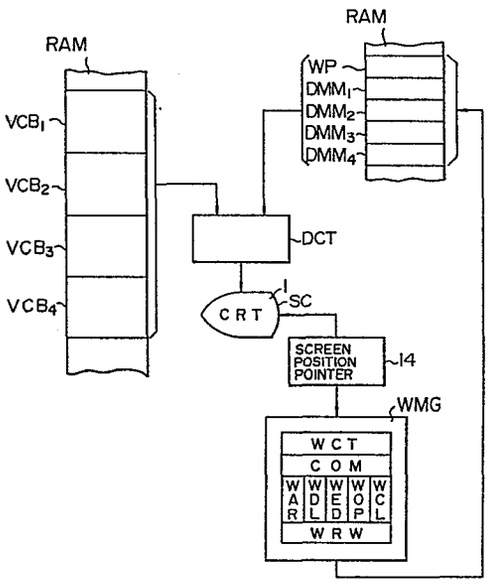
(71) Applicant: **HITACHI, LTD., 6, Kanda Surugadai 4-chome Chiyoda-ku, Tokyo 100 (JP)**

(72) Inventor: **Sakata, Yoshiaki Hitachi Shumeiryo, 248-1, Mimomichohongo, Narashino-shi (JP)**

(74) Representative: **Patentanwälte Beetz sen. - Beetz jun. Timpe - Siegfried - Schmitt-Fumian, Steinsdorfstrasse 10, D-8000 München 22 (DE)**

(54) Apparatus for controlling screen display.

(57) Screen division display wherein a screen is divided into plural portions and an independent image is displayed on each divided portion, is achieved by the provision of a plurality of image memory areas (VCB_1–VCB_4) each corresponding to the screen; a display management memory area (DMM) for storing the positions on the screen of the storage contents of the image memory areas; and a display controller (DCT) for displaying the storage content of the designated image memory area on a predetermined divided portion of the screen, based on the storage content of the display management memory area. Further, there are provided a designating unit (14) for designating a desired one of the plural divided portions and a display management unit (WMG) for renewing the storage content of the display management area in such a way that the position on the screen of the image memory area corresponding to the divided portion designated by the designating unit is located on the entirety of the screen.



EP 0 206 330 A2

0 206 330

APPARATUS FOR CONTROLLING SCREEN DISPLAY

BACKGROUND OF THE INVENTION

The present invention relates to an apparatus for controlling an image to be displayed on the screen of an information processor such as a computer, and more particularly to an apparatus for controlling a display image wherein the screen is divided into a plurality of portions on which images are independently displayed.

Plural application programs are executed in parallel at a time in a computer and the processing results are sequentially monitored by a display device such as a CRT. It is preferable to monitor the results of plural application programs on a single display device. Such a requirement has been met by means of a screen division method wherein the screen is divided into a plurality of portions and the processing result of each program is displayed on its specifically allocated portion. In such a method, the dimension and position of each divided portion, or the number of portions may be set as desired. With this method, execution processes of a plurality of programs can be monitored concurrently and allows an efficient operability of the computer. During or after execution of application programs, a desired one of the images displayed on the plurality of divided portions is

sometimes desired to be displayed on the entire screen at a magnified scale. To this end, conventionally an image erasure function for erasing unnecessary images has been provided. However, on the contrary, a function for retaining a desired image has not been considered so that the erasure function can only perform an erasure of a single image at a time. Thus, to retain a desired image among a plurality of images and display it on the entire screen at a magnified scale, it has been necessary to use the erasure function as many times as the number of unnecessary images, thereby resulting in complicated operations. The above conventional screen division method is disclosed for example in Japanese Patent Unexamined Publication (JP-A) No. 51-114829.

SUMMARY OF THE INVENTION

In view of the above problems, the present invention seeks to provide an apparatus for controlling a screen display and displaying a desired image, among a plurality of images displayed on plurally divided portions of a display screen, with a simple operation at a magnified scale.

Image display on divided portions of the screen can be achieved by the provision of: a plurality of image memory areas each corresponding to an image to be displayed on each divided portion of the screen; a display management memory area for storing the positions

of the images on the screen; and a display controller for displaying the content of the image memory area at a designated portion of the screen, based on the content of the display management memory area. In more particular, each image memory area is assigned to a particular program to be executed by a computer. Each program sequentially writes display information in the assigned image memory area. The display management memory area stores information as to which program is to be allocated to a particular image memory area, and which divided portion of the screen is to be used for displaying the content of the image memory area. Based on the stored information in the display management memory area, the display controller controls to display the content of a desired image memory area on a desired position of the screen.

A screen position pointer is provided for designating a desired image among a plurality of images respectively corresponding to the image memory areas and respectively displayed on divided portions of the screen. The screen position pointer may be a coordinate input device of a various type or a keyboard. A window management unit is further provided for renewing the content of the display management memory area to display the image designated by the screen position pointer on the entire screen at a magnified scale.

Upon designation by the screen position pointer of an image desired to be retained among a

plurality of images, the window management unit can renew the content of the display management memory area. As a result, the display controller, which controls the display status based on the content of the display management memory area, can display the content of the image memory area corresponding to the desired image on the entire screen at a magnified scale. In summary, a simple designation operation by the screen position pointer enables a display of a desired image, among a plurality of images displayed on divided portions of the screen, on the entire screen at a magnified scale.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing the principle of the present invention;

Fig. 2 is a block diagram of a computer to which the present invention is applied;

Fig. 3 shows an example of a display image on a screen;

Figs. 4 and 5 show a correspondence between the display images and the image memory areas;

Figs. 6 and 7 show an example of the screen position pointer and its operating principle;

Fig. 8 shows one example how a display image is divided;

Fig. 9 is a flow chart of an open operation for opening an image to the screen;

Fig. 10 is a flow chart of a close operation for closing an image displayed on the screen;

Fig. 11 is an example of the memory map of the display management memory area;

Fig. 12 is a block diagram showing the outline of the window management program;

Fig. 13 shows one example how a display image is reorganized;

Fig. 14 is a flow chart showing one example of image reorganizing operation;

Fig. 15 is a flow chart showing the relationship between the window management program and the screen position pointer;

Fig. 16 is a flow chart of the window control program;

Fig. 17 is a flow chart of the command recognizing routine;

Fig. 18 is a flow chart of the judgement procedure for each command icon;

Fig. 19 shows the icon representation "RETAIN" on the screen;

Fig. 20 is an example of the memory map of the icon position information table,

Fig. 21 is a flow chart showing one example of the window open routine;

Fig. 22 is a flow chart showing one example of the window close routine;

Fig. 23 is a flow chart showing one example of

the window reorganizing routine;

Fig. 24 shows examples of window patterns on the screen;

Figs. 25 and 26 show transitions of window patterns; and

Fig. 27 is the memory map of memories RWNO and TNO.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

An embodiment of the present invention will now be described with reference to the accompanying drawings. Fig. 2 is a block diagram showing the construction of a computer to which the present invention is applied, the computer being of the type called generally a personal computer or a business computer. Referring particularly to Fig. 2, a display device 1 is a cathode ray tube (merely called CRT hereinafter). The computer further includes a random access memory (merely called RAM hereinafter), a microprocessor 3 constituting a processing unit, a keyboard 4 constituting an input unit, a read-only memory 5 (abbreviated ROM hereinafter), a clock oscillator 6, a timing controller 7, a character font ROM 8, a CRT controller 9, a refreshing memory 10, a parallel/serial converter 11, a disk store 12 constituting an outer storage device, a disk controller connecting the disk store 12 to the microprocessor 8 via the bus. A screen position pointer 14 is for indicating coordinates on the CRT 1

and delivers designated coordinate information to the bus. The above circuit elements are coupled to the microprocessor 8 via the bus as shown in Fig. 2.

Fig. 3 shows a display example of a plurality of divided portions (hereinafter called a window) of the screen SC of the CRT 1 and an area ICN (hereinafter called an icon area) within which a symbolized representation of an image, whose content has already been stored, is displayed. Different application programs are respectively assigned to each window WD_1, WD_2 and WD_3 of the screen SC.

Fig. 4 shows the relationship between the real screen indicated by a solid line and screens (hereinafter referred to as virtual screen) VS_1, VS_2 and VS_3 indicated by a broken line which programs assigned to the respective windows WD_1, WD_2 and WD_3 can be displayed. The program corresponding to the window WD_1 performs its display assuming the virtual screen VS_1 as its real screen, while the program corresponding to the window WD_2 performs its display assuming the virtual screen VS_2 as its real screen. Similarly, the Program corresponding to the window 3 performs its display assuming on the virtual screen VS_3 as its real screen. Thus, each window is a real image portion of the virtual screen VS_1, VS_2 or VS_3, while each program runs assuming its display area as the whole area of the virtual screen VS_1, VS_2 or VS_3. The entire content cannot therefore be displayed unless the real image area

screen equals the size of the screen. The virtual screens VS_1, VS_2 and VS_3 are realized by provision of image memory areas corresponding thereto in the RAM 2. Fig. 5 shows the memory map of image memory areas VCB_1, VCB_2, VCB_3 and VCB_4 of the RAM 2. Each of different application programs, which are stored in the RAM 2, ROM 5 or external disk store 12 and executed by the microprocessor 8, is assigned to a desired one of the image memory areas VCB_1, VCB_2, VCB_3 and VCB_4.

In this example, the screen position pointer 14 is used to pick up the position information on the screen SC of the CRT 1 and controls the position and dimension of each window WD_1, WD_2 or WD_3. Fig. 6 shows an example of the screen position pointer 14 which displays a graphic pointer Pi on the screen SC, for example, at the character train as shown in Fig. 6. The position where the pointer Pi is displayed can be set by moving the screen position pointer 14 which in this embodiment is a so-called mouse 140. The position information from the mouse 140 is outputted to the bus as shown in Fig. 2 so that the microprocessor 8 controls the position of the graphic cursor Pi based on the information received via the bus. The mouse 140 is provided with a left button switch S_L and a right button switch S_R, the switches being used for control over the pointed character train, window WD_1, WD_2 or WD_3. Fig. 7 shows the proportional relation between the movement of the mouse 140 and the movement of

the graphic cursor Pi. Upon movement of the mouse 140 in the direction XP, XN, YP or YN indicated by an arrow, the graphic cursor Pi on the screen SC moves in the direction XP', XN', YP' or YN', respectively. Such operation is achieved by programs stored in the RAM 2, ROM 5 or the like.

Fig. 8 shows one example of a window control using the screen position pointer 14 and the graphic cursor Pi, wherein Fig. 8(a) shows the screen SC with window WD_2 is closed while Fig. 8(b) shows the screen SC with windows WD_1 and WD_2 opened. Reference 15 represents the names of the opened windows WD_1 and WD_2. To close the window, the name 15 is pointed by the graphic cursor Pi, and thereafter while pushing the left button S_L of the mouse 140, the graphic cursor is moved to the icon area ICN. An icon 16 is used for changing the dimension of an opened window on the screen SC. For instance, to make wider the width of the window WD_1 of Fig. 8(b), the graphic cursor Pi is first pointed at the icon 16 displayed on the upper side of the window WD_1, and thereafter the mouse 140 is moved to the right while pushing the left button S_L. An icon 17 is used for deletion of image information on a window to thereby inhibit re-opening the deleted window. An icon 19 displayed on the icon area ICN is used for indication of a window which can be opened on the screen SC in such a way that the icon 19 is moved to the area where the image information is to be

displayed.

The operation procedure with respect to Fig. 8 will be described by referring to the flow charts of Figs. 9 and 10. The flow chart of Fig. 9 shows the processes for opening a window indicated by the icon 19 and displaying two windows WD_1 and WD_2 on the window area, i.e., the processes from Fig. 8(a) to Fig. 8(b). First at step 9a, by moving the mouse 140 the icon 19 displayed on the icon area ICN is pointed by the graphic cursor Pi and the left button switch S_L is pushed for designation of the icon 19. Next, at step 9b the designated icon 19 is moved to the right area on the screen SC by moving the mouse 140 while pushing the left button switch S_L. At succeeding step 9c the left button switch S_L of the mouse 140 is released. As a result, the window WD_2 indicated by the icon 19 is opened on the right area of the screen SC as shown in Fig. 8(b).

On the contrary, the flow chart of Fig. 10 shows the processes for closing the window WD_2, i.e., the processes from Fig. 8(b) to Fig. 8(a). First, at step 10a by moving the mouse 140 the window name "HIJ" of the window WD_2 is pointed by the graphic cursor Pi and the left button switch S_L is pushed for designation of the window name "HIJ". Next, at step 10b the designated icon 19 is moved to the icon area on the screen SC by moving the mouse 140 while pushing the left button switch S_L. At succeeding step 10c the left

button switch S_L of the mouse 140 is released. As a result, the window WD_2 is erased and the screen SC becomes as shown in Fig. 8(a).

Fig. 1 shows the entire construction of the screen control system. There are provided the image memory areas VCB_1, VCB_2, VCB_3 and VCB_4 in the RAM 2 as described previously. Although four image memory areas are used in this embodiment, it is only for convenience of explanation and the number of memory areas is not limited thereto. A display management memory area DMM is provided in the RAM 2 at a predetermined area. Display management memory areas DMM_1, DMM_2, DMM_3 and DMM_4 correspond to the image memory areas VCB_1, VCB_2, VCB_3 and VCB_4, respectively. The display management memory area DMM is constructed of several memory areas each storing information on a window to be displayed, the range of content to be displayed of the image memory area VCB_1, VCB_2, VCB_3 or VCB_4, and the like. Fig. 11 shows the concrete memory map of the display management memory area including WP, DMM_1, DMM_2, DMM_3 and DMM_4. The display management memory areas DMM_1, DMM_2, DMM_3 and DMM_4 each have position memories for storing coordinates of the corresponding window displayed on the screen SC based on the content of the corresponding image memory area VCB_1, VCB_2, VCB_3 or VCB_4. That is, the position memories include ML_1, ML_2, ML_3 and ML_4 each for storing the upper left coordinates of the corresponding window on the screen SC, and MR_1,

MR_2, MR_3 and MR_4 each for storing the lower right coordinates of the corresponding window. The window pattern area WP stores the display pattern information of the window, which will later be described.

