

EXHIBIT 1.12

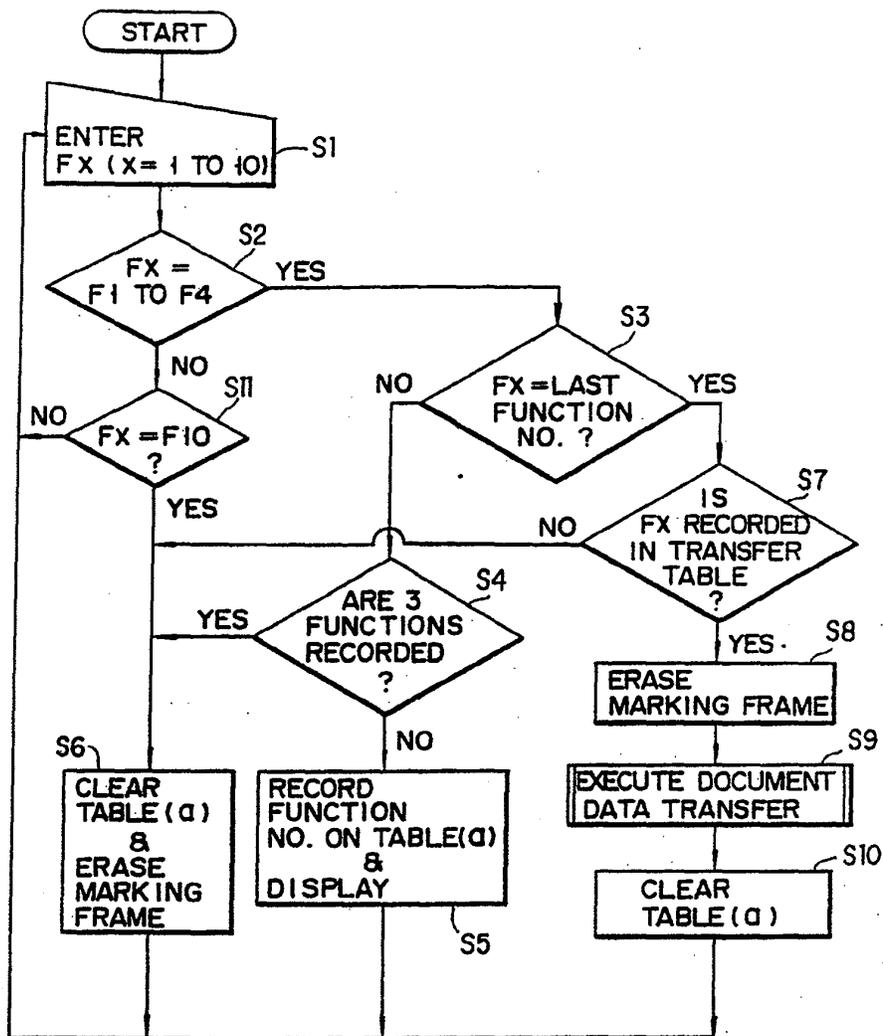


FIG. 19

(a) DESIGNATION TABLE

NUMBER OF DESIGNATED ICONS	1ST	2ND	3RD

(b) TRANSFER TABLE

NO	NUMBER OF DESIGNATED ICONS	1ST	2ND	3RD
1	2	SCANNER F1	DISPLAY F3	
2	2	SCANNER F1	PRINTER F2	
3	2	SCANNER F1	FILE F4	
4	2	FILE F4	DISPLAY F3	
5	2	FILE F4	PRINTER F2	
6	2	DISPLAY F3	PRINTER F2	
7	2	DISPLAY F3	FILE F4	
8	3	SCANNER F1	DISPLAY F3	PRINTER F2
9	3	SCANNER F1	DISPLAY F3	FILE F4
10	3	FILE F4	DISPLAY F3	PRINTER F2

F I G. 20

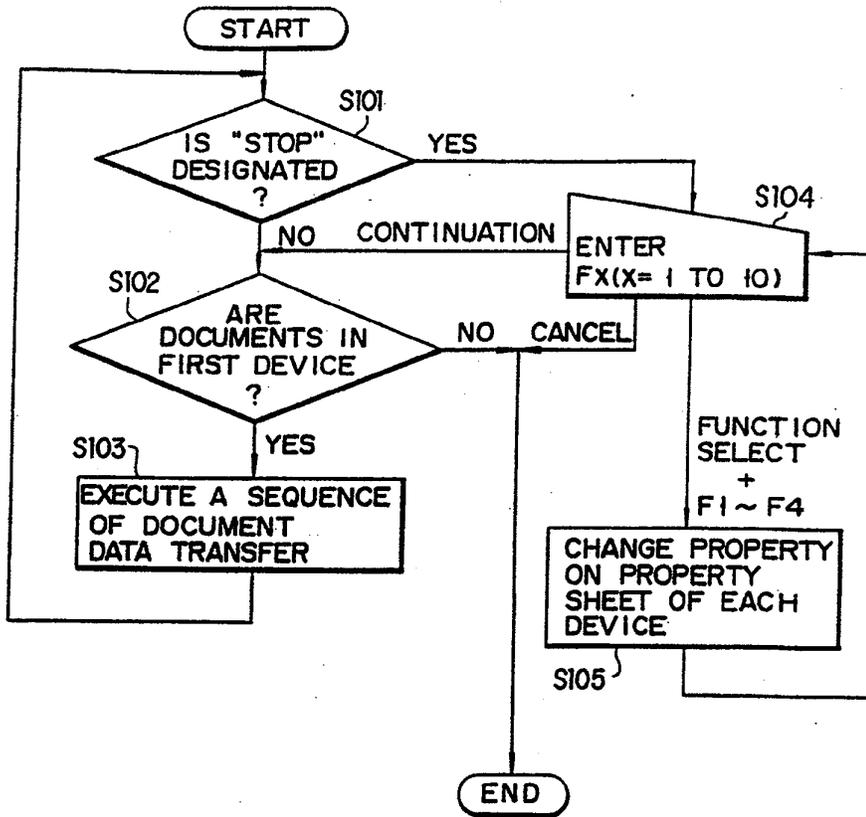
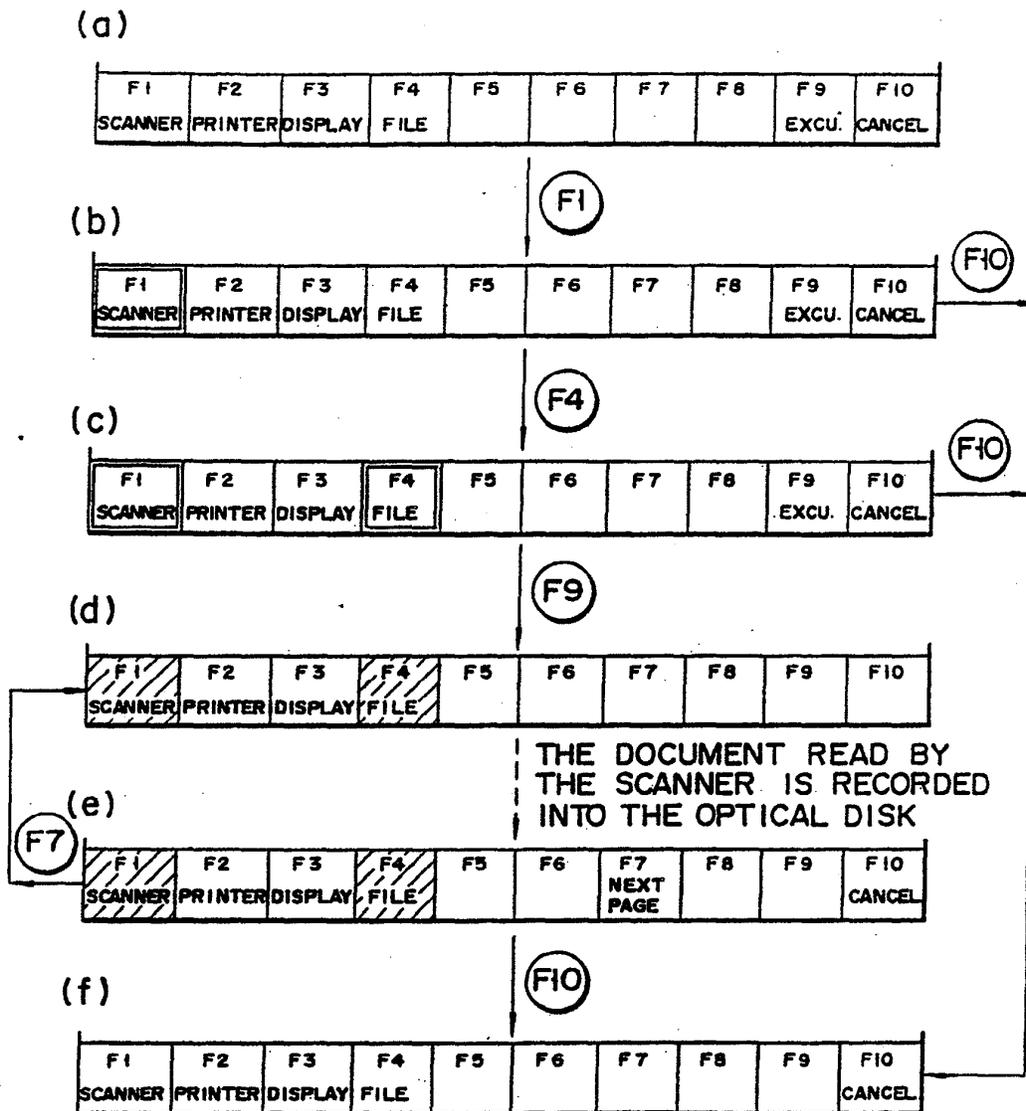


FIG. 21



F I G. 22

IMAGE INFORMATION PROCESSING APPARATUS

FIELD OF THE INVENTION

This invention relates to an image information processing apparatus which is adapted for storing and retrieving image information by using an optical disk memory. More specifically, the invention relates to an image information processing apparatus for executing an image processing operation according to a menu displayed on a display device which includes a plurality of display areas.

BACKGROUND OF THE INVENTION

In recent times, several data processing apparatuses have been developed and put to practical use.

For example, an image information processing apparatus, such as a document filing apparatus, is used for storing and retrieving image information on and from an optical disk memory. One conventional document filing apparatus is disclosed in U.S. Pat. No. 4,604,653. In the document filing apparatus of the type described above, image information on documents is read by a scanner and such image information is stored on the optical disk memory. The apparatus is provided with a display device using a cathode ray tube. The document is scanned by the scanner and the scanned image is transferred to the display device. Corresponding image information obtained from the scanning is stored on the optical disk memory. Desired image information is retrieved as needed and displayed by the display device.

The above-mentioned apparatus can perform a variety of operations such as image retrieval, image recording, image deletion, and others. To operate the apparatus, an operator designates or selects one of the above operations, as desired, whereupon the apparatus performs the desired operation.

For example, when the operator designates "image retrieval", the apparatus then operates in the image retrieval mode and, having retrieved the desired image information from the optical disk memory, displays the image information in the display device, and prints the image information.

Each operation mode of the apparatus is composed of many hierarchy sub-modes, which implies that, in any of the operation modes, a given task can be accomplished only after the sub-modes have been performed. In addition, in the apparatus, a given task can be accomplished only by use of a specific designated operation mode. Therefore, when the apparatus completes every operation, it must return to the initial mode. Consequently, the apparatus cannot continuously operate in different modes, and its operation as a whole is rather inefficient.

To solve the above problem, a related U.S. application, Ser. No. 099,902 filed September 22, 1987, now U.S. Pat. No. 4,885,704 discloses a document filing apparatus capable of executing a plurality of processing operations with a few key operations.

Also, as is well known, a data processing apparatus, such as a personal computer system, having a plurality of display areas has been developed. The above-mentioned system is disclosed in "Window and Window Based Tools: Beginner's Guide" Part No.: 800-1287-03 Revision A of February 17, 1986 pp39-44. In the apparatus, a plurality of display areas are displayed on one screen of a display device. Also, a cursor is displayed on

the screen of the display device and is moved on one of the display areas by means of a pointing device such as a mouse, to designating a desired processing execution. However, the document filing apparatus of the type described above has not been used with a display device having a plurality of display areas.

SUMMARY OF THE INVENTION

Accordingly, it is an object of the present invention to provide an improved image information processing apparatus which has a plurality of display areas for image information.

It is another object of the present invention to provide an image information processing apparatus having a plurality of display areas capable of executing image processing operations, such as image retrieval and image record, with a few key operations.