A display controller DCT controls display of the respective contents of the image memory areas VCB_1, VCB_2, VCB_3 and VCB_4 on the screen SC at appropriate areas, based on the stored information in the display management memory areas DMM_1, DMM_2, DMM_3, and DMM_4. A window management unit WMG performs a renewal management of the windows on the screen SC, based on output information from the screen position pointer 14. Fig. 12 shows the window management unit WMG in more detail in the form of practical programs. The programs of the window management unit WMG are stored in the RAM 2 ROM 5 or the like to achieve predetermined functions through execution by the microprocessor 8. Referring to Fig. 12, a window management program WMGP of the WMG is mainly divided into two programs, i.e., a window control program WCTP of a window controller WCT and a program execution control program PECP of a program execution controller PEC. The window control program WCTP controls the position, dimension, and number of windows and is constructed of the following routines: a window close routine WCL, a window open routine WOP, a window dimension/position changing routine WED, a window deleting routine WDL, a window reorganizing routine WAR which is a main feature of the invention,

a command recognizing routine COM, and a window renewing routine WRW. The window close routine WCL is for closing desired one of the windows opened on the screen SC. On the contrary, the window open routine WOP is for opening a closed window to the screen SC. The window dimension/position changing routine WED is for changing the dimension or position of a window opened on the screen SC. The window deleting routine WDL is for deleting an application program of a window. The program reorganizing routine WAR constituting the main part of the invention will be later described. The operation results of the routines WCL, WOP, WED, WDL and WAR are given to the window renewing routine WRW which in turn renews the contents of the display management memory areas DMM_1, DMM_2, DMM_3 and DMM_4 of the designated window.

Fig. 13 show the screen displays for explaining the window reorganizing routine WAR, wherein the window WD_4 is remained unclosed while the other windows WD_1, WD_2 and WD_3 are closed. Fig. 13(a) shows a screen display before reorganization by the window reorganizing routine WAR, while Fig. 13(b) shows a screen display after reorganization. To reorganize, first an icon "RETAIN" on the screen SC is pointed by the cursor Si by moving the mouse 140 to thereby start operating the window reorganizing routine WAR whose processes are shown in the flow chart of Fig. 14. First, at step 14a by moving the mouse 140 the graphic cursor Pi is made

to point the icon "RETAIN" of the window WD_4 named "STU". Next, the left button switch S_L of the mouse 140 is pushed for designation of the icon "RETAIN" of the window WED_4. Then, at step 14c the screen display is reorganized to obtain the display as shown in Fig. 13(c).

Each routine WCL, WOP, WED, WDL or WAR starts operating upon designation of an icon displayed on the screen SC. The command recognizing routine COM is used for discriminating information from the mouse 140 and starting an appropriate routine. The window control program WCTP having the above functions controls over the windows to attain a suitable status of each window.

The program execution control program PECP starts operating the application program APP corresponding to the window indicated by the graphic cursor Pi. If another window is indicated, the operation of the application program APP of last indicated window is stopped and the program for the other window starts operating. The program execution control program PECP controls to execute the application program APP in the manner described above in an ordinary state.

Fig. 15 is a flow chart showing the relationship between the program execution control program PECP and the window control program WCTP under management by the window management program WMGP, wherein mouse interruption is incorporated. The position information of the mouse 140 is picked up every time when an

unrepresented mouse interruption routine starts, for example, at every 20 msec. The movement of the graphic cursor Pi is controlled in the manner as shown in Fig. 7 in accordance with the position information. In an ordinary state, the application program APP operates under the program execution control program PECP. When the mouse interruption routine starts at every 20 msec, the position information of the mouse 140 is sent to the window execution control program WCTP, which in turn judges, when the mouse 140 is moved, whether the information corresponds to any command icon or not. In case of a command icon, the corresponding routine WCL, WOP, WEP, WDL or WAR is executed. If not, the application program corresponding to the window indicated by the graphic cursor Pi is prepared for execution. The process returns to the program execution control program PECP after the window control program WCTP to thereby execute the current application program APP. The window control program WCTP and the program execution control program PECP are alternately executed upon mouth interruption.

Fig. 16 is a flow chart illustrating the window control program WCTP which is operated when a mouth interruption is effected under management by the window management program WMGP as discussed with Fig. 15. When the window control program WCPT operates, first at step 16a the window program information now executed is saved in a predetermined area in the RAM 2. Thereafter, at step 16b the command recognizing routine

COM is called for execution, which judges if the graphic cursor Pi points at a command icon, and in case of a command icon, prepares for execution of the application program APP. The preparation for execution means that the window program information saved in the RAM 2 at step 16a is restored and wait for a further execution.

Fig. 17 is a flow chart showing the detail of the command recognizing routine COM, which operates as in the following. The coordinates of the graphic cursor Pi on the screen SC are checked if they correspond to any command icon. In case of a command icon, the corresponding command execution routine WCL, WOP, WED, WDL or WAR starts operating. First, at step 17a it is judged if the graphic cursor Pi indicates the dimension/position changing icon 16. This judgement will be explained later in more detail. In case of the diemnsion/position changing icon, then step 17b follows to start operating the window dimension/position changing routine WED. If not at step 17a, then step 17c follows to judge if the graphic cursor Pi indicates the deletion icon. In case of the deletion icon, step 17d follows to start operating the window deleting routine WDL. If not, step 17e follows. Similarly, at steps 17e, 17g, 17i, it is judged if the graphic cursor Pi indicates the reorganizing icon 17, window open icon 19, or window close icon 15. If affirmative, at step 17f, 17h, or 17i the window reorganizing routine WAR,

window open routine WOP, or window close routine WCL is executed. After the end of execution of any one of the routines WED, WDL, WAR, WOP and WCL, or the graphic cursor Pi indicates no command icon, then the process returns to step 16c of Fig. 16 from the routine COM.

Fig. 18 is a flow chart showing one example how the command icon is discriminated at steps 17a, 17c, 17e, 17g and 17i. Information on position or area where each command icon is to be displayed is previously stored in the RAM 2 to realize the discrimination of command icons. For instance, in Fig. 19 showing the reorganizing icon "RETAIN" on the screen SC, the display area of the icon can be defined by the coordinates of the upper left corner I_{CL} and the lower right corner I_{CR}. As shown in Fig. 20, a combination of X coordinate I_{CLx} and Y coordinate I_{CLy} of the upper left corner I_{CL} and X coordinate I_{CRx} and Y coordinate I_{CRy} of the lower right corner I_{CR}, is stored beforehand in the RAM 2 at an appropriate memory area. Such coordinate information is stored for each command icon. Upon operation of the command icon judgement routine, first at step 18a, the coordinate information of the graphic cursor Pi on the screen SC is read from a cursor memory provided at the CRT controller 9 of Fig. 2 or in the RAM 2. At succeeding step 18b position information I_{CL} and I_{CR} of the icon to be judged are read from the predetermined memory area of the RAM 2. At steps 18c and 18d the coordinates of the graphic cursor Pi are judged if they

are within the area defined by the upper left corner I_{CL} and the lower right corner I_{CR}. If both conditions at steps 18c and 18d are met, then at step 18e it is decided that the graphic cursor Pi indicates the icon now concerned. On the contrary, if one of the conditions is not met, then at step 18f it is decided that the graphic cursor Pi does not indicate the icon now concerned.

The window open routine WOP, window close routine WCL and window reorganizing routine WAR will be described in connection with Figs. 21 to 23.

In the present embodiment, the number of windows are four and the patterns of these windows can take eight different configurations which change from one after another in accordance with the status transition diagrams of Figs. 25 and 26. It is noted that Fig. 24 only concerns about the pattern itself and does not show the relative dimension of the windows WD_1, WD_2, WD_3 and WD_4. The numeral shown above each screen SC of Fig. 24 is the number for identifying its pattern. The numeral parenthesized within a circle in Fig. 25 corresponds to the identification number of the pattern. The screen SC of the CRT has a dimension of 640 dots in the horizontal or X-direction and 400 dots in the vertical or Y-direction, wherein the upper left corner coordinates are represented by (0, 0) while the lower right corner coordinates are represented by (639, 399). In Fig. 25, Pix and Piy represent the X and Y coordinates

of the graphic cursor Pi, respectively.

Referring first to Fig. 25, the position of the graphic cursor Pi may be at any coordinates on the screen at the start of window opening. To change the screen SC from pattern 1 to pattern 2, it is necessary that the X coordinate of the graphic cursor Pi is not "0" nor "399". To change the screen SC from pattern 2 to pattern 4, it is necessary that the X coordinate of the graphic cursor Pi is larger than "0" and smaller than "319" and the Y coordinate is larger than "0" and smaller than "199", or that the X coordinate is larger than "320" and smaller than "639" and the Y coordinate is larger than "0" and smaller than "199". Similarly, the pattern transition is conducted on the basis of the conditions described in the figure, wherein word "or" means a logical OR and word "and" means a logical AND. Fig. 26 is a transition graph for sequential window closing, wherein numerals parenthesized within a circle correspond to the pattern numbers of the screen SC of Fig. 24 similar to those in Fig. 25. For instance, to change the screen SC from pattern 8 to pattern 4, one of the windows WD_3 and WD_4 is deleted, or to change the screen SC from pattern 8 to pattern 4, one of the windows WD_1 and WD_2 is deleted. Other pattern transitions are also performed in accordance with the transition graph of Fig. 26.

Referring now back to Figs. 21 to 23, the three routines will be described with reference to

Figs. 24 to 26. Fig. 21 is the flow chart of the window open routine WOP which is executed when the graphic cursor Pi points the window open icon at step 17g of Fig. 17. First, at step 21a it is judged if the left button switch S_L of the mouse 140 is being pushed. If affirmative, the coordinate information of the graphic cursor Pi at that time is saved at step 21b. The memory area of this coordinate information may be a predetermined memory area in the RAM 2 or in the cursor memory area provided at the CRT controller 9. When the release of the left button switch S_L of the mouth 140 is detected at step 21a, then at step 21c information on the current pattern displayed on the screen SC is read from the pattern information memory area WP provided at the RAM 2. In this embodiment, information discriminating the current pattern from those shown in Fig. 24 is always stored in the RAM 2. At succeeding step 21d a window pattern is decided in accordance with the transition graph of Fig. 25, taking into consideration of the current window pattern information read at step 21c and the coordinate information saved at step 21b of the cursor Pi. The pattern information of the decided window is set at the window pattern information memory area WP. Steps 21e and 21f show the window renewing routine WRW where all the opened windows of the window pattern decided at step 21d are renewed.

Fig. 22 is the flow chart of the window close

routine WCP which is executed when the graphic cursor Pi points the window close icon at step 17i of Fig. 17. Steps 22a and 22b are the same with those in Fig. 21. At step 22c it is judged if the coordinates saved at step 22b of the graphic cursor Pi are within the icon area ICN on the screen SC. If not within the icon area ICN, the routine executes no further process. If within the icon area ICN, then at step 22d the current window information is read from the memory area WP of the RAM 2, similarly to the case of step 21c of Fig. 21. At succeeding step 22d a window pattern is decided in accordance with the transition graph of Fig. 26, taking into consideration of the current window pattern information read at step 22d and the coordinate information saved at step 22b of the cursor Pi. The pattern information of the decided window is set at the window pattern information memory area WP. Steps 22e and 22f show the window renewing routine WRW where all the opened windows of the window pattern decided at step 21d are renewed.

Fig. 23 is the flow chart of the window reorganizing routine WAR which is executed when the graphic cursor Pi points the window reorganizing icon 18 at step 17e of Fig. 17. To execute this routine, two memories at predetermined memory areas in the RAM 2 are used additionally. One is a retained window number memory RWNO for storing the window number of the window retained during window reorganization, and the other is

a temporarily window number memory TNO for temporarily storing a window number. The window numbers "1", "2", "3" and "4" are assigned to the respective windows WD_1, WD_2, WD_3 and WD_4, in this order.

Upon start of the window reorganizing routine WAR, the current window pattern information is read at step 23a, similarly to step 21c of Fig. 21 and to step 22d of Fig. 22. Next, at step 23b the window number of the window to be retained is decided based of the coordinates of the graphic cursor Pi on the screen SC and the window pattern information read at step 23a. The number information is set in the retained window number memory RWNO. At succeeding steps the window whose number set in the memory RWNO is displayed while the other windows are closed. To this purpose, first at step 23c the window number "1" is stored in the temporary window number memory TNO. Next, at step 23d it is judged if the content of the temporary window number memory TNO is more than 4 or not, since in this embodiment the number of windows is set as 4 at a maximum. At succeeding step 23e it is checked if the content of the retained window number memory RWNO coincides with that of the temporary window number memory TNO. The absence of a coincidence means that the window whose number has been stored in the temporary window number memory TNO is allowed to be closed. Therefore, at step 23f the icon of the window corresponding to the number stored in the temporary window

number memory TNO is displayed on the icon area ICN. At step 23g the content of the temporary window number memory TNO is incremented by 1 and step 23d follows. The procedure of steps 23d, 23e, 23f and 23g is repeated until the contents of the memories TNO and RWNO coincide each other at step 23e, or until the content of memory TNO exceeds 4 at step 23d. Upon coincidence between the contents of the memories TNO and RWNO, i.e., when the window to be retained is the window now concerned, then step 23h follows where the content of the window pattern memory WP is set as of pattern 1. At succeeding step 23i window renewal is executed by the window renewing routine WRW so that the pattern to be retained is displayed on the screen SC in the form of pattern 1 of Fig. 24. Next, the procedure advances to step 23g where window closing operations for the other windows are repeated.

The window renewing routine WRW shown in Figs. 1 and 12 performs a renewal of the contents of the display management memory areas DMM_1, DMM_2, DMM_3 and DMM_4, corresponding respectively to the windows WD_1, WD_2, WD_3 and WD_4, in accordance with the execution results of the respective routines WAR, WDL, WED, WOP and WCL. Thus, the contents of the display management memory areas DMM_1, DMM_2, DMM_3 and DMM_4 are renewed in accordance with the decided and stored window pattern information at steps 21d, 22e and 23h of Figs. 21 to 23. Assuming that the window WD_2 is intended to be

retained on the screen SC. The window renewing routine WRW performs the following procedure for renewal of the display management memory area DMM based on the execution result of the window reorganizing routine WAR: The upper left corner coordinates of the window WD_2 are stored in the memory area ML_2 of the display management memory area DMM_2, while the lower right corner coordinates are stored in the memory area MR_2. The other remaining display management memory areas DMM_1, DMM_3 and DMM_4 are all cleared.

As described previously, the display controller DCT controls to display the contents of the image memory areas VCB_1 to VCB_4 on the screen SC in accordance with the contents of the respective display management memory areas DMM_1 to DMM_4. Therefore, with the content of the display management memory area DMM_2 whose window WD_2 has been designated as a retained one, the following procedure is performed: The display controller DCT transfers only the content of the image memory area VCB_2 to the refreshing memory 10 of Fig. 2 for storage thereof, since the content of the window pattern information memory area is that of pattern 1 and the coordinate information is stored only in the display management memory area DMM_2. As a result, the window to be retained is displayed on the screen SC in the pattern configuration decided in accordance with the window pattern information.

In the above manner, of the windows opened to

the screen SC by using the window open routine WOP, only the window to be retained can be readily displayed and the other windows can be closed, by a simple operation pointing its reorganizing icon with the graphic cursor Pi through the screen position pointer 14.

In the above embodiments, although a CRT has been used as the display device, the present invention is not limited thereto but other various display devices such as a liquid crystal display device may also be used. Further, although the number of divided portions or windows of the screen has been set as 4 for convenience of description, an optional number may be used. The number of patterns of the windows is also optional. The greater the number of these windows and patterns are, the more advantageous effects the present invention affords.

Also, in the above embodiments, the designation of a window during window reorganizing has been conducted by using the screen position pointer 14 such as the mouth 140. However, other input devices such as a keyboard may be used for that purpose.

Moreover, although the image memory areas VCB_1 to VCB_4, the display management memory areas DMM_1 to DMM_4 and other areas have been allocated to the RAM 2, this physical allocation of memory areas is not limited thereto but any storage device may be used so long as it can be properly accessed.

CLAIMS

1. A apparatus for controlling a screen display comprising: a display device (1) having a display screen (SC); a plurality of image memory areas (VCB_1 - VCB_4) each corresponding to the screen; a display management memory area (DMM) for storing the positions on the screen of the storage contents of the image memory areas; display controller means (DCT) for displaying the storage content of the designated image memory area on a predetermined divided portion of the screen, based on the storage content of the display management memory area; designating means (14) for designating a desired one of the divided portions of the screen, said divided portions of the screen corresponding respectively to said plurality of image memory areas; and display management means (WMG) for renewing the storage content of the display management area in such a way that the position on the screen of the image memory area corresponding to the divided portion designated by the designating means is located on the entirety of the screen.

2. An apparatus for controlling a screen display according to claim 1, wherein said designating means is coordinate designating means for designating the coordinates of a representation indicating a divided portion of the screen.

3. An apparatus for controlling a screen display according to claim 1, wherein said display management

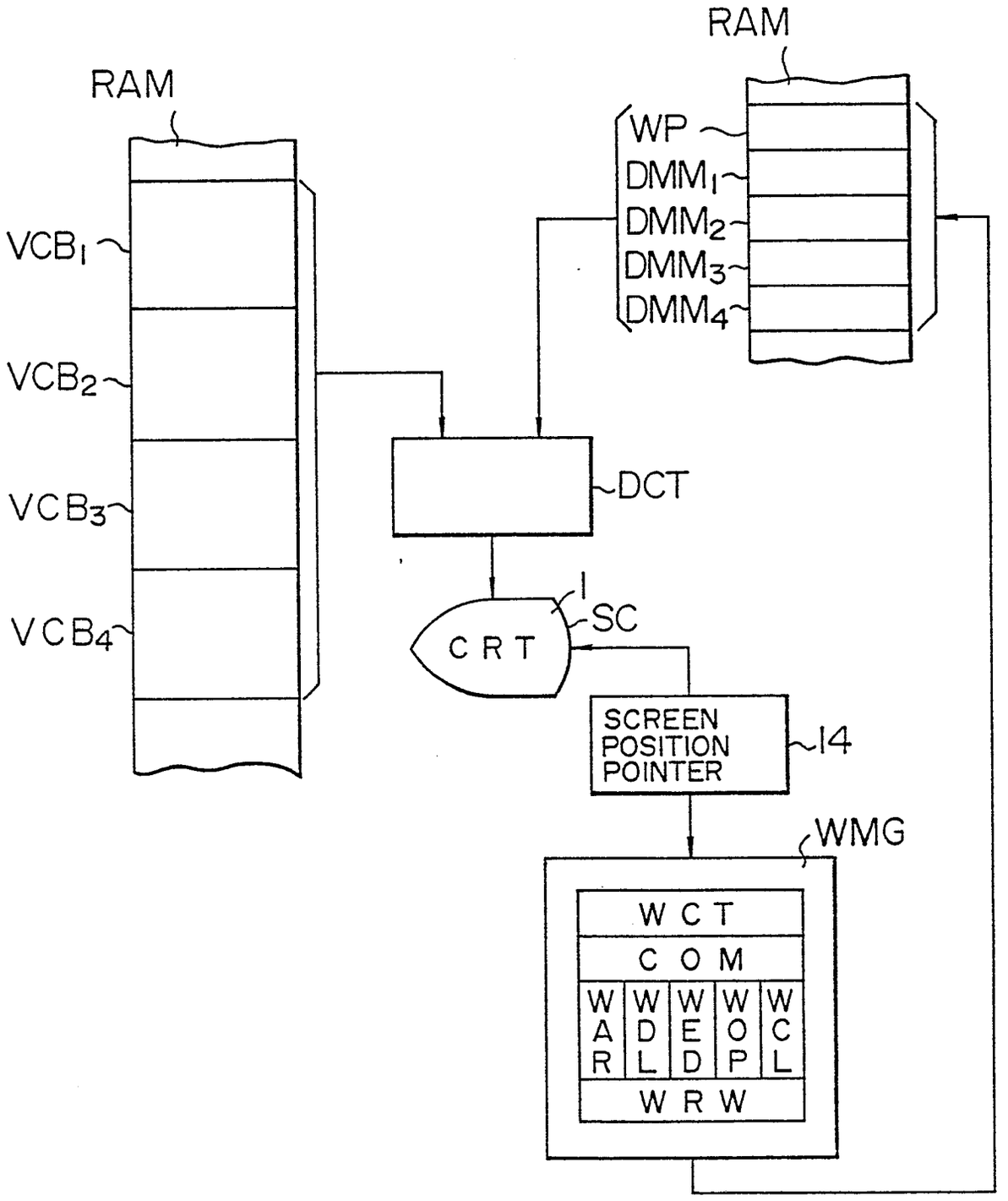
means comprises a memory area (DMM_1 - DMM_4) for storing information corresponding to a divided portion of the screen designated by said designation means, and renews the storage content of the display management memory area based on the storage content of the memory area.

4. An apparatus for controlling a screen display according to claim 1, wherein said display management means comprises a first memory area for storing information corresponding to a divided portion of the screen designated by said designation means; a second memory area for sequentially storing information corresponding to divided portions; and judgement means for judging through comparison between the storage contents of the first and second memory areas, whereby said display management means renews the storage content of the display management memory area based on the judgement result.

5. An apparatus for controlling a screen display according to claim 4, wherein said display management means comprises renewing means for renewing the storage content of said second memory area to information corresponding to renewed divided portions, when said judgement means completes its judgement.

0 206 330

FIG. 1



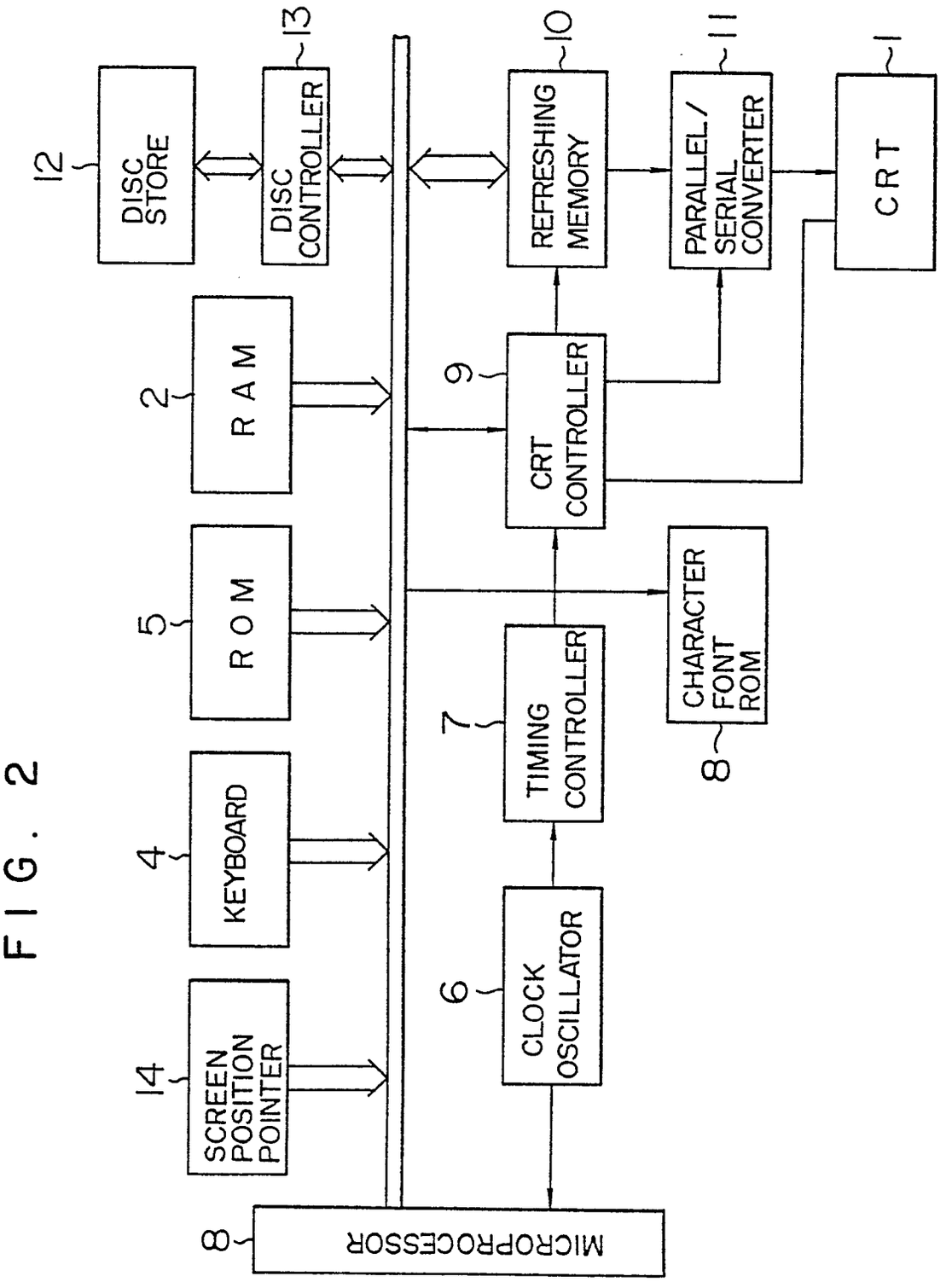

MICROPROCESSOR
8
SCREEN POSITION POINTER
14
KEYBOARD
4
ROM
5
RAM
2
DISC STORE
12
DISC CONTROLLER
13
CLOCK OSCILLATOR
6
TIMING CONTROLLER
7
CRT CONTROLLER
9
CHARACTER FONT ROM
8
REFRESHING MEMORY
10
PARALLEL / SERIAL CONVERTER
11
CRT
1