In accordance with the present invention, the foregoing objects and others are achieved by providing an image information processing apparatus including selecting means for sequential manual selecting at least two from among a plurality of image processing commands including at least a scanning command, a display command and a file command; scanning means for inputting image information from an original image only in response to sequential selection of the scanning command and at least one of the display command and the file command in the selection means, optical memory means for storing the image information input by the scanning means only in response to sequential selection in the selection means of the scanning command followed by at least one of the file command and a sequential combination of the display command and the file command, display means, including at least two display areas, for displaying the image information from one of the scanning means and the optical memory means in only one of the two display areas in response to one of sequential selection of the scanning command and the display command and sequential selection of the file command and the display command, respectively and designation means for manually designating the one of the two display areas to display the image information.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a block diagram of an image information processing apparatus applied to a document filing apparatus with an optical disk memory according to the present invention;

FIG. 2 shows a display device including a single display area on a display screen;

FIGS. 3A, 3B and 3C show display devices, each containing four display areas;

FIG. 4 shows a display device displaying scanner properties;

FIG. 5 shows a display device displaying printer properties;

FIG. 6 shows a display device displaying display properties;

FIG. 7 shows a display device displaying file properties;

FIGS. 8 to 18 diagrammatically show transient phases of various document data transfer modes;

FIG. 19 is a flowchart illustrating a flow of a document data transfer operation;

FIG. 20 shows a designation table and a transfer table;

FIG. 21 is a flowchart illustrating a flow of the document data transfer operation when a continuous document read-out mode is used; and

FIG. 22 shows a transient phase of document data transfer operations, using a function area with a function for starting document data transfer operations.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In an image information processing apparatus shown in FIG. 1, a scanner 1 and a printer 5 are connected to a scanner/printer interface 2. Interface 2 is connected to image data buses (I) and (II), and control buses (I') and (II'). These buses (I) and (II), and (I') and (II') are coupled with an image-enlarging/reducing circuit 6, a character generator 11, and a compander 12. Compander 12, coupled with an optical disk memory 13, appropriately compresses and expands image information recorded and reproduced on and from optical disk memory 13.

Image buses (I) and (II), and control buses (I') and (II') connected through an image bus select controller 3 to memory data buses (I) and (II). Memory data buses (I) and (II), and control bus (I') and control buses (I) and (II) are coupled with an image buffer memory 4, a display memory 7, a display controller 8, a vertical/horizontal conversion circuit 10 and two-dimensional address generators 15 and 16. Generators 15 and 16 are connected through an address (ADRS) bus (I) to image buffer memory 4, and through an address (ADRS) bus (II) to display memory 7. Display memory 7 is coupled with display controller 8 for controlling a display device 9.

Scanner printer interface 2, image-enlarging/reducing circuit 6, display controller 8, vertical/horizontal conversion circuit 10, character generator 11, compander 12, and two-dimensional address generators 15 and 16 are coupled via the CPU bus to a CPU 14. The CPU bus is connected to a CPU memory 17, an interface 18, a mouse 19, and a keyboard 20.

In the apparatus shown in FIG. 1, the image information read out by scanner 1 is loaded into the apparatus through interface 2, and stored via image bus select controller 3 into image buffer memory 4.

The image information stored in image buffer memory 4 is subjected to enlarging or reducing by circuit 6, and stored into display memory 7, and is displayed by display device 9 under control of display controller 8. When it is not necessary to enlarge or reduce the image information, image-enlarging/reducing circuit 6 does not perform its operation. The image information is transferred from display memory 7 to image buffer memory 4 in a similar way.

FIG. 2 shows an initial display on the screen of display device 9 in the image information processing apparatus according to the present invention. As shown in FIG. 2, a display area 201 for displaying image information including characters is substantially centered in the display screen. Icons are arrayed in the right portion of display area 201, and give an operator various necessary indications, such as image-enlarging and reducing, and rotation and scroll of the displayed image information. The icons are designated by mouse 19 or related keys on keyboard 20. For example, if the "Enlarge" icon is designated, the image information displayed within display area 21 is enlarged.

The display further contains a function area 202 provided in connection with function keys F1 to F10.

Function area 202 contains icons F1 to F10 indicating various devices for inputting and outputting image information such as scanner 1, printer 5, display device 9, and optical disk memory 13 (file). The functions of these icons F1 to F10 are:

Scanner (F1): To enter the document data from scanner 1, and transfer it to another device.

Printer (F2): To print the document data transferred from another device, by printer 5.

Display (F3): To display the document data transferred from another device by display device 9, and to store the document data into internal image buffer memory 4. Further, this icon indicates that the document data is read out from internal image memory buffer 4 and transferred to another device.

File (F4): To record the document data transferred from another device on optical disk memory 13, to read out the document data from optical disk memory 13, and to transfer it to another device.

Many processing operations can be performed by appropriately combining the above four devices represented by F1 to F4, and by transferring the document data between devices. For example, if the document data is transferred from scanner 1 (F1) to printer 5 (F2), a hard copy of the document data can be obtained.

The image information processing apparatus according to the present invention can execute ten types of document data transfers, or ten types of document data transfer modes as given below, by appropriately combining the scanner (F1), the printer (F2), the display (F3) and the file (F4).

(1) Scanner (F1)/Display (F3); Mode 1 Enter the document data from scanner 1 and to display it on display device 9.

(2) Scanner (F1)/printer (F2); Mode 2 Enter the document data from scanner 1 and to print it by printer 5.

(3) Scanner (F1)/file (F4); Mode 3 Enter the document data from scanner 1 and to record it on optical disk 13.

(4) File (F4)/display (F3); Mode M4 Retrieve the document data from optical disk 13 and to display it in display area 201 of display device 9.

(5) File (F4)/printer (F2); Mode M5 Retrieve the document data from optical disk 13 and print it by printer 5.

(6) Display (F3)/printer (F2); Mode M6 Print out the document data currently displayed within display device 9 by printer 5.

(7) Display (F3)/file (F4); Mode M7 Record the document data currently displayed within display area 201 on optical disk 13.

(8) Scanner (F1)/display (F3)/printer (F2); Mode M8 Enter the document data from scanner 1, and display it on display device 9, and check and print it by printer 5.

(9) Scanner (F1)/display (F3)/file (F4); Mode 9 Enter the document data from scanner 1, display it on display device 9, and check and record it on optical disk 13.

(10) File (F4)/display (F3)/printer (F2); Mode M10 Retrieve the document data from optical disk 13, display it in display device 9, and check and print it by printer 5.

As seen from the above descriptions, most of the retrieving and recording operations of those filing operations can be made by transferring among those devices such as scanner 1, printer 5, display device 9 and optical disk memory (file) 13.

As shown in FIG. 2, DB function in menu bar area 203 contains a display mode, an edition function and DB function. These are for executing the following processing operations.

Display mode: Changes the number of display areas 201 displayed by display device 9 to one of 1 and 4, as shown in FIGS. 2 and 3.

Editing function: Edits the document image displayed in display area 201, for example, composing, layout, text input, etc.

DB function: To format, open and close optical disk 13, and to arrange all of the documents recorded on optical disk 13 under predetermined conditions, and to save the arranged document data into a file or to delete documents meeting predetermined conditions.

As shown in FIGS. 2 and 3, if the display mode in a menu bar area 203 is designated, one and four display areas 201 can be alternately displayed. Four image buffer memories corresponding to the four display areas are provided in the processing apparatus. Also, as shown in FIG. 3 an indicia 204 represented by a black mark is displayed at the top of the menu bar area of one of four display areas 201. When four display areas 201 are used, the document data transferred from scanner 1 or optical disk 13 is displayed on display area 201a including indicia 204. Also, when four display areas are used, indicia 204 indicates that area 201a is to be recorded on optical disk 13 or printed out by printer 5.

Every time the space key on keyboard 20 is depressed, indicia 204 is shifted from one display area to another display area. When the space key is depressed, indicia 204 is cyclically shifted among those four display areas 201. This allows the operator to display a maximum of four documents in desired display areas 201, and further to record the document data in desired display areas 201 on optical disk 13 or to print the desired document data by printer 5.

In the display of FIG. 3(a), display area 201a in the upper left on the screen is the input/output display area. In the display of FIG. 3(b), display area 201a is in the right upper is the input/output display area. As shown in FIG. 3(c), indicia 204 represented by a black mark can be represented by other marks such as a stripe mark, etc.

Scanner 1, printer 5, display device 9 and optical disk (file) 13 have their own properties affecting image information transfers. Property sheets for changing or checking the properties of these devices can be displayed by display device 9. FIGS. 4 to 7 show displays with these properties. These properties can be displayed by operating the function keys corresponding to function select keys on keyboard 20. The properties can also be displayed by using mouse 19. In this case, a mouse cursor is moved to a desired icon, and the right button (not shown) of mouse 19 is pushed and released from its pushed state (clicked). To erase the property sheet, key F9 or F10 is pushed on keyboard 20. When using mouse 19 for property sheet erasure, the mouse cursor is set to the icon of F9 (confirmation) or F10(erasure), and the left button (not shown) of mouse 19 is clicked.

As shown in FIG. 4, the property sheet of scanner 1 is used for setting properties such as the size of an input document, document density, read-out resolution, and feed mode (book mode) or ADF (automatic document feed) mode). The book mode enters the image information of one sheet of document. The ADF mode successively enters the image information of the documents contained in the ADF holder.

As shown in FIG. 5, the property sheet of printer 5 is used for setting the property of the number of copies to print, for example. As shown in FIG. 6, the display property sheet is used for setting a display time in display area 201. The display time is useful when scanner 1 is operating in the ADF mode. For example, when document data are successively transferred from scanner 1, one sheet of the document is displayed for this display time. When the display time terminates, a sequence of entering the next document begins.

As shown in FIG. 7, in the property sheet of optical disk 9, display device 9 displays information indicating the number of documents currently stored in optical disk 13, and which document data is now read out from optical disk 13. When the document data is read out from optical disk 13, the continuous mode or batch mode is selected. This is similar to the automatic feed mode in the scanner property sheet.

Ten types of document data transfers are described below.

Mode 1

In the initial state, display device 9 displays the image as shown in FIGS. 2 and 3. FIG. 8 shows the operations and the transient phases of changing states of the function area when the document data transfer mode of scanner (F1)/display (F3) is performed. Icon Fx (where x=1 to 10) indicates the operation to push a keyboard function key or the operation to click the left button by setting the mouse cursor to the icon corresponding to the function key within the function area.

In operation, as shown in FIG. 8, for example, icon F1 is designated by the mouse cursor and at this designated position, the button of mouse 19 is clicked (step a). In response to the click action, a marking frame is displayed on icon F1 (step (b)). Then, the mouse cursor is moved to icon F3 and the mouse button is clicked, and the marking frame is displayed on F3 icon (step c). Under this condition, when icon F3 is being indicated, if the mouse button is clicked again, the document data transfer from scanner 1 to display device 9 starts. That is, when CPU 14 detects that icon F3 corresponding to the device finally designated has been designated two times, CPU 14 outputs a signal (command) representing an execution of processing.

At this time, icons F1 and F3 are shaded, in step (d). Then, the document data is entered from scanner 1, and the document image is displayed on display area 201a including indicia 204 shown in FIG. 3. At the same time, the "Next page" is displayed on icon F7 in the function area 202, (step e). Under this condition, when icon F7 is designated by mouse 19 or keyboard 20, the next page of document data is transferred from scanner 1 to display device 9. When F10 icon (cancel) is designated, the document filing apparatus can be returned to the initial state in (step f). Icon F10 can be designated at any time. At steps (b) and (c), the designation of icon F1 or icons F1 and F3 is cancelled. At step (e), the document data transfer from scanner 1 to display device 9 is ended.

In summary, the document data transfer is operated in the following way. In the order of the devices to which the document data are transferred, the icons corresponding to those devices are successively designated one time each. Finally, the icon corresponding to the last device is designated a second time. Then, the document data transfer according to an execution signal from CPU 14. The designations already made can be

cancelled any time by designating the cancel icon F10, before the document transfer starts.

Modes 2 and 3

FIGS. 9 and 10 diagrammatically show the operation modes 2 and 3 of the scanner (F1)/printer (F2) and the scanner (F1)/file (F4). These operations are almost the same as the scanner (F1)/display (F3) operation. The only difference is that the document data input by scanner 1 is not displayed on display device 9, but is printed out by printer 5 or recorded on optical disk 13.

Modes 4 and 5

FIGS. 11 and 12 are diagrammatical representations of the operation modes 4 and 5 of the file (F4)/display (F3) and the file (F4)/printer (F2). In the mode 4 of FIG. 11, the file-to-display document data transfer is executed when icon F4 is designated one time and icon F3 is designated two times. In the mode 5 of FIG. 12, the file-to-print document data transfer is executed when icon F4 is designated one time and icon F2 is designated two times.

In the operations of the above modes 4 and 5 shown in FIGS. 11 and 12, the steps (a) to (b) are the same as those of the previous modes. In step (e), one document is read out from optical disk 13. The read out document data is displayed in display area 201a in the mode 4 of FIG. 11, and is printed out in the mode 5 of FIG. 12. Also in step (e), "Next page" and "Previous page" are displayed on icons F7 and F8. When "Next page" appears, the counter, which indicates which page of the document data being read out, is incremented by one. When "Previous page" appears, the counter is decremented by one. Then the operation returns to step (d), and a similar document transfer is performed. When the cancel icon F10 is designated, the document transfer is ended.

Modes 6 and 7

FIGS. 13 and 14 show the operation modes M6 and M7 of display (F3)/printer (F1) and display (F3)/file (F2). The operations of those modes are similar to the above ones. In step (d), the document data displayed on display area 201a is printed out from printer 5 in the mode M6 of FIG. 13, and is recorded on optical disk 13 in the mode M7 of FIG. 14. The operations of modes 1 to 7 are for document transfer between two devices. Document transfer among three devices is performed as follows:

Mode 8

FIG. 15 shows a sequence of mode 8 operations of scanner (F1)/display (F3)/printer (F2). In this mode, icons F1 and F3 are designated one time each in a successive manner. Finally, icon F2 is designated two times. When CPU 14 detects that icon F2 has been designated two times, the document data is transferred through the route of scanner 1—display device 9—printer 5.

Steps (a) to (e) are for directing the document data transfer. In step (f), the document data is scanned by scanner 1 and displayed in display area 201a including indicia 204. At this point, the document data has been transferred between scanner 1 and display device 9. Also in step (f), "Print" and "Reenter" are displayed on the icons F7 and F8.

Under this condition, the operator checks the document data in display area 201a. If the operator decides

that the density of scanner 1 is improper, he selects the scan property sheet shown in FIG. 4 and alters its properties and designates "Reenter" icon F8. Upon this action by the operator, scanner 1 rescans the document using the properties from the altered property sheet and displays the document data in display area 201a including indicia 204. If the operator decides that the document data in display area 201a is satisfactory, the print icon F7 is designated, so that the document data in display area 201a is transferred between display device 9 and printer 5 (step g). At this time, the icon F7 displays "Next page".

When the icon F7 of displaying "Next page" is designated, the operation returns to step (e), and the mode 8 for the scanner 1—display device 9—printer 5 is re-executed from the first step. When icon F10 of "Cancel" is designated, the document data transfer operations is stopped. The operation of displaying the property sheets of the printer, displaying and filing, and checking and changing the properties may be performed at any time.

Modes 9 and 10

FIGS. 16 and 17 show a sequence of document data transfers operations of each of modes 9 and 10. Mode 9 transfers document data among scanner 1 (F1), display device 9 (F3), and optical disk 13 (file) (F4). Mode 10 transfers document data among optical disk 13 (file) (F4), display device 9 (F3) and printer 5 (F2).

In the operation modes, as mentioned above, the properties of scanner 1 and the file each indicates the book mode and the successive read-out mode. Every time one document is transferred, the step to check the displayed document data is needed. This step corresponds to the steps (f) and (g) in FIG. 16.

The operation of the document filing apparatus when the property of scanner 1 is the ADF mode, will now be described. The batch read-out mode operates similarly. FIG. 18 diagrammatically shows a sequence of document data transfer among scanner 1 (F1), display device 9 (F3) and optical disk 13 (file) (F4) when scanner 1 is in the ADF mode. In this operation sequence, the steps (a) to (e) are for directing the document data transfer. This mode is different from the above-mentioned one in that in step (e) "Stop" is displayed on icon F6. Unless the "Stop" icon is designated in step (e), scanner 1 scans and reads out the document data of all of the documents contained in the ADF holder of scanner 1. Each of these documents is displayed for a predetermined period of time on display area 201a, and then these document data are successively stored into optical disk 13.

The display time of a document on display area 201a is set by the display property. When an operator designates the icon "Stop" in step (e), the document data transfer is temporarily stopped, and function area 202 shows the display shown in step (g). Under this condition, if "Reenter" (F8) is designated, scanner 1 reads out the document data.

When the stop of document transfer is removed and the document data transfer is started again, icon (F6) of "Continue" is designated, and the operation returns to step (e). When the document data transfer of the ADF holder contained documents is completed, or when the icon "Cancel" (F10) is designated, the document data transfer is completed and operation passes to step (f).

As described above, when using the image information filing system according to the present invention, the directions for retrieval, recording, and the like can

be executed by merely designating three to four icons representing devices.

The operations of the FIG. 1 circuit relating to the document data transfer operations as illustrated in FIGS. 8 to 18, will be described referring to a flowchart of FIG. 19. Tables to be used in the description of the operations are illustrated in FIG. 20. A designation table (a) holds the number of icons designated by the operator, and the order in which the icons were designated. A transfer table (b) holds valid combinations of icons for the document data transfer that can be displayed by the image information processing apparatus, and the order of function numbers. These tables are referred to during the document data transfer processings. These tables (a) and (b) are stored in CPU memory 17.

In operation, when the power supply is turned on, the initial states at the respective portions of the image information processing apparatus are set up according to a program stored in CPU memory 17. At this time, display device 9 displays display area 202, function area 201, and menu bar area 203, as shown in FIG. 2. This display pattern data is the data as transferred from the CPU memory 17 or an external floppy disk to display memory 7. Under this condition, the data corresponding to any of icons F1 to F10 is designated and entered by means of the keyboard or the mouse 19 (S1). The entered data is checked by CPU 14 to determine if it coincides with one of icons F1 to F4 (S2). If the answer is YES, CPU 14 checks if the function number corresponding to the input or entered data coincides with the last function number in the designation table (a) of FIG. 20 (S3). If NO, CPU 14 checks if the function numbers of three or more have been recorded on the designation table (S4). If the answer is No, the number of icons in the designation table is increased by one, and the function number as designated and entered is written into the designation table (S5). The marking frame is displayed on the icon corresponding to the function number recorded anew (S5). After this, the operation returns to step S1.

If more than three function numbers are stored in the designation table, i.e., if the answer in step S4 is YES, the designation table (a) is cleared, and the marking frame of icon is erased (S6). If the designated and entered function number coincides with the last function number in the designation table (S3), the order (e.g., F1-F3) of the function numbers in the designation table (a) is compared with each of the orders (F1-F3, F1-F2, F1-F4, . . .) of function numbers in the transfer table (b) (S7). If the same order is found, the marking frame in the icon is erased (S8) and the document data transfer processing is started in the order shown in the designation table (a) (S9). For example, when the function numbers are stored in the designation table in the order of F1-F3, the document is read out by scanner 1, and the document data is transferred to image data bus (I) through scanner printer interface 2. The document data in display memory 7 is read out by display controller 8, input to display device 9, and displayed as a document image. Thus, the document data transfer from scanner 1 to display device 9 is automatically performed.

Upon completion of the document data transfer, the designation table (a) is cleared (S10) and operation returns to step S1. If the order in the designation table (a) is not registered in the transfer table (b) (S4), the designation table (a) is cleared and the marking frame of the icon is erased (S6), and the operation returns to step

S1. If the designated and entered data is F10 (S11), the designation table (a) is cleared, and the marking frame of the icon is erased (S6), and the operation returns to step S1.

If the designation table (a) stores the function numbers in the order of F1-F3-F4 corresponding to the document data transfer operation of FIG. 16, the document is read out by scanner 1, and the document data is transferred to image data bus (II) via scanner printer interface 2. The document data on image data bus (II) is transferred through image bus select controller 3 to image buffer memory 4 and display memory 7. The document data in display memory 7 is read out by display controller 8, and input to display device 9, and displayed in the form of a document image. The document data in image buffer memory 4 is read out, and input to compandar 12 via image bus select controller 3 and the image data bus. The document data is subjected to the companding processing in compander 12, and is input to and stored in optical disk 13 as the file. In this way, the document data 1 is transferred to display device 9 and optical disk 13 from scanner 1.

FIG. 21 is a flowchart showing the operation of FIG. 18 in which the mode of the first device in the document data transfer contains the ADF mode for automatically feeding documents.

When the document data transfer processing begins, CPU 14 checks the read-out mode of the first device (the first device in the designation table (a)), for example, scanner 1, in the document data transfer. When this mode is the batch read-out mode such as the ADF mode, the following operation procedure will be performed. CPU 14 checks if the "Stop" icon is designated (S101). If the answer is NO, the CPU checks whether the document to be transferred to scanner 1 is present or not (S102). If YES, a sequence of document data transfer processings are executed without interruption (S103). For example, when the designation for scanner 1, display device 9 and printer 5 is made, the document data enter, the document display, and the document print are successively executed in connection with the first sequence of document data transfer processings.

After completion of the first sequence of the document data transfer processings, CPU 14 returns to step S101 to check the designation and entering of the "Stop". The second and subsequent sequences of document data transfer processings will be continued uninterruptedly until the "Stop" is designated or no documents remain to transfer. As when no documents remain to transfer, the document data transfer processing ends.

When "Stop" is designated (S101), the document data transfer operation is stopped until the next designation and entering of data. When the "Continue" is designated (S104) CPU 14 returns to check whether or not the document to be transferred is contained in the first device, and enters the control loop for the sequence of document data transfer processings.

In step S104 for designation and entering of data, the function select and F1 to F4 are designated and entered, the property sheets of the devices corresponding to F1 to F4 are displayed and the properties of these devices are checked and if necessary, changed (S105). The control returns to step S104.

In the above-mentioned embodiment, at the start of document data transfer, the icon corresponding to the finally designated device is designated two times. An alternative measure to start the document data transfer

is illustrated in FIG. 22. In FIG. 22, Icon F9 is designated by a mark cursor or by related corresponding function keys on the keyboard. More specifically, as shown in FIG. 22, to start the execution of the mode M3 of the scanner-file document transfer, the icons are designated in the order of F1 (scanner), F4 (file) and F9 (execution). After the start of data transfer processing, the word "Execution" disappears from icon F9.

As described above, the document data transfer processing can be repeatedly executed uninterruptedly by setting the first device in the continuous read-out mode.

As described above, the number of steps for many operations, for example, document data retrieval, is reduced, and the operation shifts can be made smoothly.

What is claimed is:

1. An image information processing apparatus comprising:

selecting means for sequentially manually selecting at least two image processing commands from among a plurality of image processing commands including a scanning command, a display command and a file command;

scanning means for inputting image information from an original image only in response to sequential selection of the scanning command and at least one of the display command and the file command in the selecting means;

optical memory means for storing the image information input by the scanning means only in response to sequential selection in the selecting means of the scanning command followed by at least one of the file command a sequential combination of the display command and the file command;

display means, including a display screen having at least two display areas, for displaying the image information from one of the scanning means and the optical memory means in response to one of sequential selection of the scanning command and the display command and sequential selection of the file command and the display command, respectively; and

designation means for manually designating the one of at least two display areas to display the image information from the one of the scanning means and the optical memory means in only one of at least two display areas.

2. The apparatus of claim 1, wherein the selection means also includes means for selecting a print command from among the plurality of image processing commands, and printing means for printing the image information from one of the scanning means, the display means and the memory means in response to sequential selection in the selection means of one of the scanning command, the display command, and the file command, followed by the printing command.

3. The apparatus of claim 2, wherein the printing means also includes means for printing the image information from one of the scanning means and the memory means in response to sequential selection of one of the scanning command and the file command, respectively, followed by the display command and the print command.

4. The apparatus of claim 1, further comprising buffer memory means for storing the image information inputted by the scanning means.

5. The apparatus of claim 1, wherein the designation means includes an indicia indicating a current display area for displaying the image information and shifting

means for cyclically shifting the indicia from one display area to another display area.

6. An image information processing apparatus comprising:

selecting means for sequentially manually selecting at least two image processing commands from among a plurality of image processing commands including a file command, a display command and a printer command;

optical memory means for storing the image information;

display means, including a display screen having at least two display areas, for displaying the image information from the optical memory means in response to sequential selection of the file command and the display command and sequential selection of the file command followed by at least one of the display command and a sequential combination of the display command and the printer command;

printing means for printing the image information from one of the optical memory means and the display means in response to sequential selection in the selection means of the file command and the printing command and sequential selection in the selection means of the file command, the display command and the printing command; and

designation means for manually designating the one of at least two display areas to display the image information from the optical memory means in only one of at least two display areas or to print the image information displayed on the one of at least two display areas by the printing means.

7. The apparatus of claim 6, further comprising buffer memory means for storing the image information from the optical memory means.

8. The apparatus of claim 6, wherein the designation means includes an indicia indicating a current display area for displaying the image information from the optical memory means or for displaying the image information to be printed and shifting means for cyclically shifting the indicia from one display area to another display area.

9. An image information processing apparatus comprising:

selecting means for sequentially manually selecting two image processing commands from among a plurality of image processing commands including a display command and a file command;

display means, including a display screen having at least two display areas, for displaying the image information;

optical memory means for storing the image information displayed in only one of the display areas in response to sequential selection in the selection means of the display command and the file command; and

designation means for manually designating the one of at least two display areas to store the image information displayed in one of at least two display areas in the optical memory means.

10. The apparatus of claim 9, further comprising buffer memory means for storing the image information displayed in the display means.

11. The apparatus of claim 9, wherein the designation means includes an indicia indicating a current display area from which to store the displayed image information into the optical memory means and shifting means for cyclically shifting the indicia from one display area to another display area.

* * * * *



US005091866A

United States Patent [19]

[11] Patent Number: 5,091,866

Takagi

[45] Date of Patent: Feb. 25, 1992

[54] INFORMATION PROCESSING APPARATUS
DISPLAYING MULTIPLE WINDOWS AND
DISTINGUISHING INDICIA

[75] Inventor: Shiro Takagi, Yokohama, Japan

[73] Assignee: Kabushiki Kaisha Toshiba, Kawasaki,
Japan

[21] Appl. No.: 489,724

[22] Filed: Mar. 2, 1990

Related U.S. Application Data

[63] Continuation of Ser. No. 136,391, Dec. 22, 1987, abandoned.