FIG. 2

FIG. 3

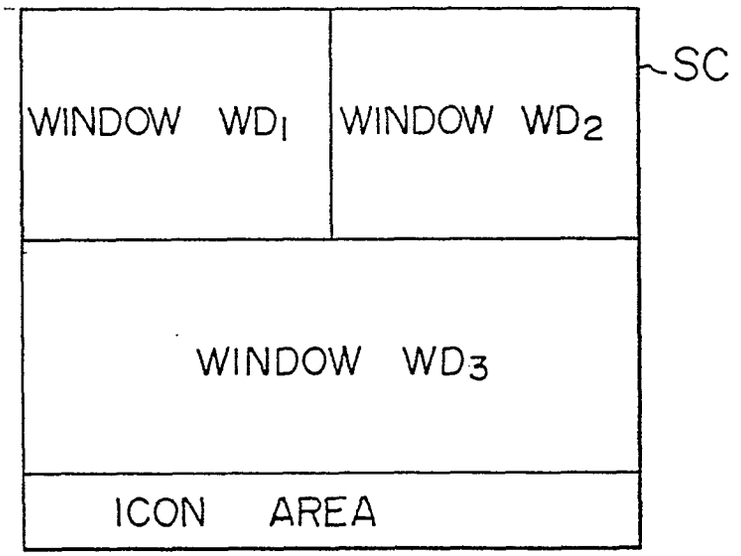
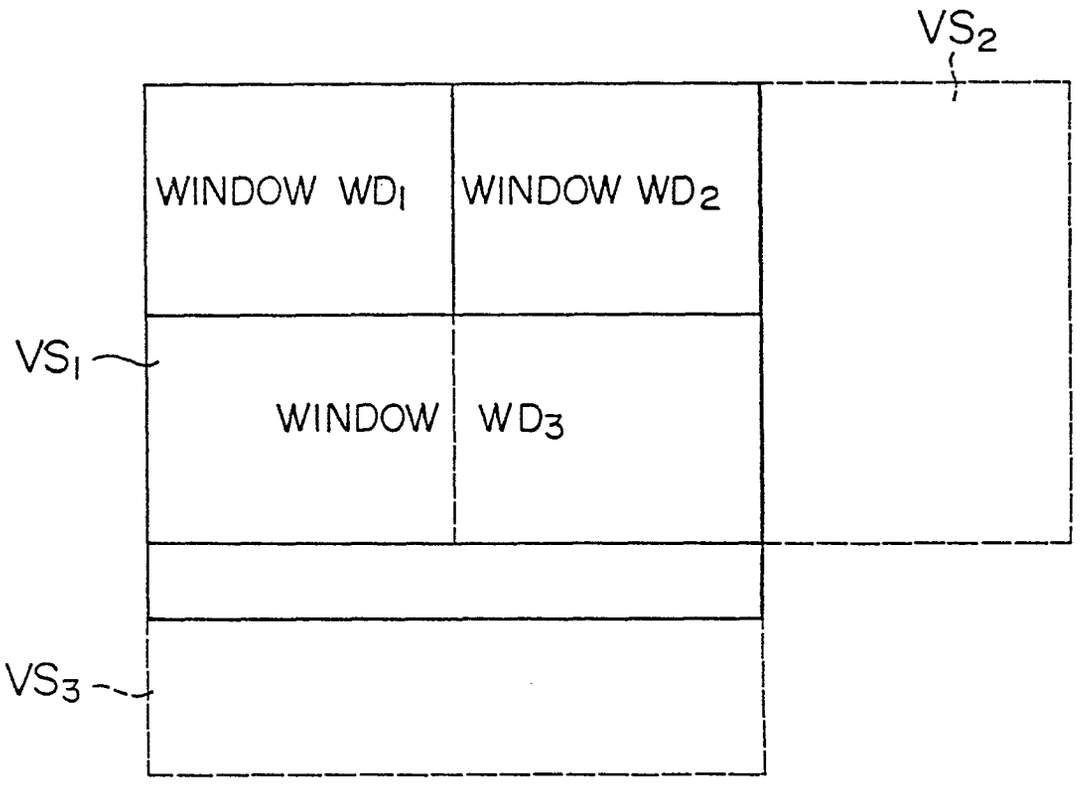

WINDOW WD1
WINDOW WD2
SC
WINDOW WD3
ICON AREA

FIG. 4


VS2
WINDOW WD1
WINDOW WD2
VS1
WINDOW
WD3
VS3

0 206 330

FIG. 5

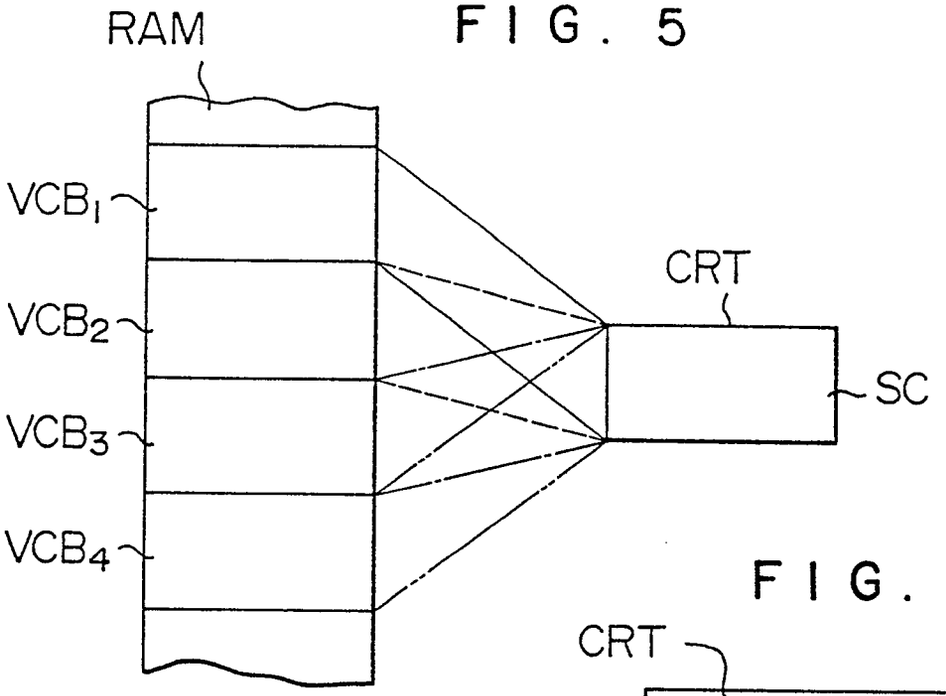


FIG. 6

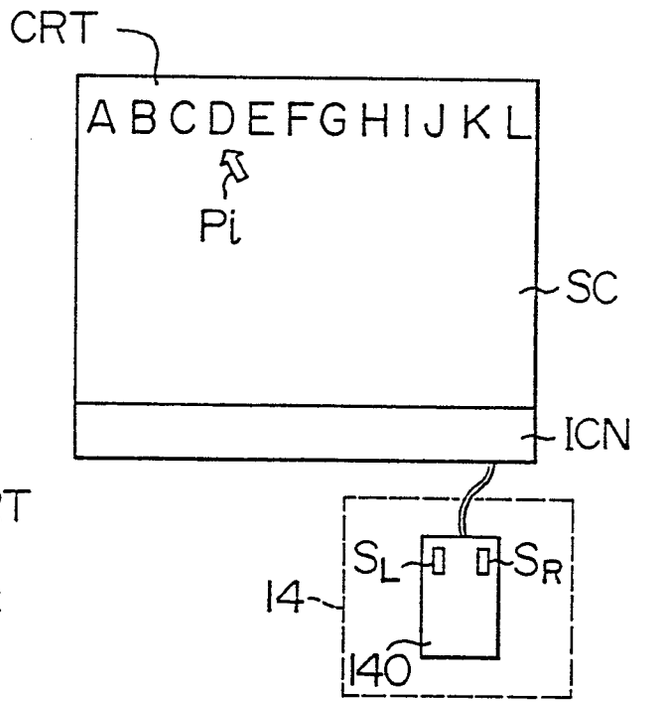


FIG. 7

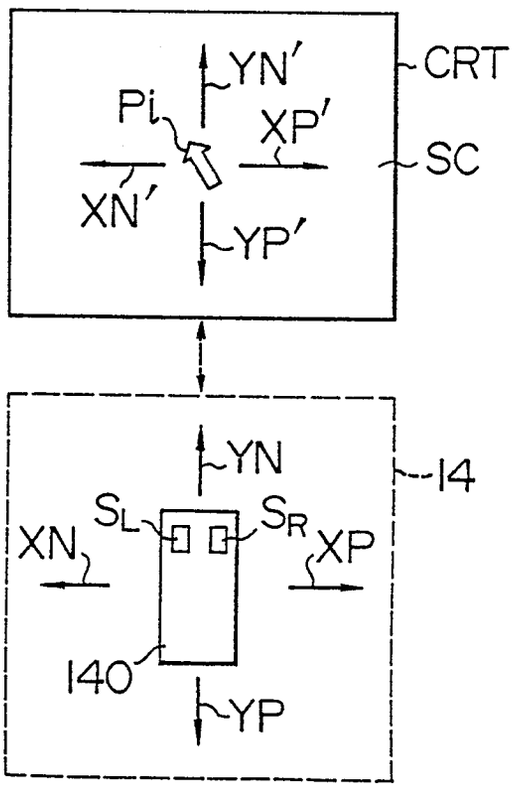


FIG. 8(a)

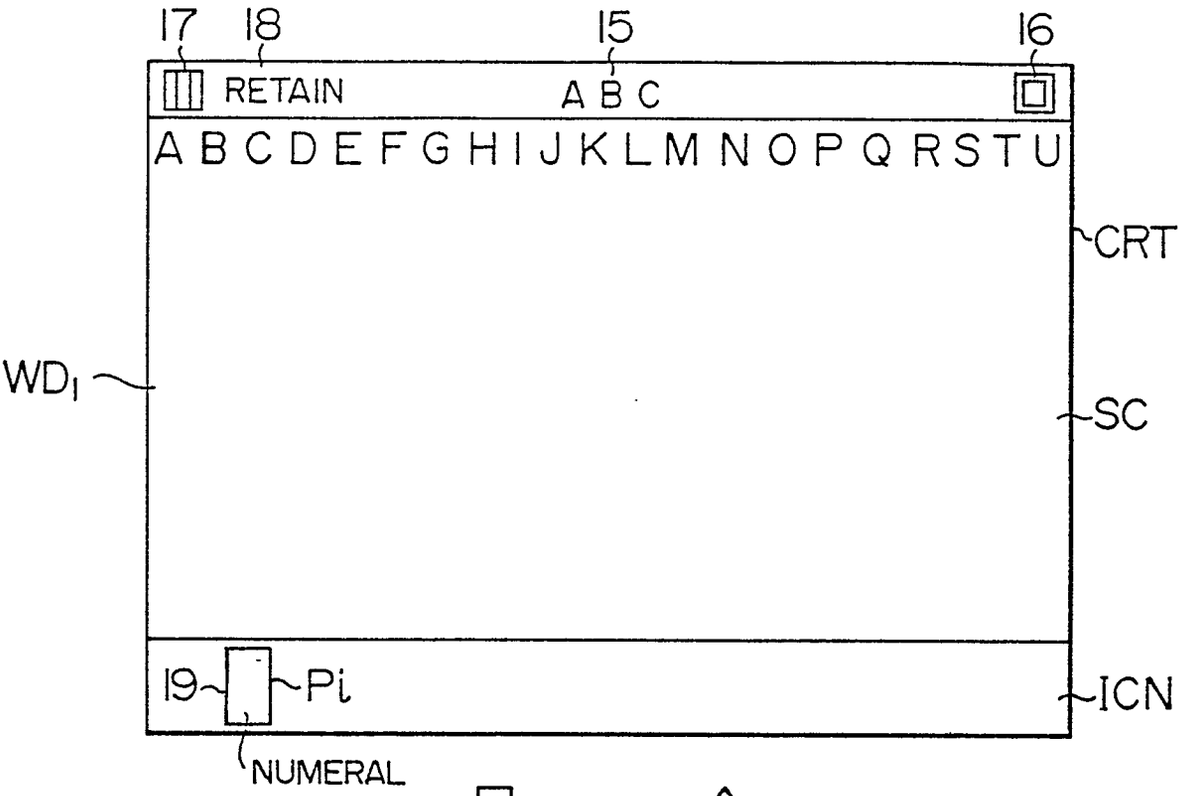

17
18
15
16
RETAIN
A B C
A B C D E F G H I J K L M N O P Q R S T U
CRT
WD1
SC
19
Pi
ICN
NUMERAL

FIG. 8(b)