Foreign Application Priority Data

Dec. 26, 1986 [JP] Japan 61-315332

[51] Int. Cl.⁵ G06F 15/62

[52] U.S. Cl. 395/158; 395/157

[58] Field of Search 364/518, 521; 340/707,
340/710, 716, 721, 723; 382/44-48

References Cited

U.S. PATENT DOCUMENTS

4,651,146	3/1987	Lucash et al.	340/721
4,653,020	3/1987	Cheselka et al.	364/900
4,661,810	4/1987	Himmelstein et al.	340/710 X
4,683,468	7/1987	Himmelstein et al.	340/710 X
4,710,767	12/1987	Sciacero et al.	340/799
4,769,762	9/1988	Tsujido	340/716 X

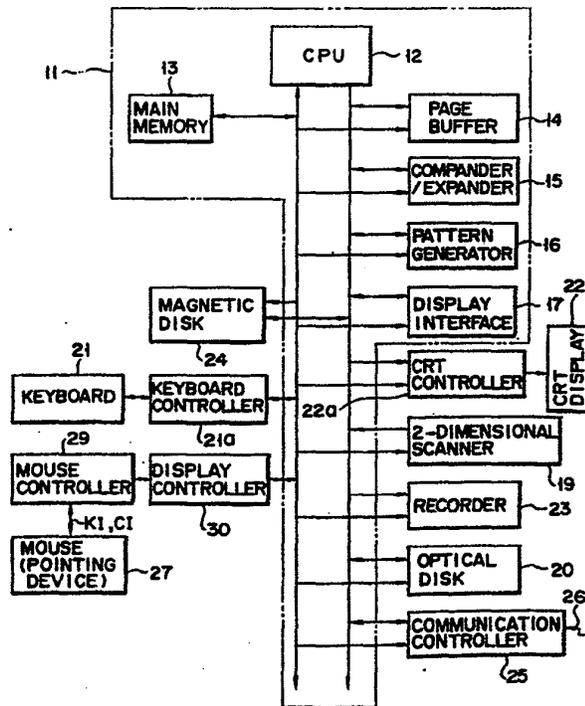
4,772,882	9/1988	Mical	340/710 X
4,806,919	2/1989	Nakayama et al.	340/721
4,819,189	4/1989	Kikuchi et al.	364/521
4,829,294	5/1989	Iwami et al.	364/521 X
4,833,596	5/1989	Buckland et al.	364/521 X
4,890,098	12/1989	Dawes et al.	340/721
4,890,257	12/1989	Anthias et al.	364/900

Primary Examiner—Heather R. Herndon
Attorney, Agent, or Firm—Finnegan, Henderson,
Farabow, Garrett, and Dunner

[37] ABSTRACT

A display unit displays data to be processed. The display unit is capable of displaying a plurality of windows for displaying the data therein. A controller outputs the control data to display the plurality of windows and an indicia on the display unit. An operating unit supplies the controller with a position signal to move the indicia to a desired location. An area unit checks whether the indicia is located a predetermined window area of the windows being displayed on the display unit, any data displayed in the predetermined window area being processed prior to the data displayed in any other window area. A specifying unit supplies signals to the controller, the signals specifying, in accordance with the check results made by said area check unit, a condition in which the indicia is displayed to demonstrate whether the indicia is being displayed in the predetermined window area or in the other window areas.