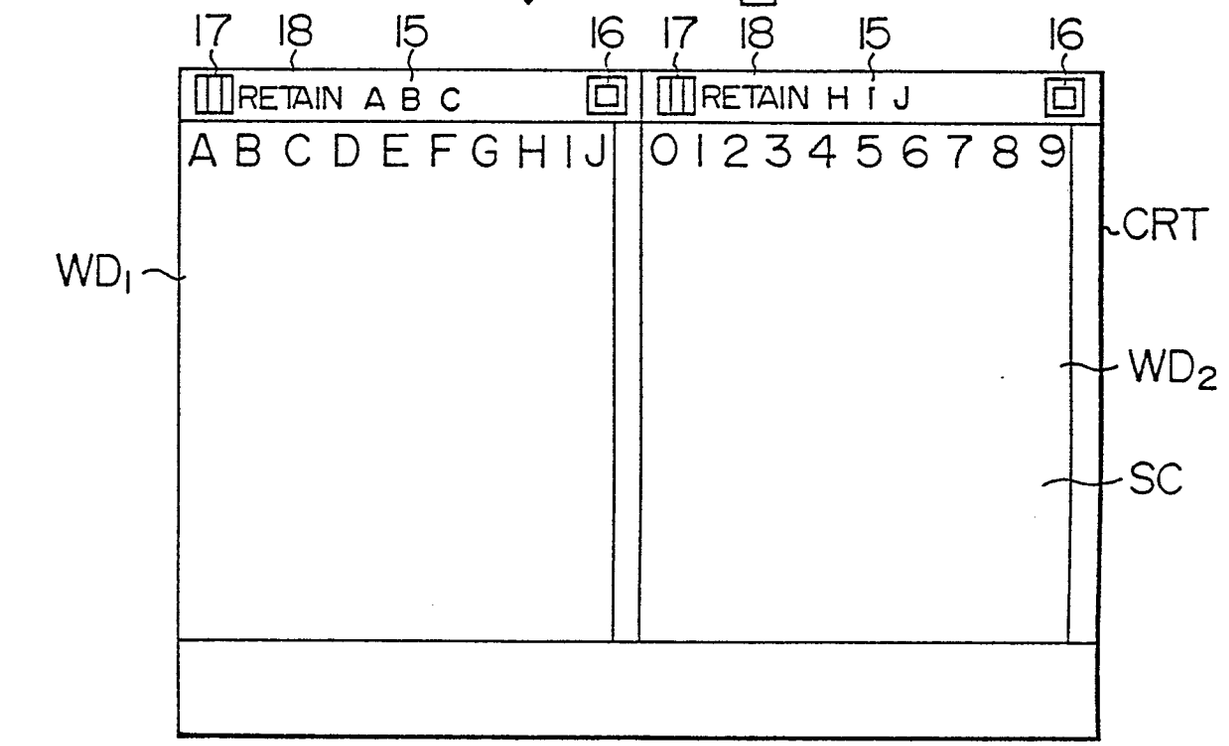

17
18
15
16
RETAIN A B C
RETAIN H I J
A B C D E F G H I J
0 1 2 3 4 5 6 7 8 9
CRT
WD1
WD2
SC

FIG. 9

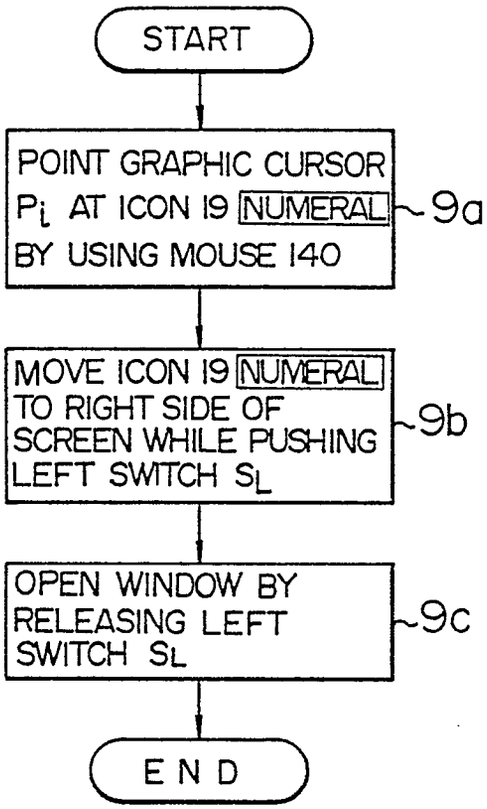


FIG. 10

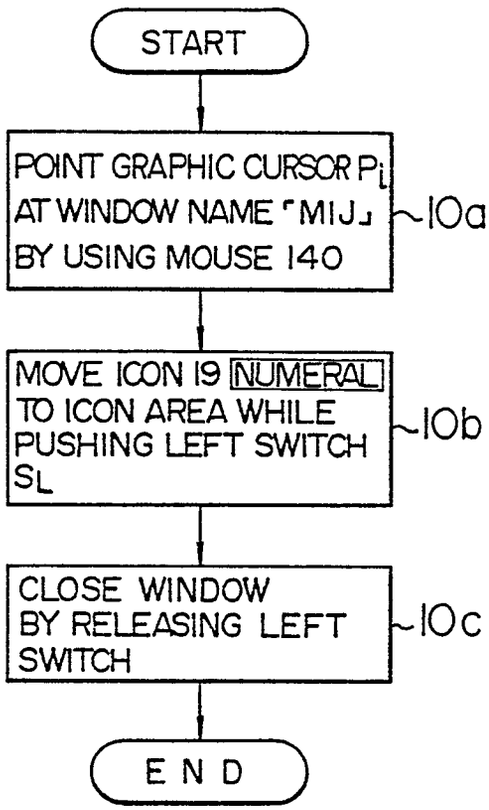


FIG. 11

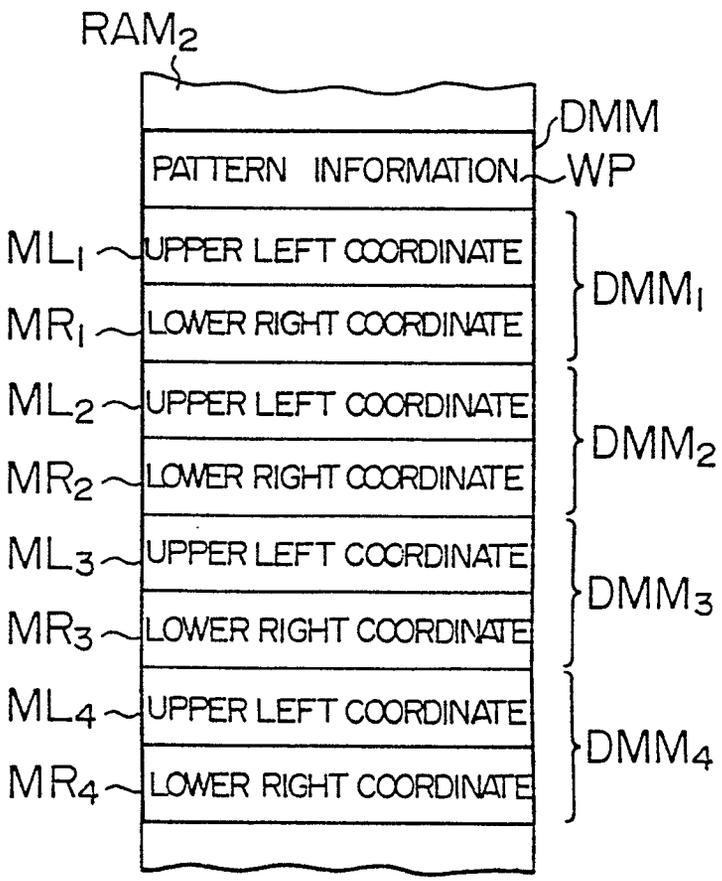


FIG. 12

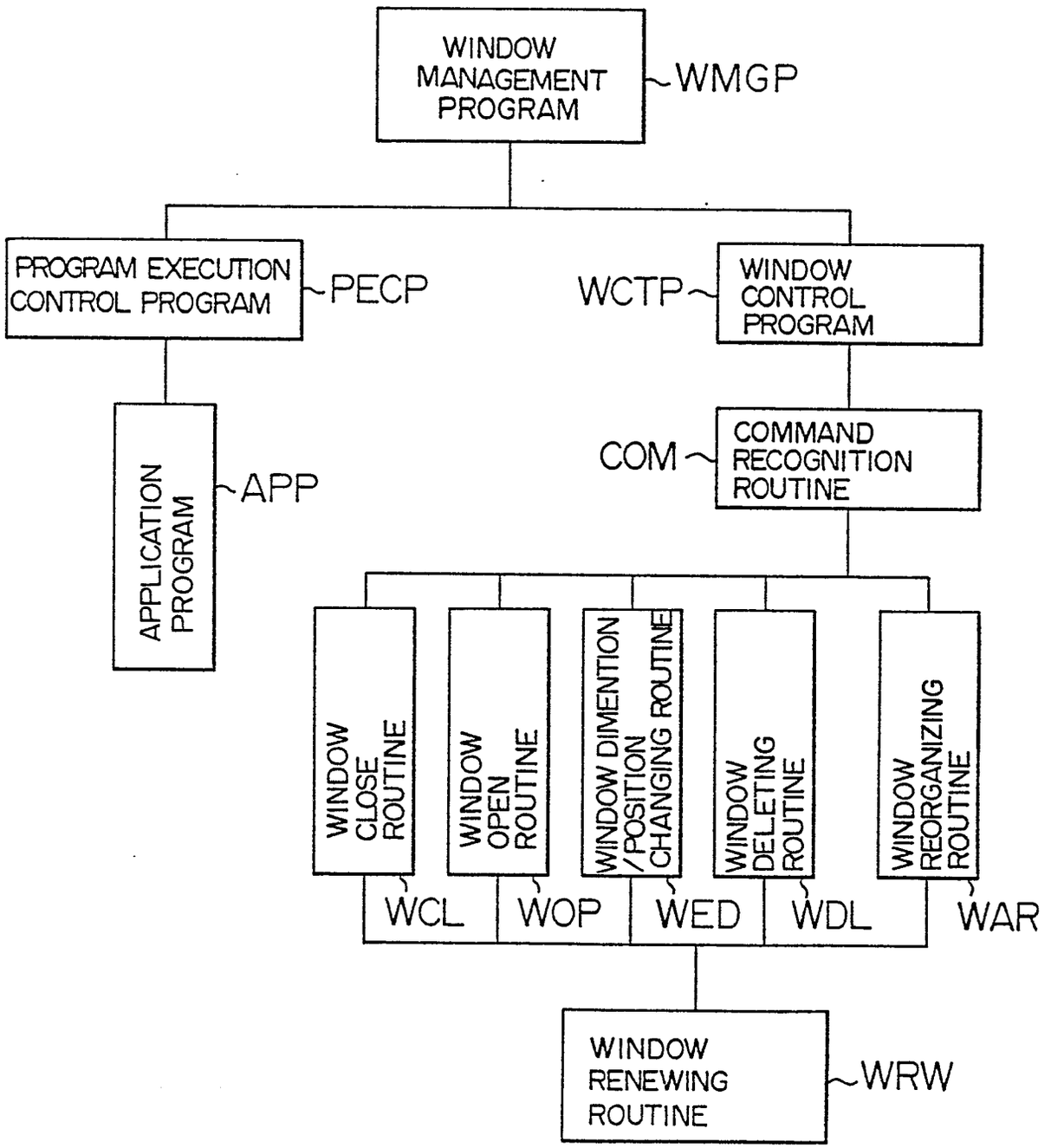

WINDOW MANAGEMENT PROGRAM
WMGP
PROGRAM EXECUTION CONTROL PROGRAM
PECP
WCTP
WINDOW CONTROL PROGRAM
APPLICATION PROGRAM
APP
COM
COMMAND RECOGNITION ROUTINE
WINDOW CLOSE ROUTINE
WINDOW OPEN ROUTINE
WINDOW DIMENTION /POSITION CHANGING ROUTINE
WINDOW DELETING ROUTINE
WINDOW REORGANIZING ROUTINE
WCL
WOP
WED
WDL
WAR
WINDOW RENEWING ROUTINE
WRW

FIG. 13(a)

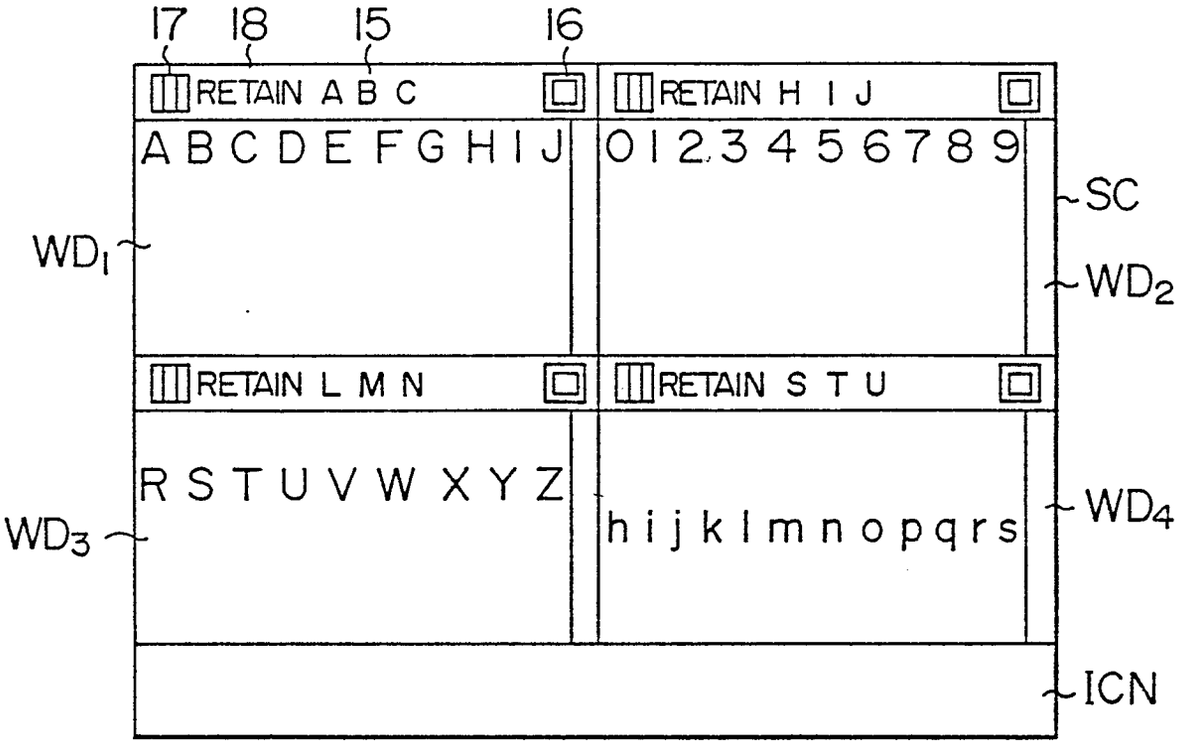


FIG. 13(b)