11 Claims, 8 Drawing Sheets



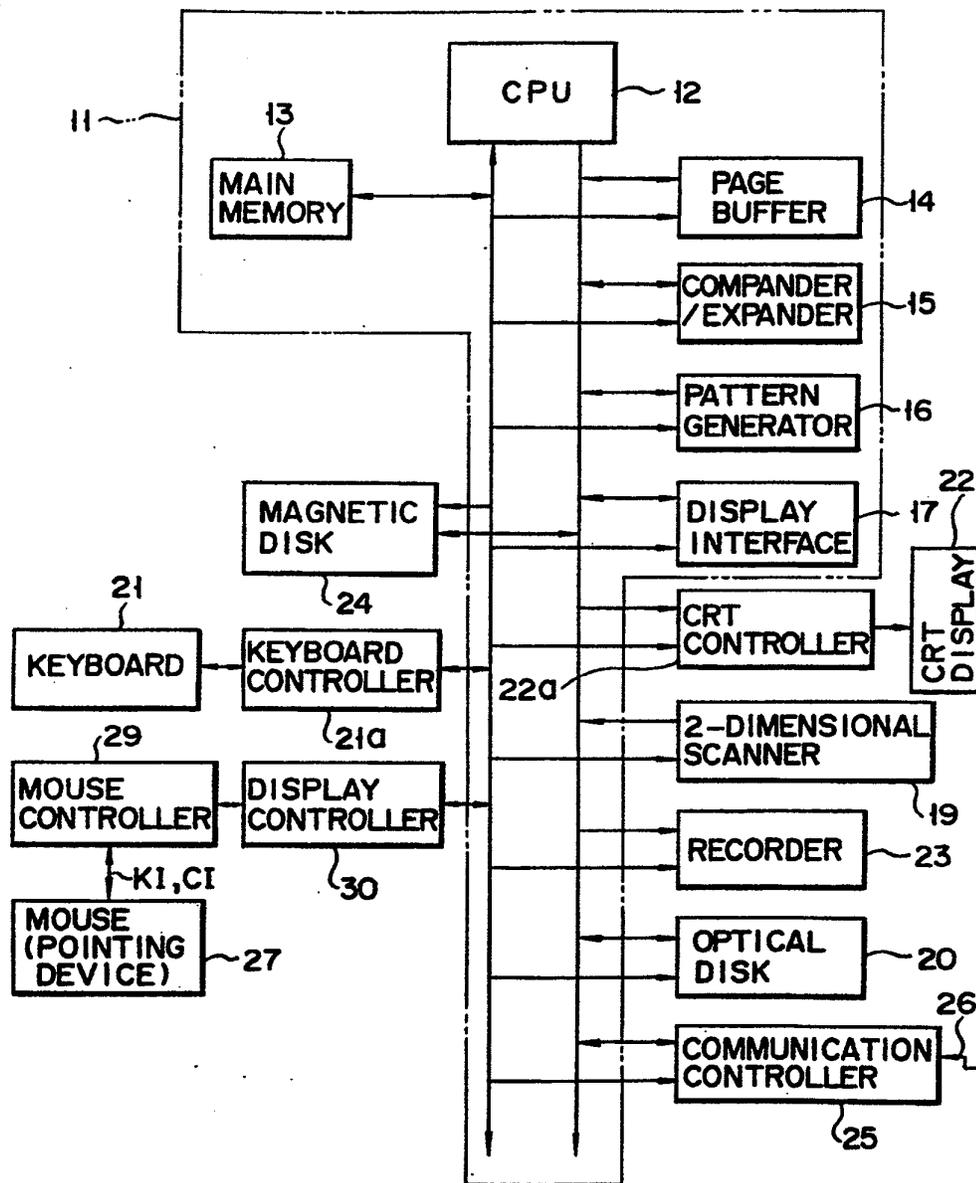


FIG. 1

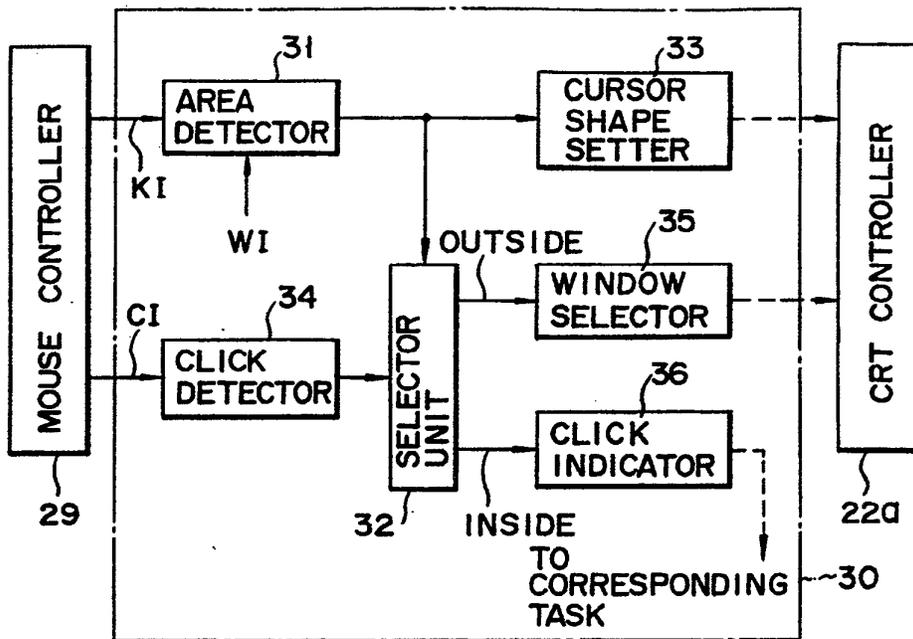


FIG. 2

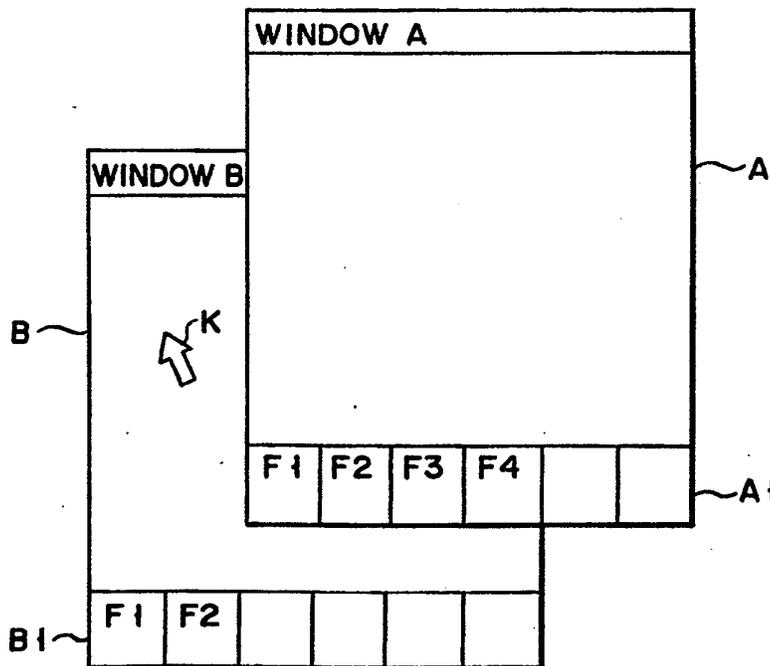


FIG. 3

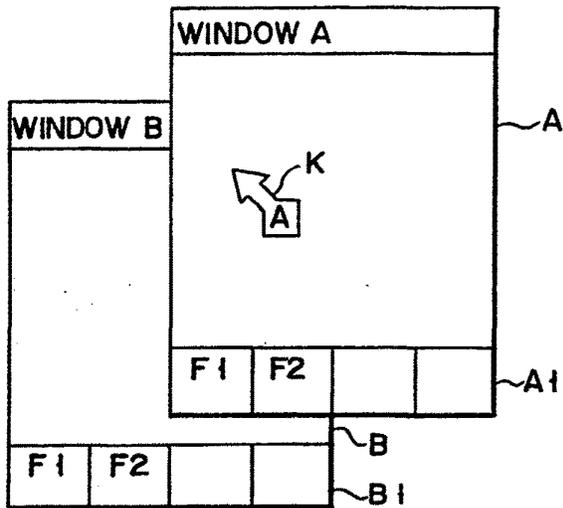


FIG. 4A

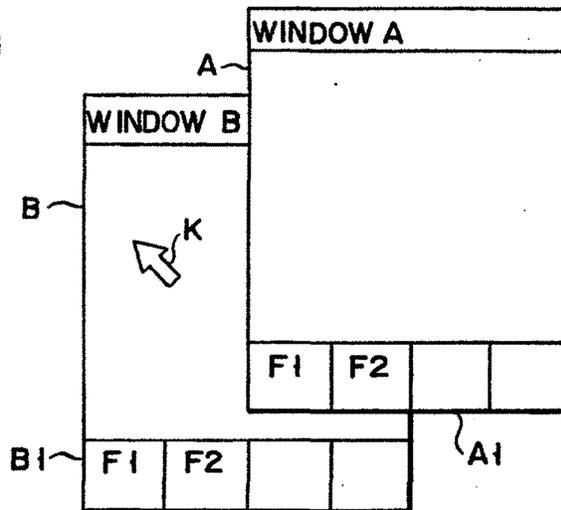


FIG. 4B

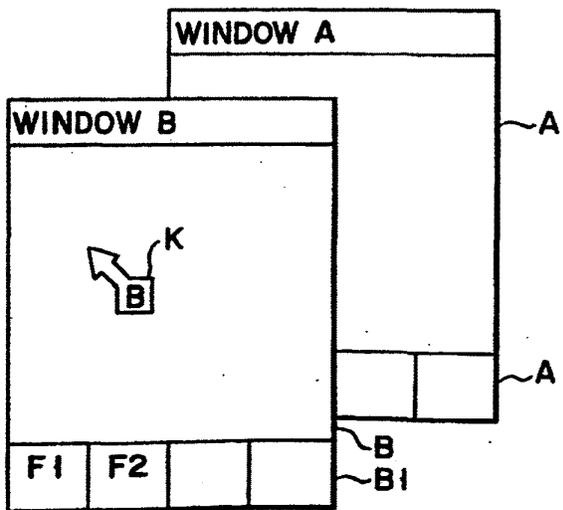


FIG. 4C

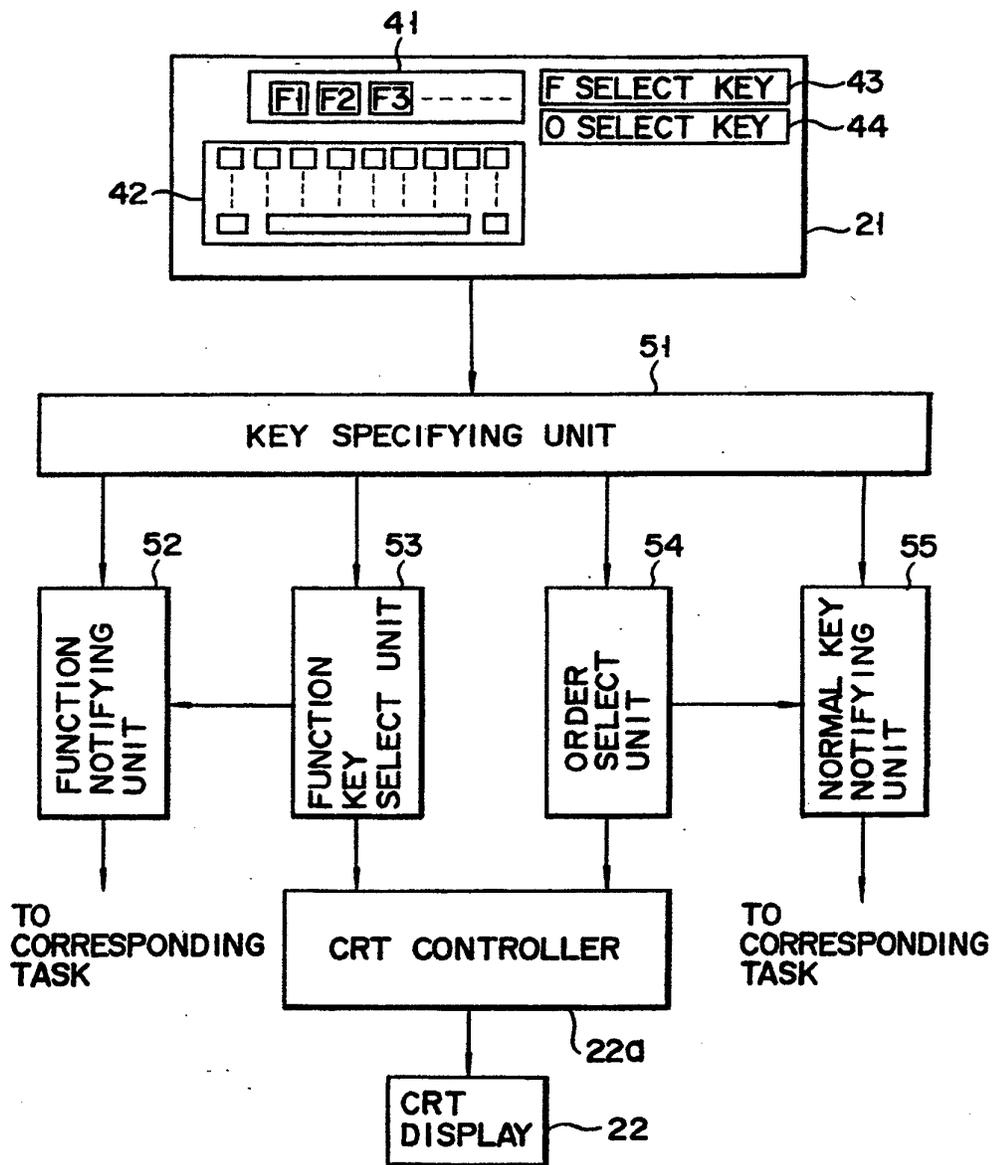


FIG. 5

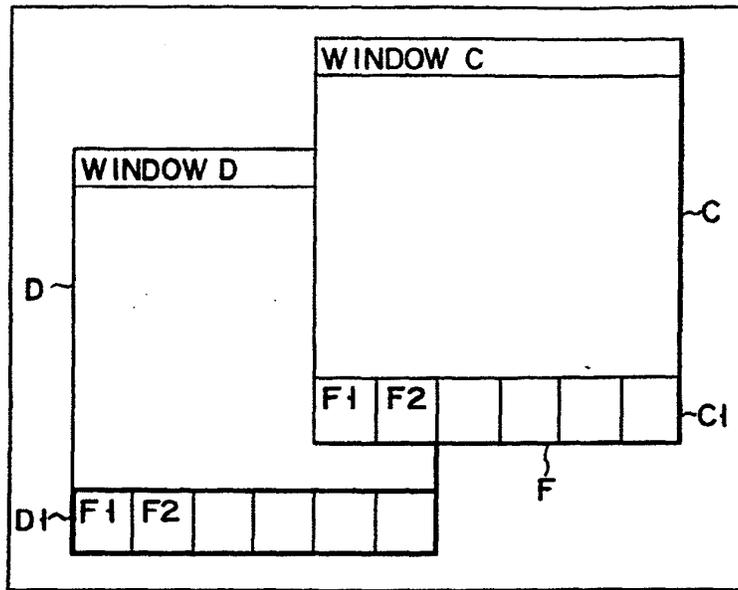


FIG. 6A

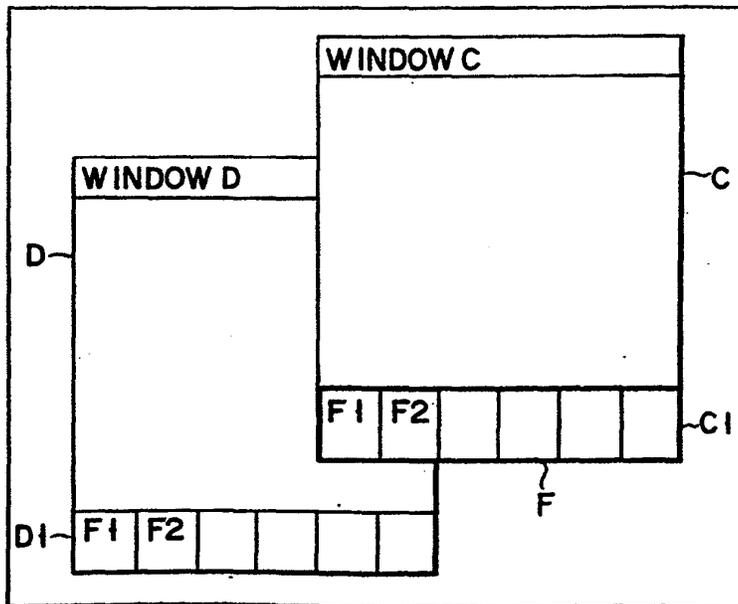


FIG. 6B

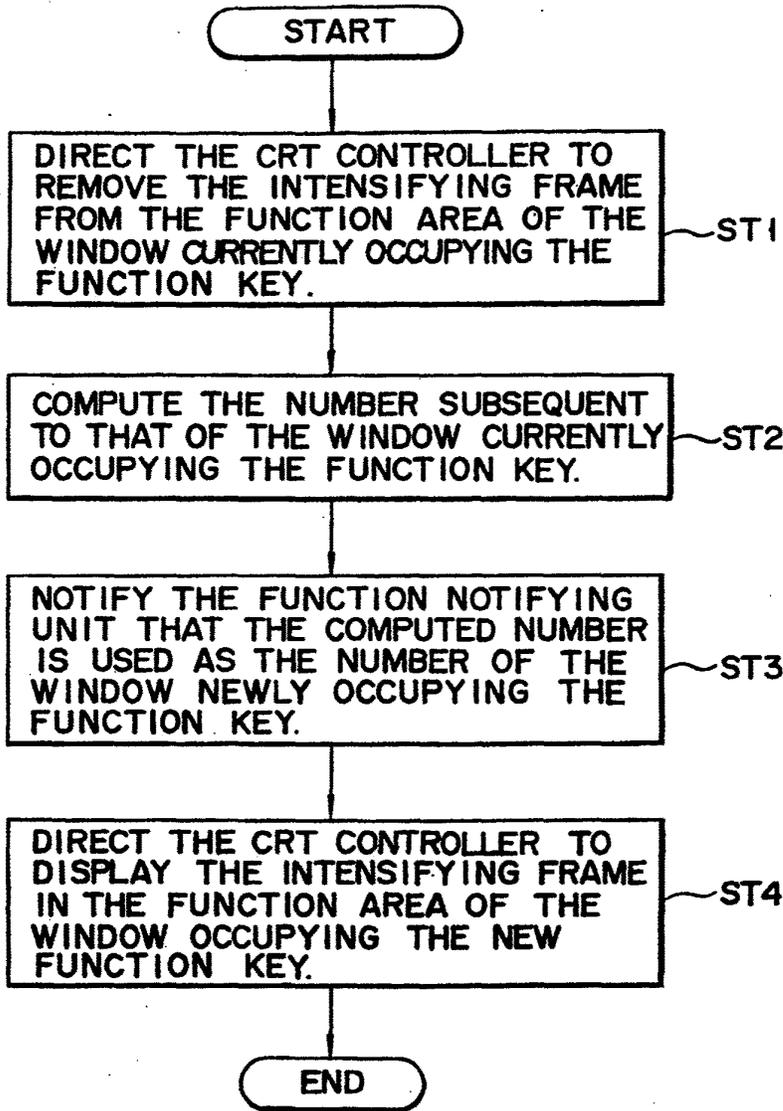


FIG. 7

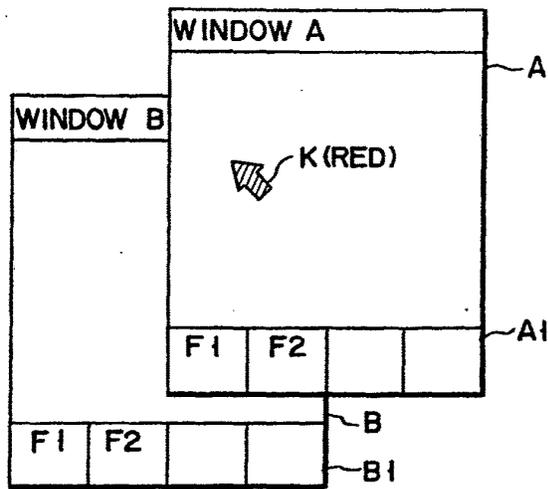


FIG. 8A

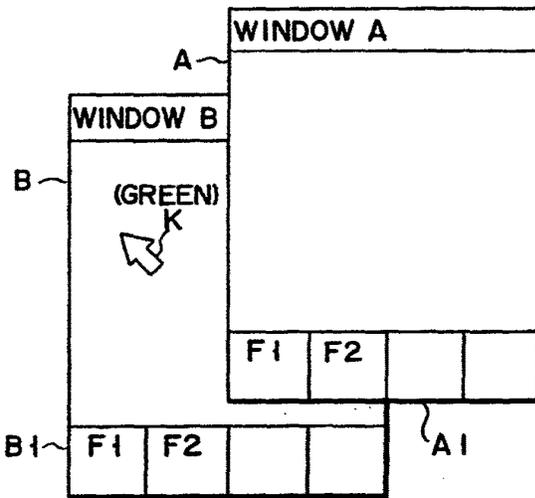


FIG. 8B

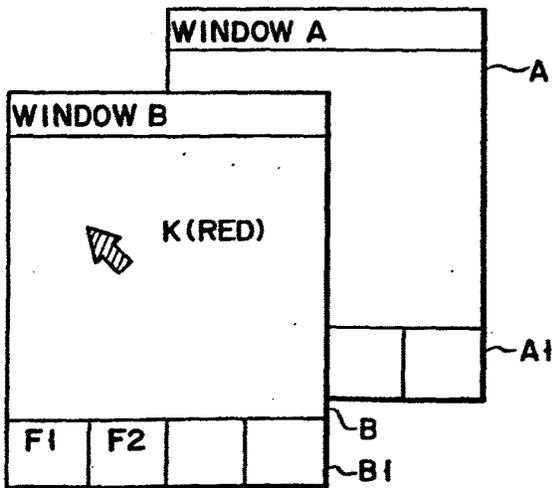


FIG. 8C

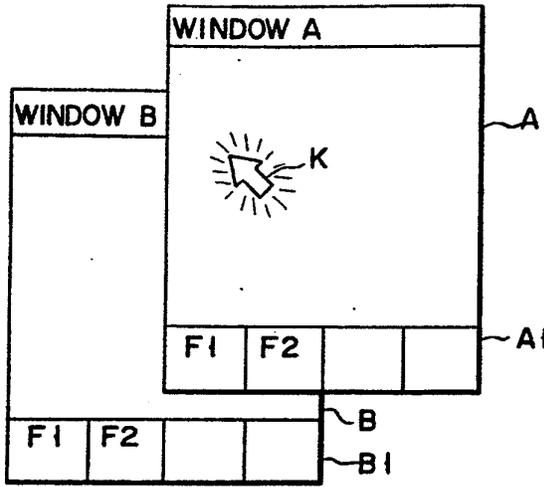


FIG. 9A

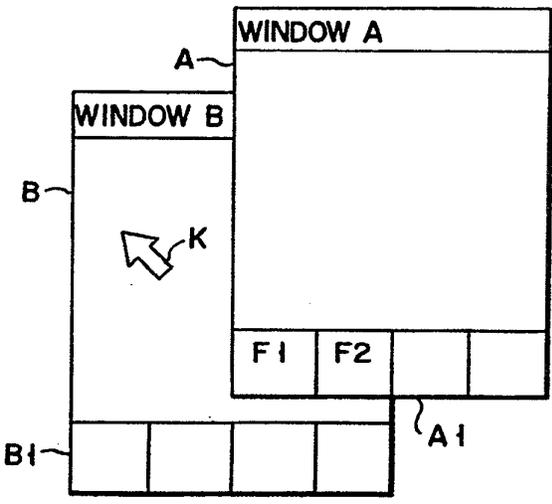


FIG. 9B

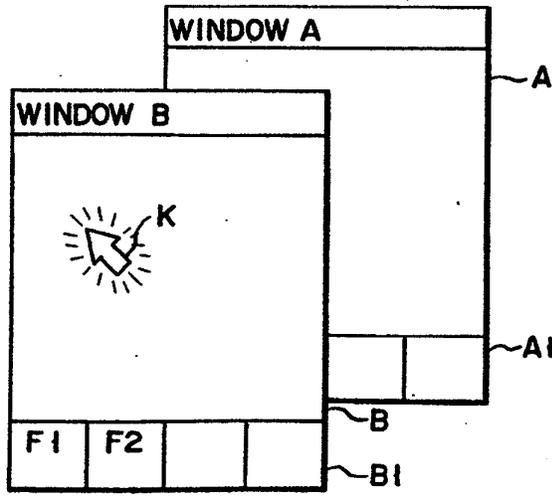


FIG. 9C

INFORMATION PROCESSING APPARATUS DISPLAYING MULTIPLE WINDOWS AND DISTINGUISHING INDICIA

This application is a continuation, of application Ser. No. 07/136,391, filed Dec. 22, 1987, now abandoned.

BACKGROUND OF THE INVENTION

This invention relates to an information processing apparatus and, more particularly, to a multi-window display controlling device of a data processor for controlling various types of operations by means of a cursor, which is moved by a pointer device such as a mouse.

Information processing apparatuses having a multi-task/multi-window system are well-known. In such an apparatus, a cursor is displayed on a CRT screen, for example, and is moved to a predetermined location on the screen by means of a pointing device such as a mouse, to designate a desired processing operation.

In the information processing apparatus, if the cursor is moved to a specific area on the CRT screen, the shape of the cursor is changed to show the operator the command that will be executed when the mouse is operated. For example, when a line is typed and echoed onto the screen, the cursor is shaped like a "pen." To move a displayed object, the cursor is shaped like a "pincette."

Indicating a command by an exact shape at the specific cursor location makes it easy for an operator to understand the operation performed by the command.

In such a multi-task/multi-window apparatus, a plurality of windows are displayed on one screen of the CRT display and tasks corresponding to those windows are executed concurrently. If the cursor shape is changed every time the cursor is moved to a specific location in each window in such a multi-task/multi-window data processing apparatus, the operator is often confused.

In this type of data processing apparatus, windows frequently partially overlap. If the cursor shape is changed when the cursor is below an overlapping window, the changed shape of the cursor does not present any effective information to the operator because the cursor is in the limited part of the window not being displayed. Additionally, every time the cursor is moved on the CRT screen, the system must determine the position of the cursor relative to the various to the various window regions resulting in increased response time for the system.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a new and improved information processing apparatus in which the operation is smooth even when an indicia, such as the cursor, is moved on the multi-window display, and the response time of the indicator is reduced.

According to the present invention, there is provided an information processing apparatus comprising:

means for displaying a plurality of window areas, each of which includes information to be processed, wherein a window area at least partially overlapped and hidden by others of the plurality of windows defines a hidden window area,

means for determining a window area which is not a hidden window area, and

means for displaying a distinguishing indicia in the window area determined not to be a hidden window area.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects and features of the present invention can be understood through the following embodiment by reference to the accompanying drawings, in which:

FIG. 1 shows a block diagram of an information processing apparatus according to a first preferred embodiment of this invention;

FIG. 2 shows a block diagram of a display control section in the FIG. 1 circuit;

FIG. 3 shows an example of display where two windows are displayed partially overlapping;

FIGS. 4A to 4C show an example of a display further explaining the display operation of FIG. 3;

FIG. 5 shows a diagram illustrating a keyboard and its related circuit;

FIGS. 6A and 6B show displays explaining the operation of the circuit of FIG. 5;

FIG. 7 shows a flow chart further explaining the operation of the circuit of FIG. 5; and

FIGS. 8A to 8C and 9A to 9C show other examples of displays.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A preferred embodiment of this invention will be described referring to the accompanying drawings.

FIG. 1 shows the construction of an information processing apparatus. Main controller 11 comprises CPU 12, main memory 13, page buffer 14, compander/expander 15, pattern generator 16, and display interface 17, as shown in FIG. 1. CPU 12 is operable in a multi-task mode for executing various types of operations. Main memory 13 stores a control program, for example. Page buffer 14 has a memory capacity of 512 kilo bytes (=128 blocks), for example. Compander/expander 15 compresses and expands the image data. As is known, data compression reduces data redundancy and data expansion restores the original redundancy of the data. Pattern generator 16 stores the data in patterns comprised of characters and symbols.

Two dimensional scanner 19 scans an image on a document by laser beams, for example, and produces electrical signals representing the image data read in. Optical disk 20 sequentially stores the image data, which is supplied from two dimensional scanner 19 through main controller 11.

Keyboard 21 is used for entering the retrieval data relating to the image data and various types of operation commands. The output signal of keyboard 21 is supplied to CPU 12 via keyboard controller 21a.

A cathode ray tube display device 22 (referred to as a CRT display) displays the image data, which is collected by two dimensional scanning device 19 and supplied to CRT display 22 through main controller 11. CRT display 22 also displays the image data, which is read out from optical disk 20 and supplied to CRT display 22 through main controller 11. CRT display device 22 is controlled by CRT controller 22a. CRT controller 22a controls the display of multi-windows and the display of the cursor moved by a pointing device.

Recorder 23 is a printer, for example, and prints the image data, which is collected by two dimensional scan-

ning device 19 and supplied thereto through main controller 11, or the image data, which is read out from optical disk 20 and supplied thereto through main controller 11.

Magnetic disk 24 stores a table of titles (also called retrieval data), which are respectively assigned to the image data.

Communication controller 25 is coupled with an external device via communication line 26, and transmits an image retrieval request signal and the retrieval data, for example, and receives an image retrieval request signal and the retrieval data, and supplies them to main controller 11.

Mouse 27 is a kind of pointing device. Mouse 27 is connected to mouse controller 29. Mouse controller 29 outputs a position signal KI as an indicia of CRT display 22, and operation signal CI indicative of an operation state of click button 28. These signals output from mouse controller 28 are supplied to display controller 30.