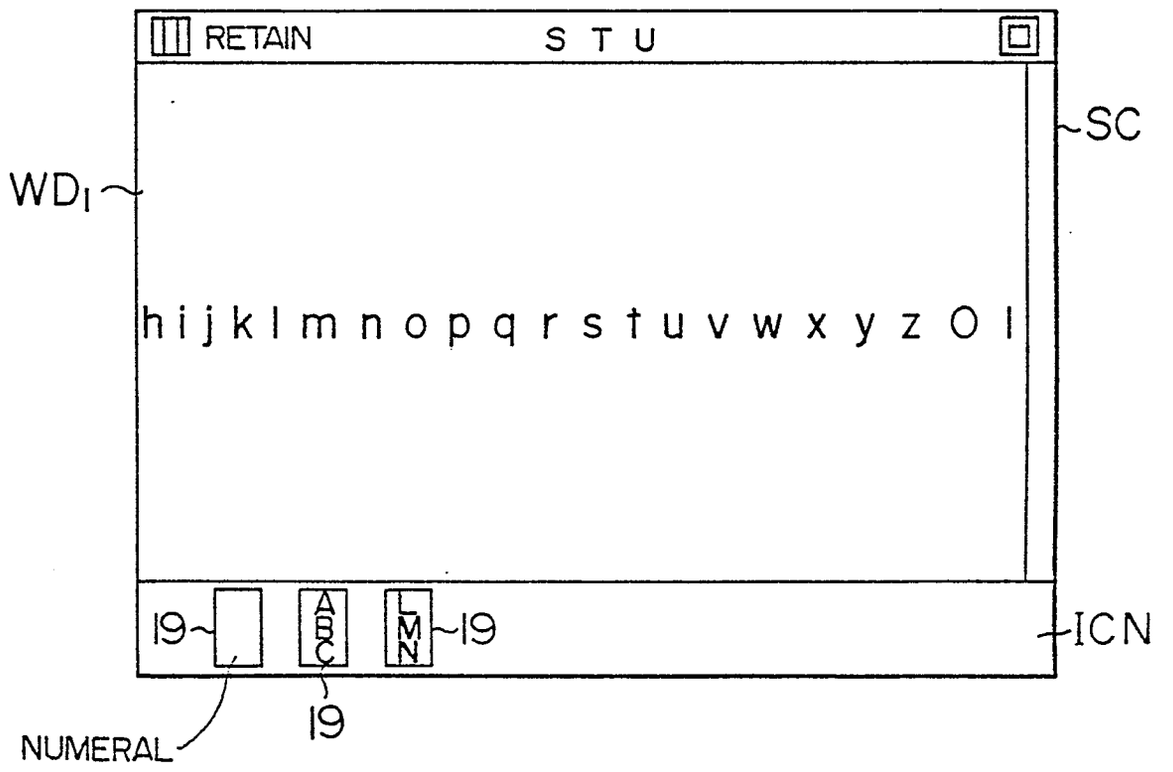


FIG. 14

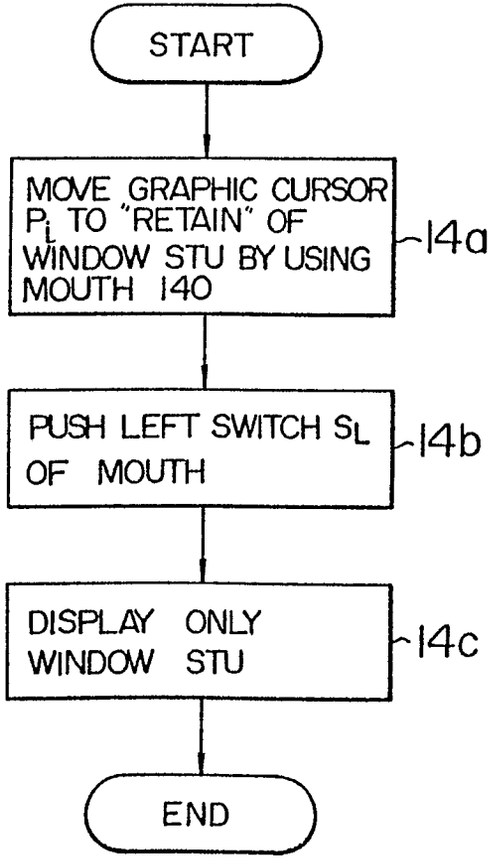


FIG. 15

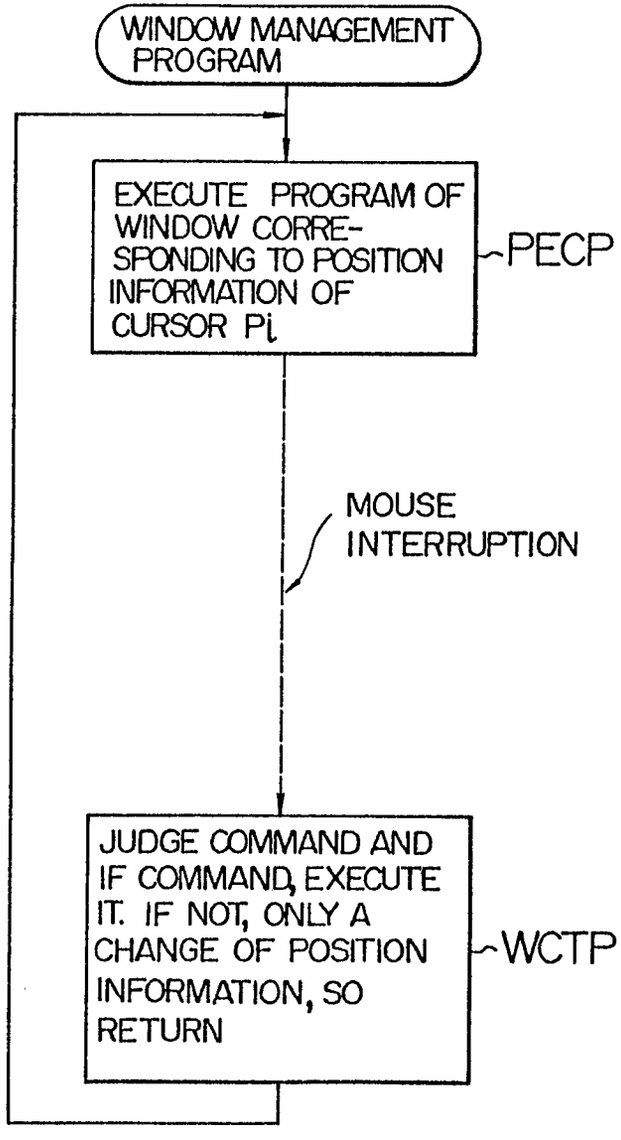


FIG. 16

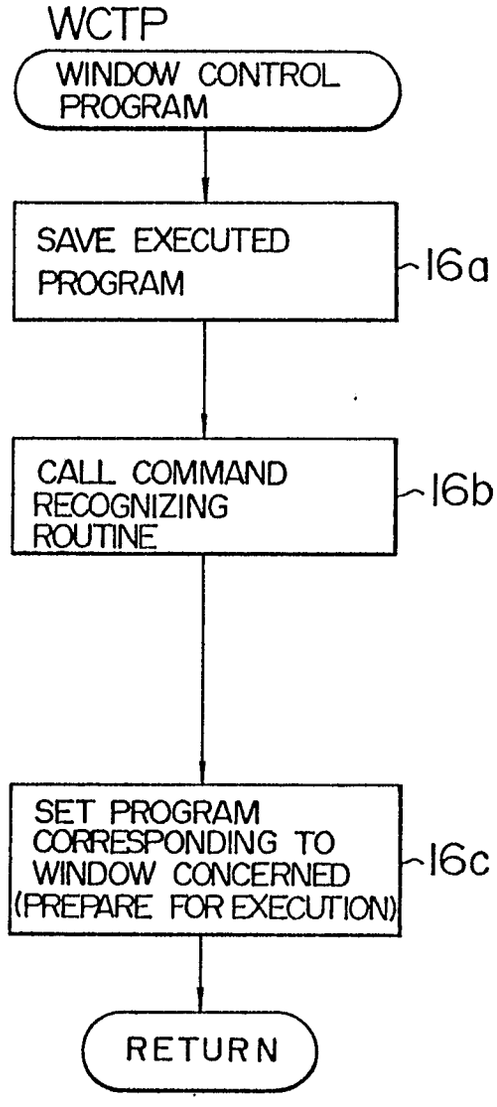


FIG. 17

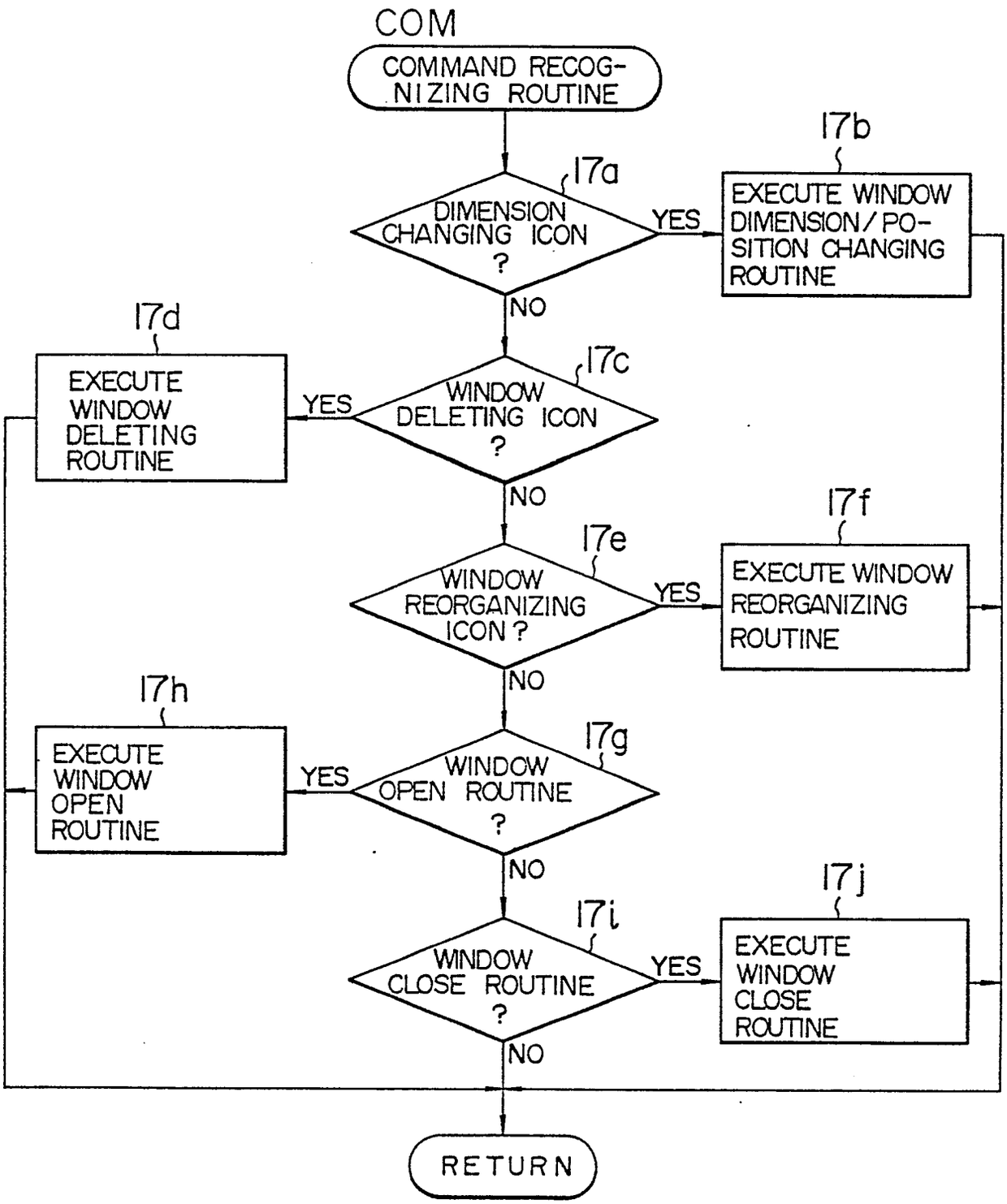

COM
COMMAND RECOG-NIZING ROUTINE
17a
DIMENSION CHANGING ICON ?
YES
17b
EXECUTE WINDOW DIMENSION/POSITION CHANGING ROUTINE
NO
17c
WINDOW DELETING ICON ?
YES
17d
EXECUTE WINDOW DELETING ROUTINE
NO
17e
WINDOW REORGANIZING ICON ?
YES
17f
EXECUTE WINDOW REORGANIZING ROUTINE
NO
17g
WINDOW OPEN ROUTINE ?
YES
17h
EXECUTE WINDOW OPEN ROUTINE
NO
17i
WINDOW CLOSE ROUTINE ?
YES
17j
EXECUTE WINDOW CLOSE ROUTINE
NO
RETURN