FIG. 2 shows a block diagram of display controller 30. The cursor position signal KI output from mouse controller 29, together with a window position signal WI supplied from CPU 12, is supplied to area detector 31. This area detector 31 detects the cursor movement into or out of the area of the window currently located at the frontmost position on the screen (referred to as the top), on the basis of the input cursor position signal KI and the window position signal WI. This detected output signal is supplied to selector unit 32 and cursor shape setter 33. Cursor shape setter 33 sets the shape of cursor to a standard shape when the detected output signal from area detector 31 represents cursor movement out of the window. When the cursor moves into the window, cursor shape setter 33 sets the unique or distinguishing cursor shape corresponding to that of the window currently located at the top, or frontmost position. The output signal of cursor shape setter 33 is supplied to CRT controller 22a. Operation signal CI output from the mouse controller 9 is supplied to click detector 34. Click detector 34 detects a transient state of the input operation signal when it changes from an off-state to an on-state. The detected signal is supplied to selector 32. Selector 32 sends a signal to window selector 35 when the cursor is located outside the top window. In response to this signal, window selector 35 outputs a select request signal to CRT controller 22a. The select request signal requests the CRT controller to move the window currently indicated by the cursor to the top. If selector 32 decides that the cursor is present in the top window, click indicator 36 notifies the task corresponding to the top window of the cursor position data.

The operation of the data processing apparatus thus arranged will now be described.

FIG. 3 shows an example of display by CRT display 22. As shown, CRT display 22 displays two windows A and B overlapping with each other and also displays cursor K. These windows A and B respectively contain function areas A1 and B1 each containing commands corresponding to the function keys. In this display, some items of image data are retrieved from optical disk device 20, using a function key in the window A, and the retrieved data is displayed in window B. In this case, to sequentially display a plurality of retrieved image data in window B, function key 1 or 2 is operated for window A. Function key 1 directs the display of the image data next to the image data currently displayed in window B. Function key 2 directs the display of the

previous image data. When function key 1 is operated in window A, the enlargement processing of the image data displayed in window A is executed.

As described above, when cursor K is located in the area of windows other than the top window, the shape of cursor K is set to the standard shape. Under this condition, if mouse 27 is operated and cursor K is moved to the area of the top window A, cursor K is changed to the shape in window A as shown in FIG. 4(A). As shown, the cursor contains character "A". Under this condition, mouse 27 is operated, and cursor K is moved from the area of window A to that of window B. The shape of cursor K is changed to the standard shape again. In this display state, if click button 28 of mouse 27 is operated, windows A and B are interchanged, as shown in FIG. 4(c), with window B becoming the top window. Under this display state, if cursor K is in the area of window B, cursor K is changed into the shape of window B.

As shown in FIGS. 4A and 4C, when cursor K is moved to the top window A or window B, the click button 28 of mouse 27 is operated and the position data of cursor K is sent to the task corresponding to the top window, and a predetermined process is executed.

According to this embodiment, when the cursor is in the top window, its shape is changed to the shape in the top window. When the cursor is moved to other windows than the top window, the shape of cursor is changed into the standard shape. Therefore, the operator can easily understand to which window the cursor corresponds, and the operator's recognition tasks are made easier.

In moving the cursor it is necessary to check only whether the cursor is in the top window. This fact leads to the improvement of the processing speed and the response speed of the cursor.

When the cursor is moved to any windows other than the top window, and the click button is operated, the top window is changed. This feature makes the data processing apparatus easier to operate.

Keyboard controller 21a will now be described. As seen from FIG. 5, keyboard 21 contains function keys 41, normal keys 42, function select key 43, and order select key 44. Normal keys 42 consist of hiragana (Japanese phonetic symbols) keys, alphabet keys and the like. Function select key 43 selects either of function keys 41 or normal keys 42. Order select key 44 selects the order of windows.

Keyboard controller 21a contains a key specifying unit 51. By using the key code input from keyboard 21, this unit specifies function keys 41, function select keys 43, order select key 44, or normal key 42. When key specifying unit 51 specifies the function key by the input key code, the input key code is input to function notifying unit 52. When the key specifying unit 51 specifies the function select key, the input key code is input to function key select unit 53. When the input key code specifies the order select key 44, the input key code is input to order select unit 54. When the key code is for the normal keys 42, it is input to normal key notifying unit 55.

Function notifying unit 52 transfers the input function key code to the task operating in the window having the ownership of the function key.

Function key select unit 53 switches the window having the function ownership from the current window to another window, and notifies the function notifying unit 52 and CRT controller 22a of the window

having a new function key ownership. CRT controller 22a controls the CRT display so that the window having the function key is displayed in an intensified manner.

Order select unit 54 is used for selecting a desired order of the window, and notifies normal key notifying unit 55 and CRT controller 22a, of the new top window. CRT controller 22a updates the screen of CRT display 22 according to the signal from order select unit 54.

Normal key notifying unit 55 supplies a normal key code to the task operating in the top window, according to the signal supplied from order select unit 54.

FIGS. 6A and 6B show display examples of CRT display 22, respectively. As shown, windows C and D are displayed overlapping with each other. Function areas C1 and D1 are contained in each of windows C and D. Names of commands, which are executed by each of the function keys are displayed in function areas C1 and D1. For example, when function key 1 is operated in window C, the enlargement of the image being displayed in window C is executed. Intensifying frame F is displayed in the function area of the window currently having the function key ownership. When the operator sees the intensifying frame F, the operator knows what command is to be executed by operating the function key 31 on keyboard 21.

FIG. 7 shows a flowchart for explaining the operation of function key select unit 53. Function select key 43 is operated, and a function select key code from key specifying unit 51 is input to function key select unit 53. Upon receipt of this function key code, function key select unit 53 directs CRT controller 22a to remove the function area intensifying frame of the window currently having the ownership of the function key. This is done in step ST1. In step ST2, a window number different from that of the window currently having the function key ownership is computed. Subsequently, in step ST3, the window bearing the computed number is transferred as a new window having the function key ownership, to function key notifying unit 52. In step ST4, control directs CRT controller 22a to display the intensifying frame in the function area of the new window.