FIG. 18

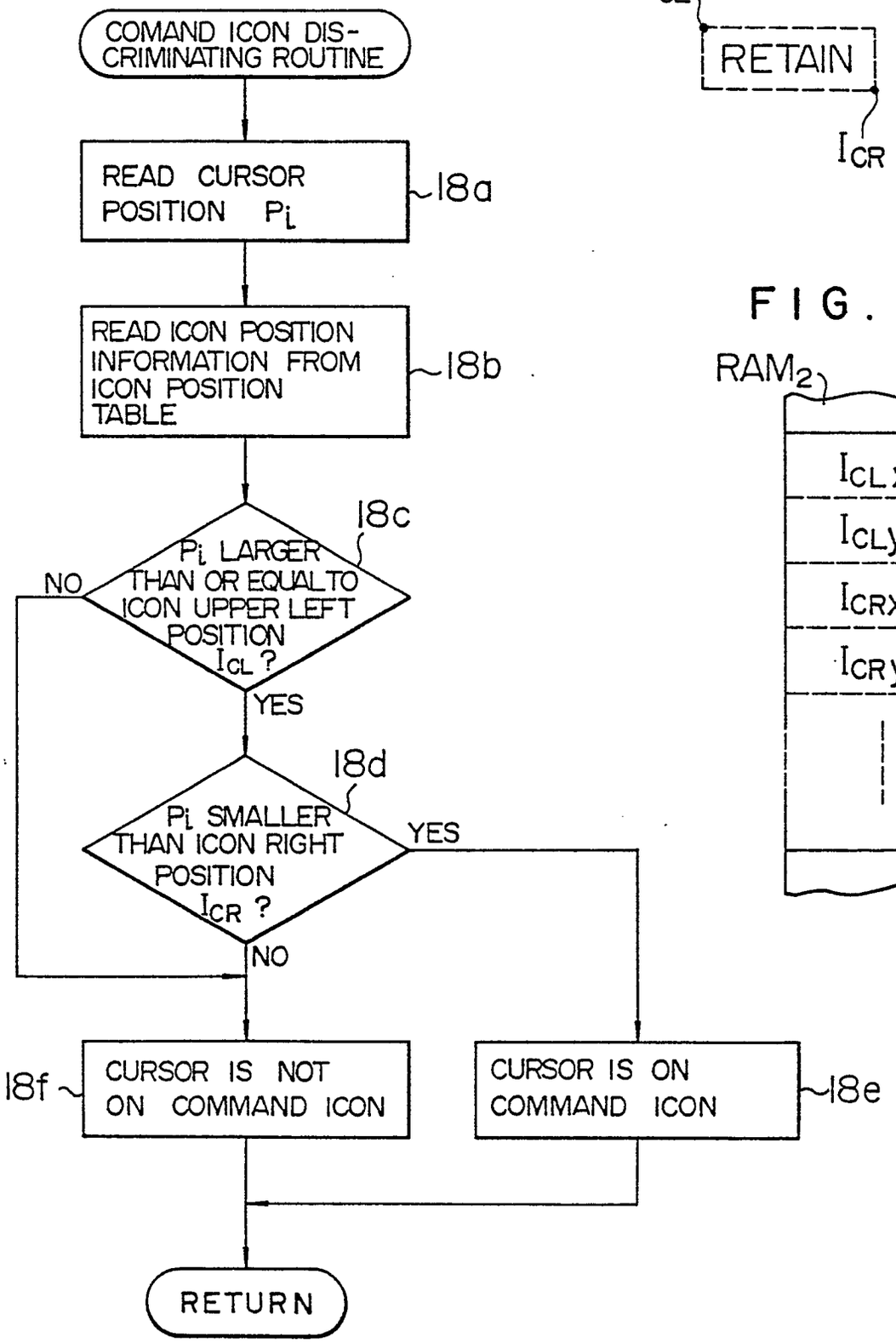


FIG. 19

FIG. 20

FIG. 21

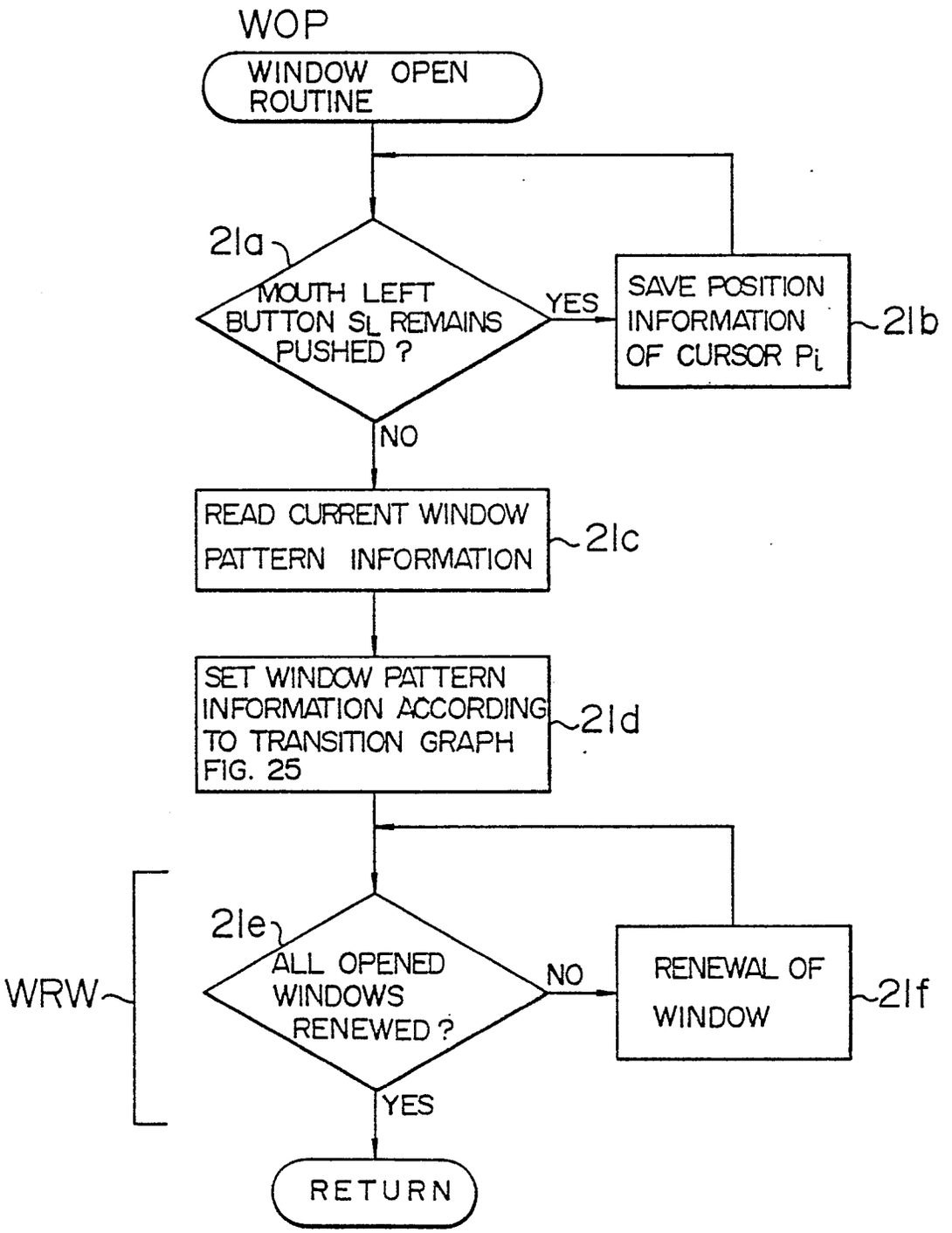

WOP
WINDOW OPEN ROUTINE
21a
MOUTH LEFT BUTTON SL REMAINS PUSHED ?
YES
SAVE POSITION INFORMATION OF CURSOR Pi
21b
NO
READ CURRENT WINDOW PATTERN INFORMATION
21c
SET WINDOW PATTERN INFORMATION ACCORDING TO TRANSITION GRAPH FIG. 25
21d
WRW
21e
ALL OPENED WINDOWS RENEWED ?
NO
RENEWAL OF WINDOW
21f
YES
RETURN

FIG. 22

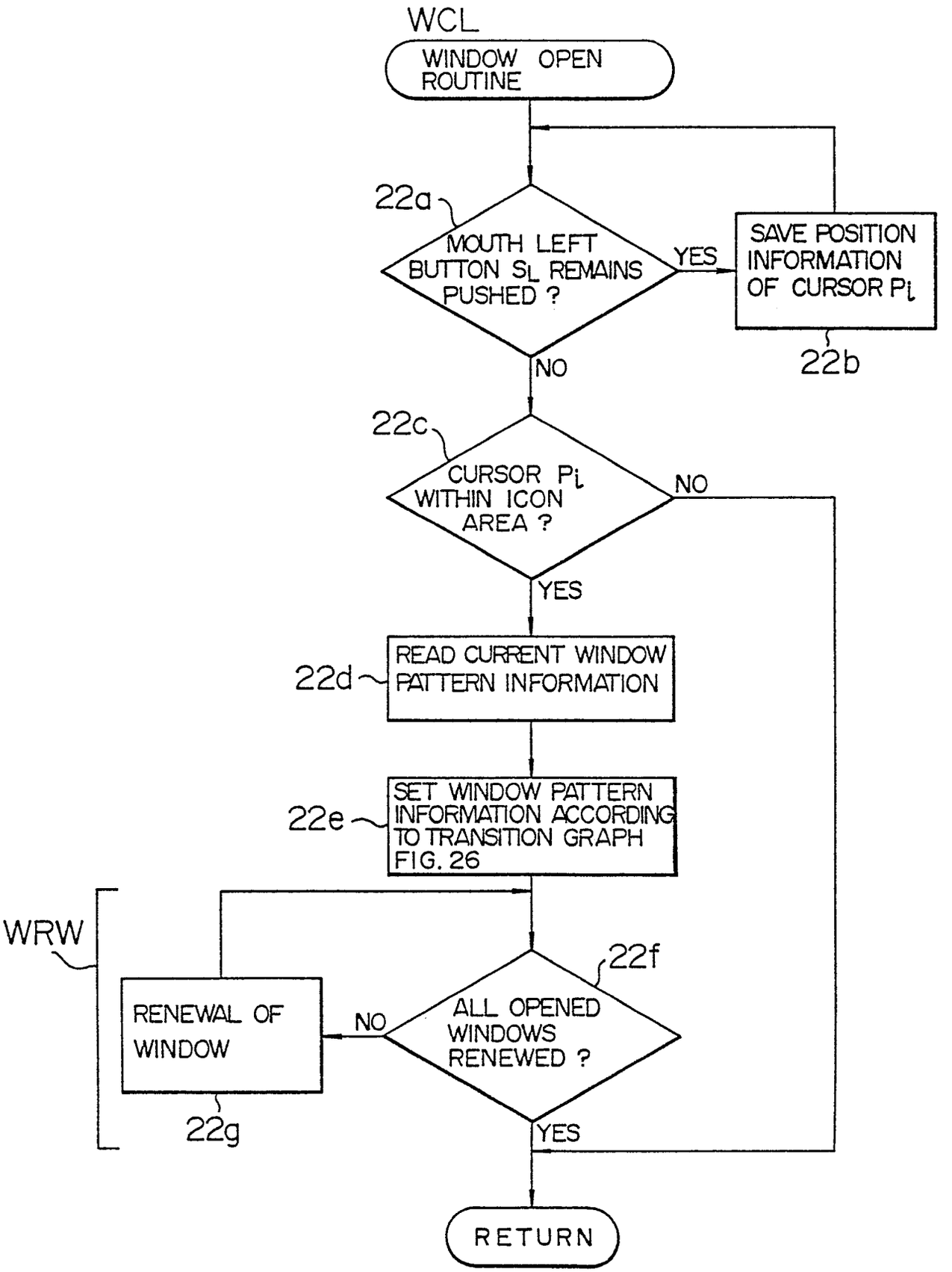

WCL
WINDOW OPEN ROUTINE
22a
MOUTH LEFT BUTTON S_L REMAINS PUSHED ?
YES
SAVE POSITION INFORMATION OF CURSOR P_i
22b
NO
22c
CURSOR P_i WITHIN ICON AREA ?
NO
YES
22d
READ CURRENT WINDOW PATTERN INFORMATION
22e
SET WINDOW PATTERN INFORMATION ACCORDING TO TRANSITION GRAPH FIG. 26
WRW
22f
ALL OPENED WINDOWS RENEWED ?
NO
RENEWAL OF WINDOW
22g
YES
RETURN