FIG. 6A shows the display when the window D has the ownership of the function key. Under this condition, if function select key 43 is input, the intensifying frame is displayed in the function area C1 of window C, as shown in FIG. 6B. The display apparatus is placed in a state that the command in window C can be specified by the function key.

As seen from the foregoing description, when the area detector 31 decides that the indicia is located in the area in the top window as displayed by the display means (CRT 22), the shape setting means (the cursor shape setter 33, for example) changes the shape of the indicia into the shape as set in that window. When the indicia is located in any window other than the top window being displayed, the shape of the indicia is changed into the standard shape. In this way, the indicia can be discriminated for each window. Even if the cursor is moved the operator can smoothly operate the display apparatus without being confused. Furthermore, the response speed of the indicia is improved.

It should be understood that this invention may variously be changed and modified within the scope of the invention.

In the above-mentioned embodiment, the shapes of indicia are different, thereby identifying specific windows. Also, the color of the indicia may be changed for the same purpose as shown in FIGS. 8A to 8C. Additionally, the indicia may be intermittently activated in a

blinking pattern in at least one window, for example, as shown in FIGS. 9A to 9C. The designer can use any method to distinguish the indicia displayed in the top window from those displayed in other windows while remaining within the scope and spirit of the present invention.

What is claimed is:

1. An information processing apparatus comprising: means for displaying a plurality of window areas, each of which includes information to be processed and each of which has an associated one of a plurality of distinguishing indicia, wherein a window area at least partially overlapped and hidden by others of said plurality of windows defines a hidden window area; means for determining a window area which is not a hidden window area; and means for causing display, in the window area determined not to be hidden window area, of a distinguishing indicia associated with the window area determined not to be a hidden window area.
2. An apparatus according to claim 1, wherein said distinguishing indicia has a predetermined shape.
3. An apparatus according to claim 1, wherein said distinguishing indicia has a predetermined color.
4. An apparatus according to claim 1, wherein said distinguishing indicia has a predetermined blinking state.
5. An apparatus according to claim 1, wherein said distinguishing indicia has a predetermined cursor shape.
6. An information apparatus comprising: means for displaying a plurality of window areas, each of which has an associated task, at least two of said plurality of window areas being displayed in an overlapping manner; means for determining an uppermost window area and another window area of the plurality of window areas, wherein at least a portion of the other window area is covered by the uppermost window area; and means for causing display of a distinguishing indicia in the uppermost window area and for displaying a standard indicia in the other window area.
7. An apparatus according to claim 6, wherein said standard indicia has a different shape from a shape of said distinguishing indicia.
8. An apparatus according to claim 6, wherein said standard indicia has a different color from a color of said distinguishing indicia.
9. An apparatus according to claim 6, wherein said standard indicia has a different blinking state from a blinking state of said distinguishing indicia.
10. An apparatus according to claim 6, wherein said standard indicia has a different cursor shape than a cursor shape of said distinguishing indicia.
11. An information processing apparatus comprising: means for displaying a plurality of window areas and an indicia, wherein each of the window areas has an associated task and an associated distinguishing indicia, said displayed indicia being one of the distinguishing indicia, and wherein a window area not overlapped by others of said window areas defines a top window area; means for moving the indicia displayed on said display means; means for determining the top window area; and means for changing said indicia to a distinguishing indicia corresponding to the top window area when said moving means moves the indicia into the top window area.

* * * *



US005640498A

United States Patent [19]

[11] Patent Number: **5,640,498**

Chew

[45] Date of Patent: **Jun. 17, 1997**

[54] ACCESSBAR ARBITER

[75] Inventor: **Chee Heng Chew, Redmond, Wash.**

[73] Assignee: **Microsoft Corporation, Redmond, Wash.**

[21] Appl. No.: **468,653**

[22] Filed: **Jun. 6, 1995**

[51] Int. Cl.⁶ **G06F 15/00**

[52] U.S. Cl. **395/133**

[58] Field of Search **395/138, 133, 395/162, 164; 345/117, 118**

[56] References Cited

U.S. PATENT DOCUMENTS

5,424,966 6/1995 Hirayama **395/155**

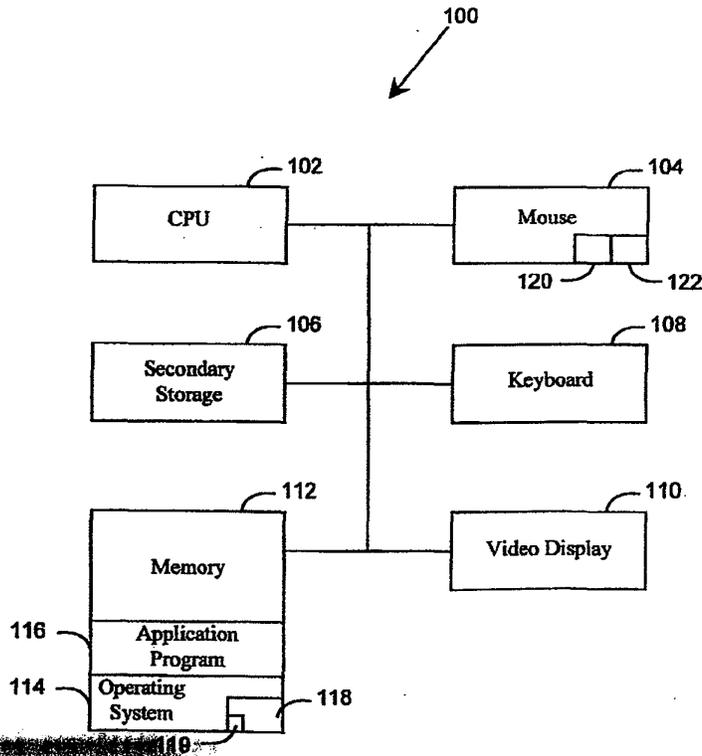
*Primary Examiner—Phu K. Nguyen
Attorney, Agent, or Firm—Seed and Berry LLP*

[57] ABSTRACT

As accessbar arbiter is provided that resolves conflicting requests from screen objects for locations on a video display. In addition, the accessbar arbiter governs the behavior of screen objects so as to prevent one screen object from

negatively affecting another screen object. An example of these screen objects includes accessbars. An accessbar is a consistently visible user interface element that provides a user with access to computer programs. In a computer system, accessbars are typically anchored at the edge of the display and there is no limit to the number of accessbars that can appear on the display at a given time, thus problems may arise. For example, more than one accessbar may wish to be displayed at a given location. In addition, the functionality of one accessbar may affect the processing of other accessbars. Thus, for the accessbars displayed on a computer display to operate together, a centralized mechanism for governing accessbars' location and behavior is desirable. The system described herein provides such a mechanism, known as an accessbar arbiter. The system described herein governs accessbars' location by receiving requests for proposed locations and by granting the requests if the proposed locations would not conflict with another accessbar. If such a conflict would occur, the system described herein provides an alternate location. The system described herein governs accessbars' behavior by providing a number of notifications to the accessbars whenever an event occurs that may affect the processing of the accessbars. Such an event includes changes to another accessbar on the same side of the display or the start of a fullscreen application, such as a screen saver.