FIG. 23

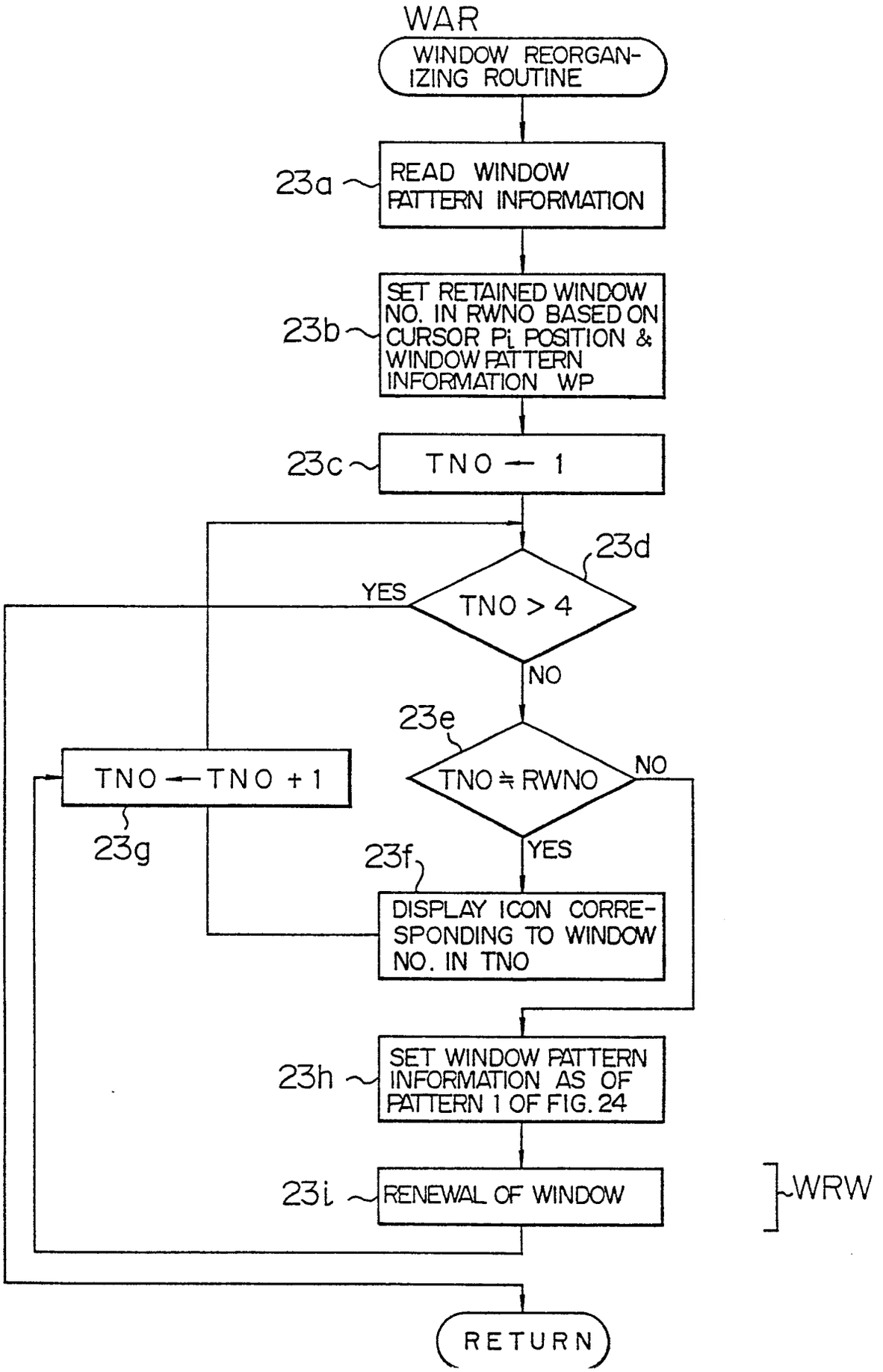
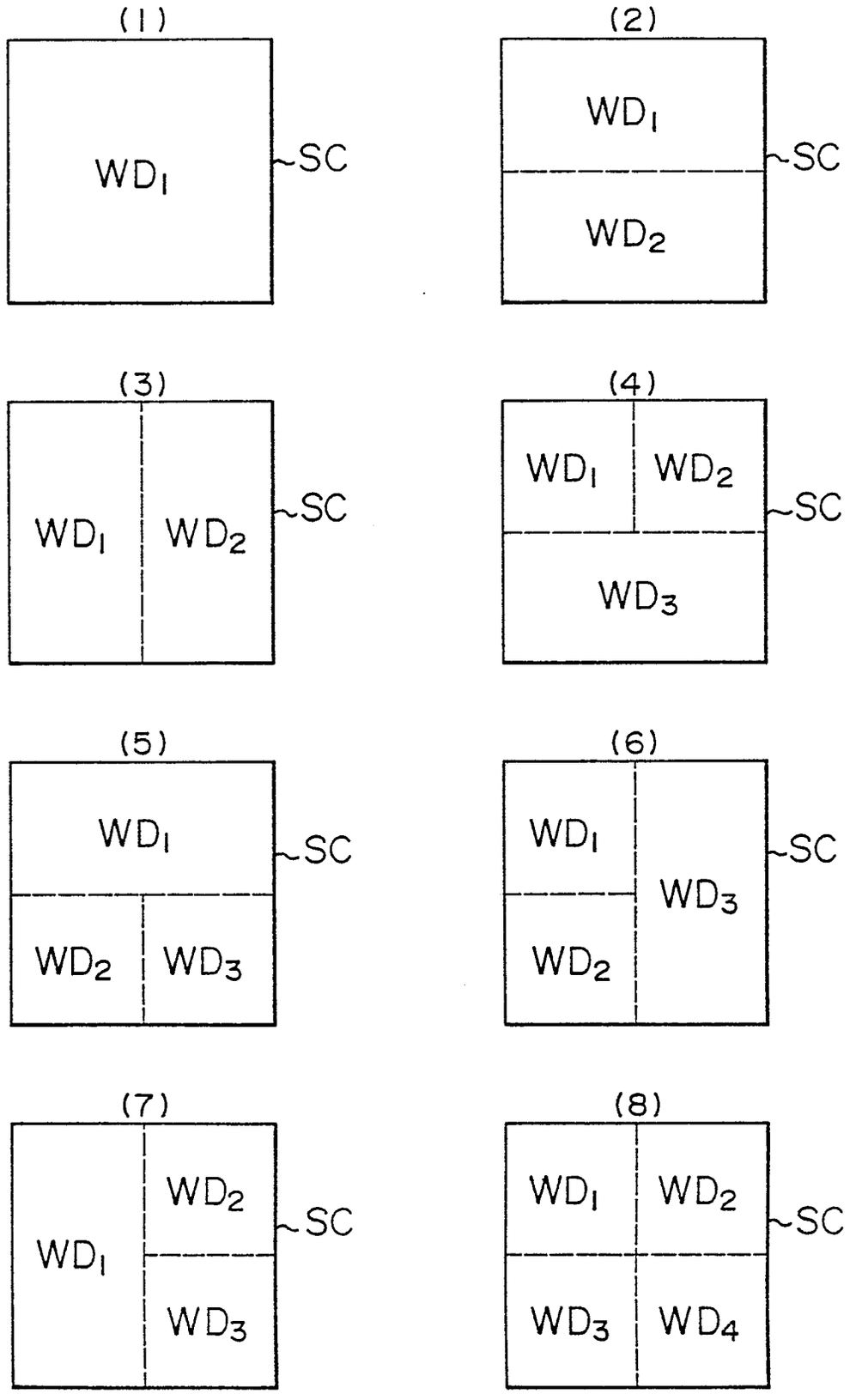

WAR
WINDOW REORGANIZING ROUTINE
23a
READ WINDOW PATTERN INFORMATION
23b
SET RETAINED WINDOW NO. IN RWNO BASED ON CURSOR Pi POSITION & WINDOW PATTERN INFORMATION WP
23c
TNO ← 1
23d
YES
TNO > 4
NO
23e
TNO ≒ RWNO
NO
YES
TNO ← TNO + 1
23g
23f
DISPLAY ICON CORRESPONDING TO WINDOW NO. IN TNO
23h
SET WINDOW PATTERN INFORMATION AS OF PATTERN 1 OF FIG. 24
23i
RENEWAL OF WINDOW
WRW
RETURN

FIG. 24


(1)
WD1
SC
(2)
WD1
WD2
SC
(3)
WD1
WD2
SC
(4)
WD1
WD2
WD3
SC
(5)
WD1
WD2
WD3
SC
(6)
WD1
WD2
WD3
SC
(7)
WD1
WD2
WD3
SC
(8)
WD1
WD2
WD3
WD4
SC

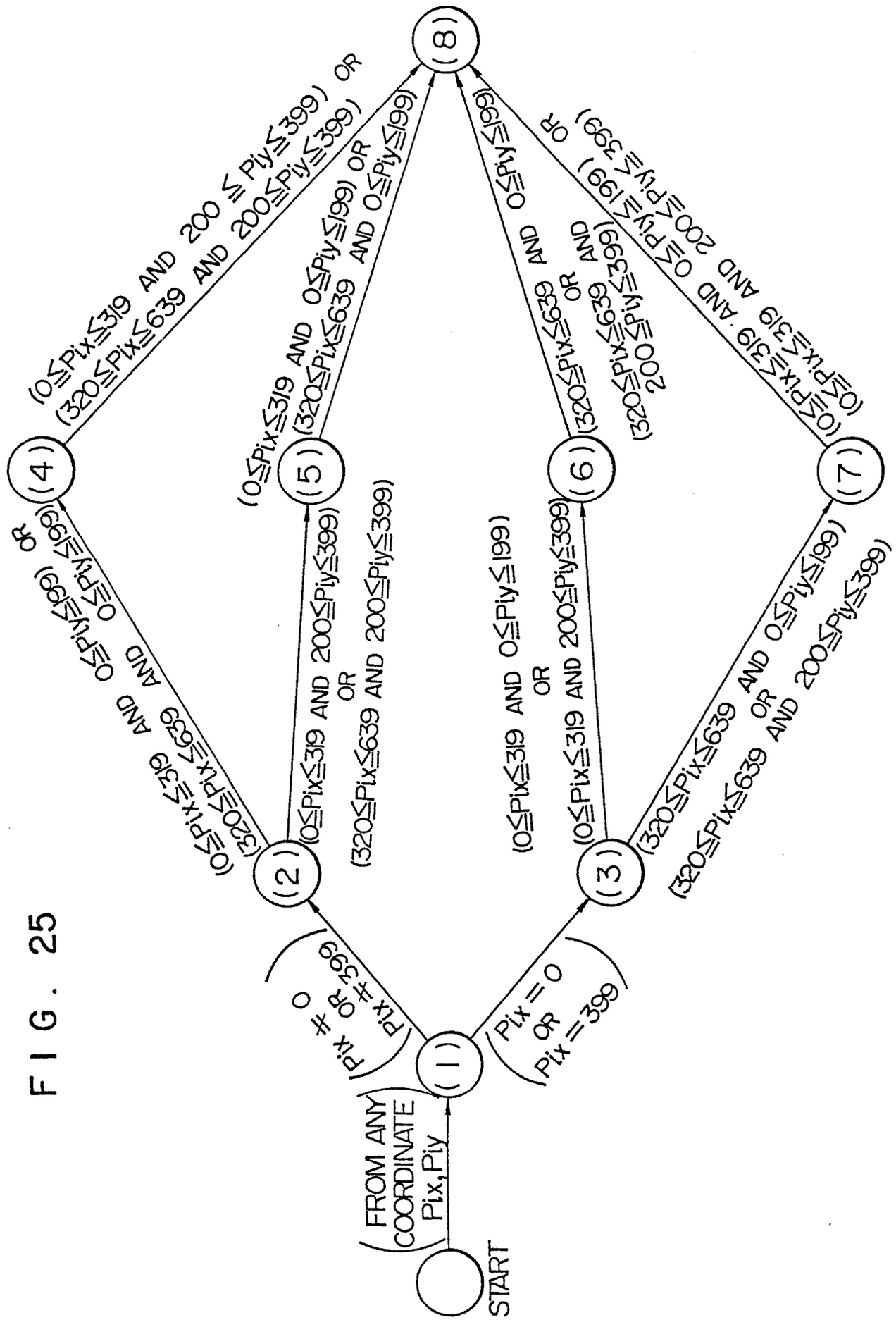

START
FROM ANY COORDINATE Pix,Piy
(1)
Pix ≠ 0 & Pix ≠ 639
Pix = 0 OR Pix = 399
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(0≦Pix≦319 AND 0≦Piy≦199) OR (320≦Pix≦639 AND 0≦Piy≦199)
(0≦Pix≦319 AND 200≦Piy≦399) OR (320≦Pix≦639 AND 200≦Piy≦399)
(0≦Pix≦319 AND 0≦Piy≦199) OR (320≦Pix≦639 AND 200≦Piy≦399)
(0≦Pix≦319 AND 200≦Piy≦399) OR (320≦Pix≦639 AND 0≦Piy≦199)
(0≦Pix≦319 AND 0≦Piy≦199)
OR
(320≦Pix≦639 AND 200≦Piy≦399)
(0≦Pix≦319 AND 200≦Piy≦399)
OR
(320≦Pix≦639 AND 0≦Piy≦199)
(320≦Pix≦639 AND 0≦Piy≦199) OR (0≦Pix≦319 AND 200≦Piy≦399)

FIG. 25

FIG. 26

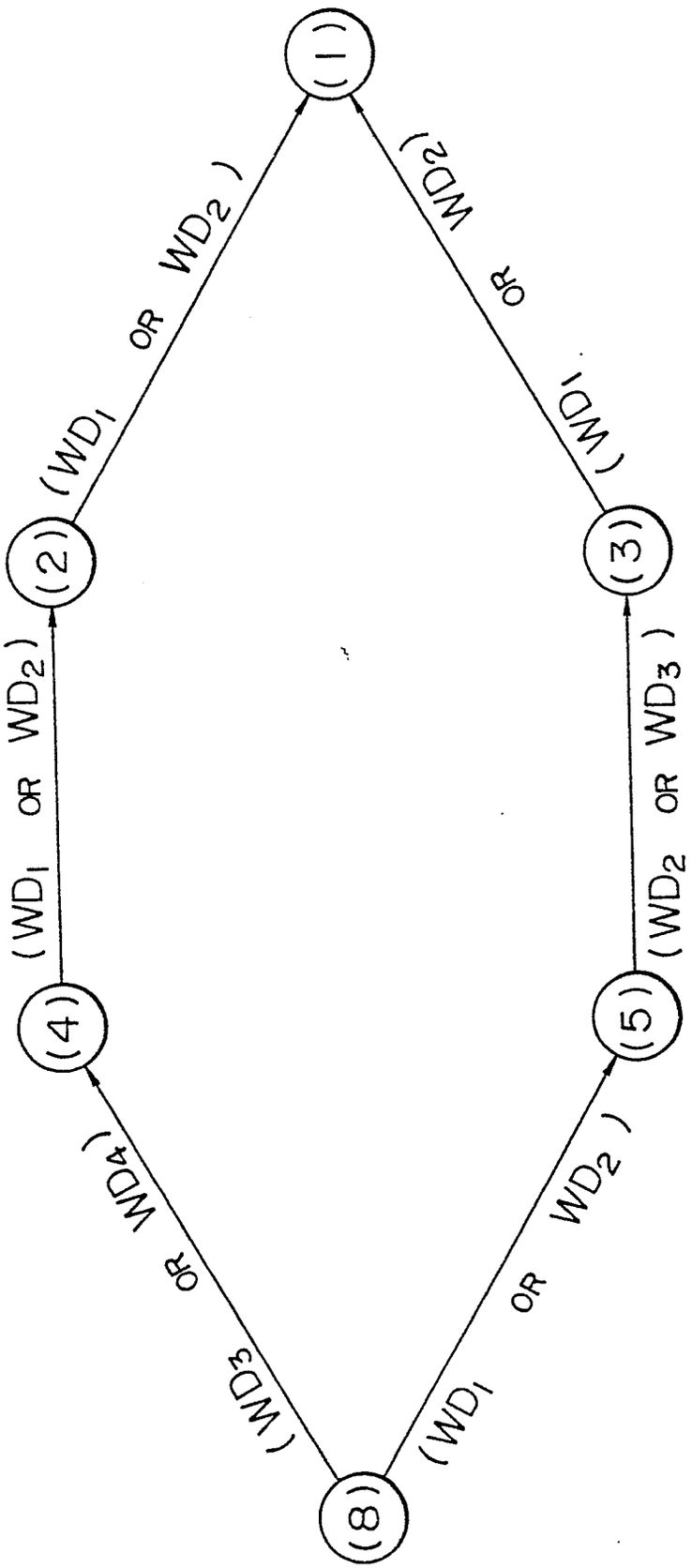
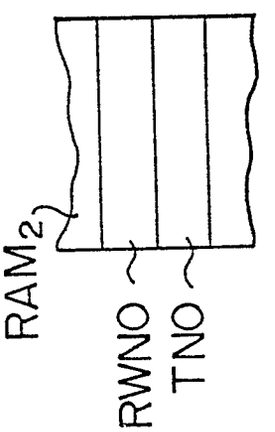

(1)
(2)
(3)
(4)
(5)
(8)
(WD1 OR WD2)
(WD1 OR WD2)
(WD1 OR WD2)
(WD1 OR WD2)
(WD2 OR WD3)
(WD3 OR WD4)

FIG. 27


RAM2
RWNO
TNO