24 Claims, 18 Drawing Sheets



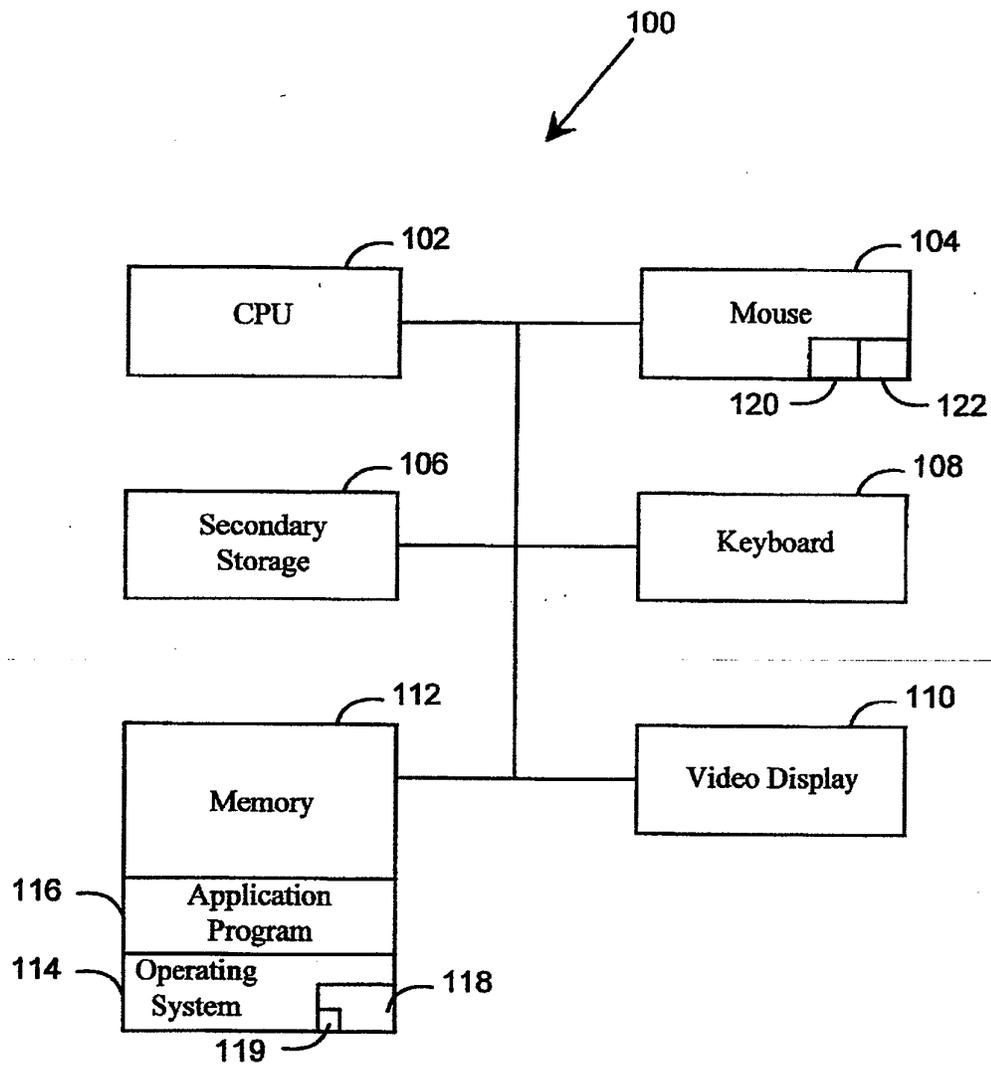


FIG. 1

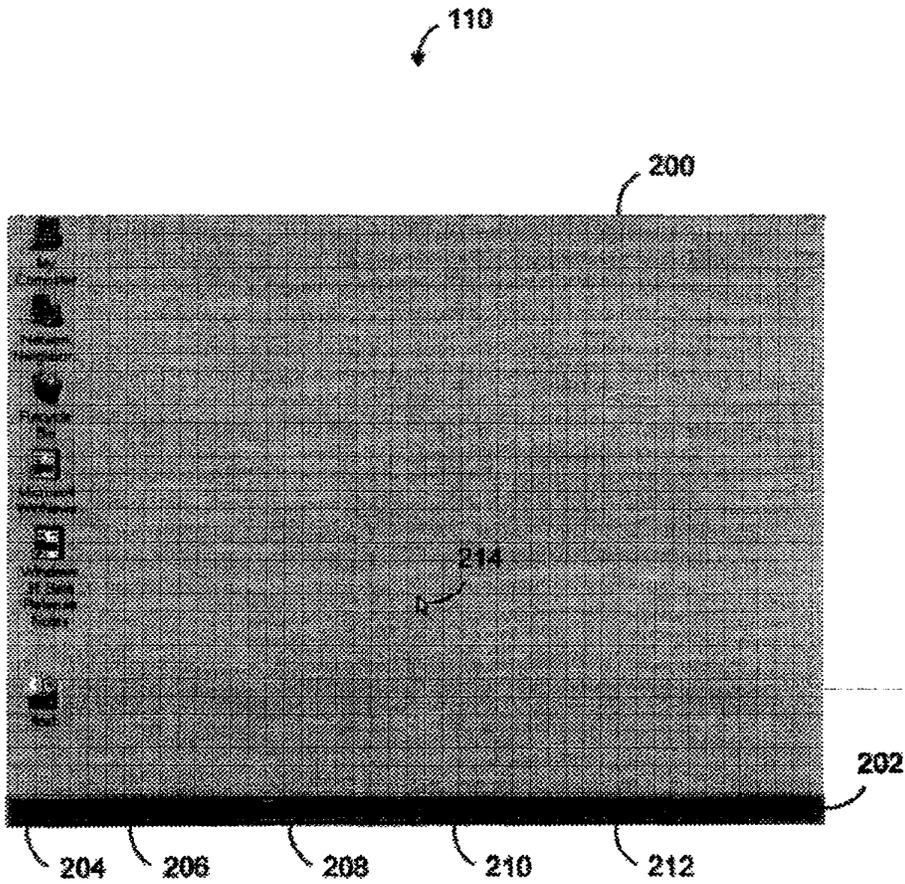


FIG. 2A

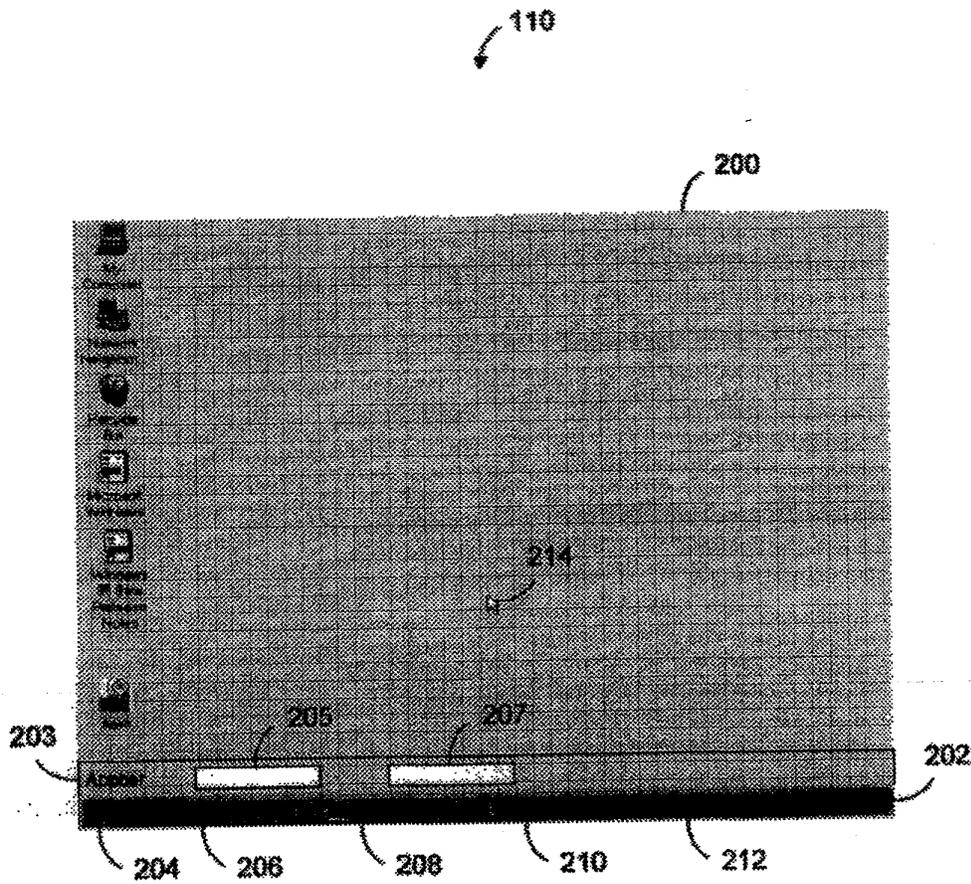


FIG. 2B

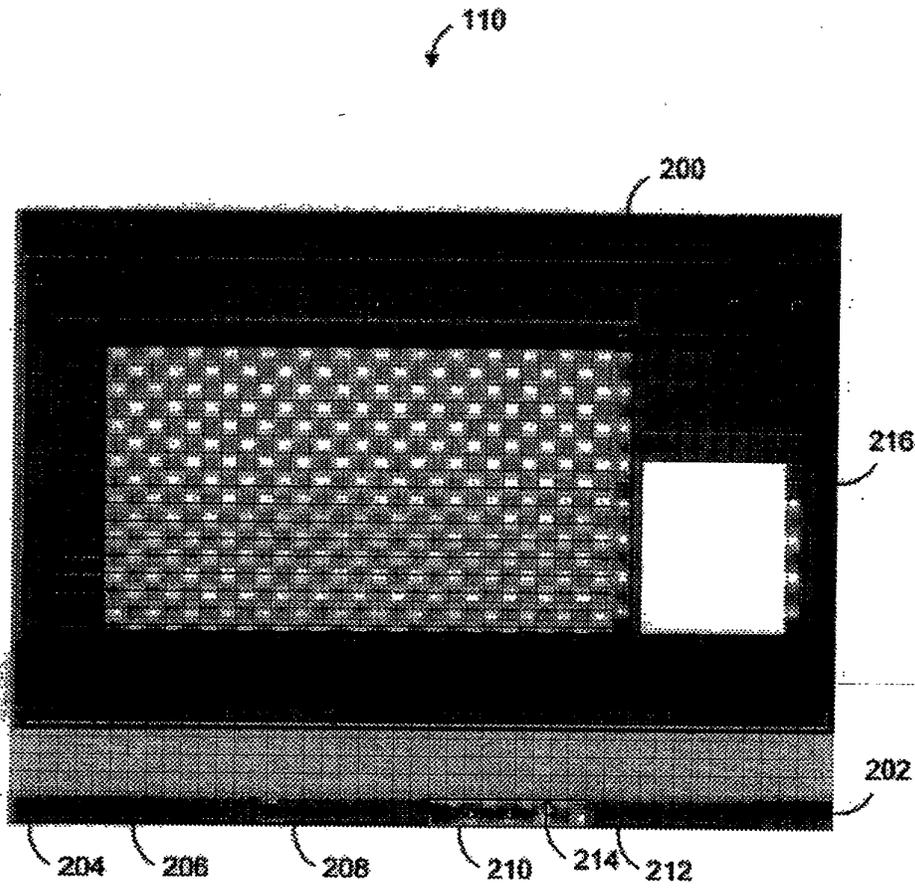


FIG. 2C

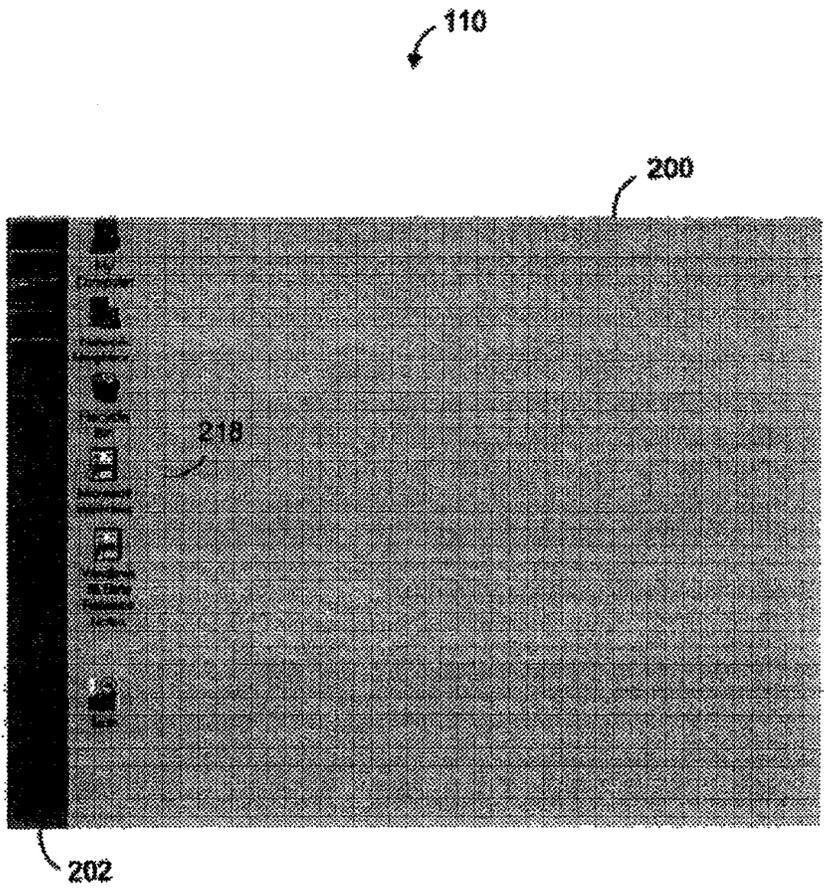


FIG. 2D

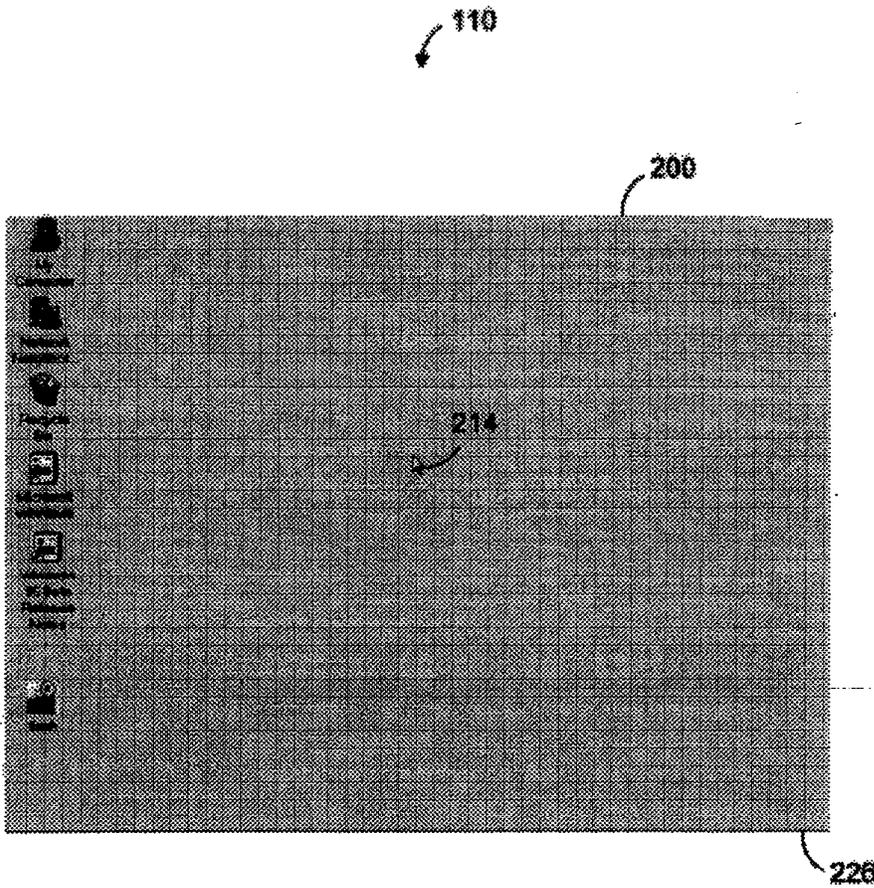


FIG. 2E

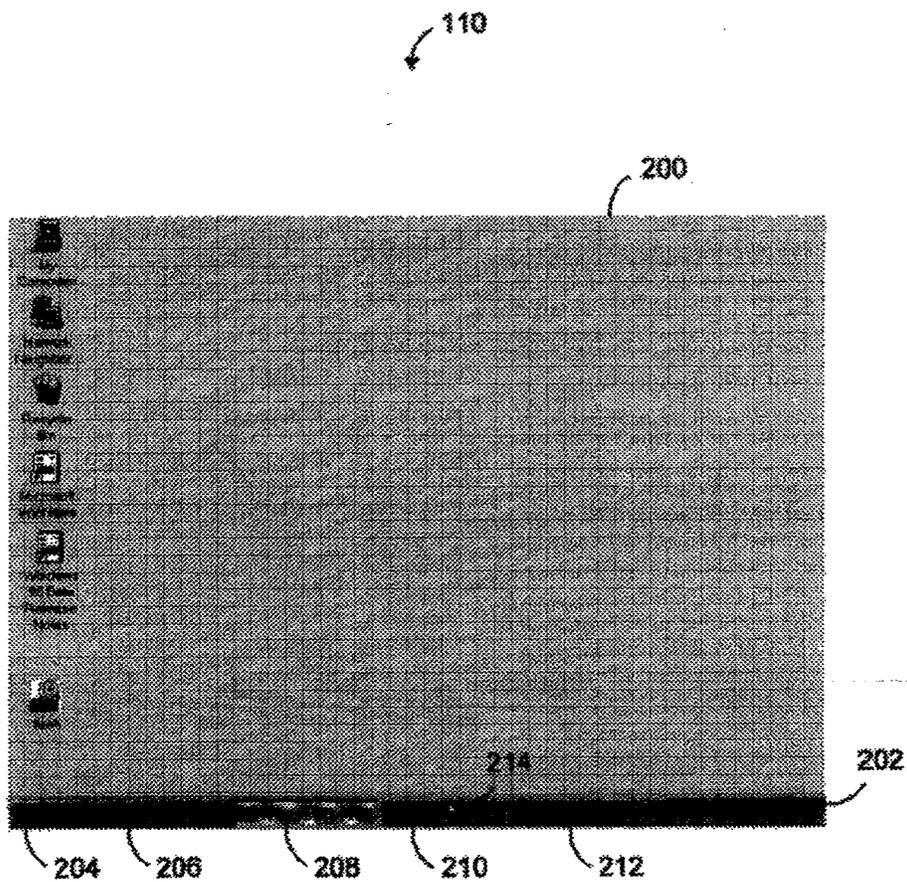


FIG. 2F

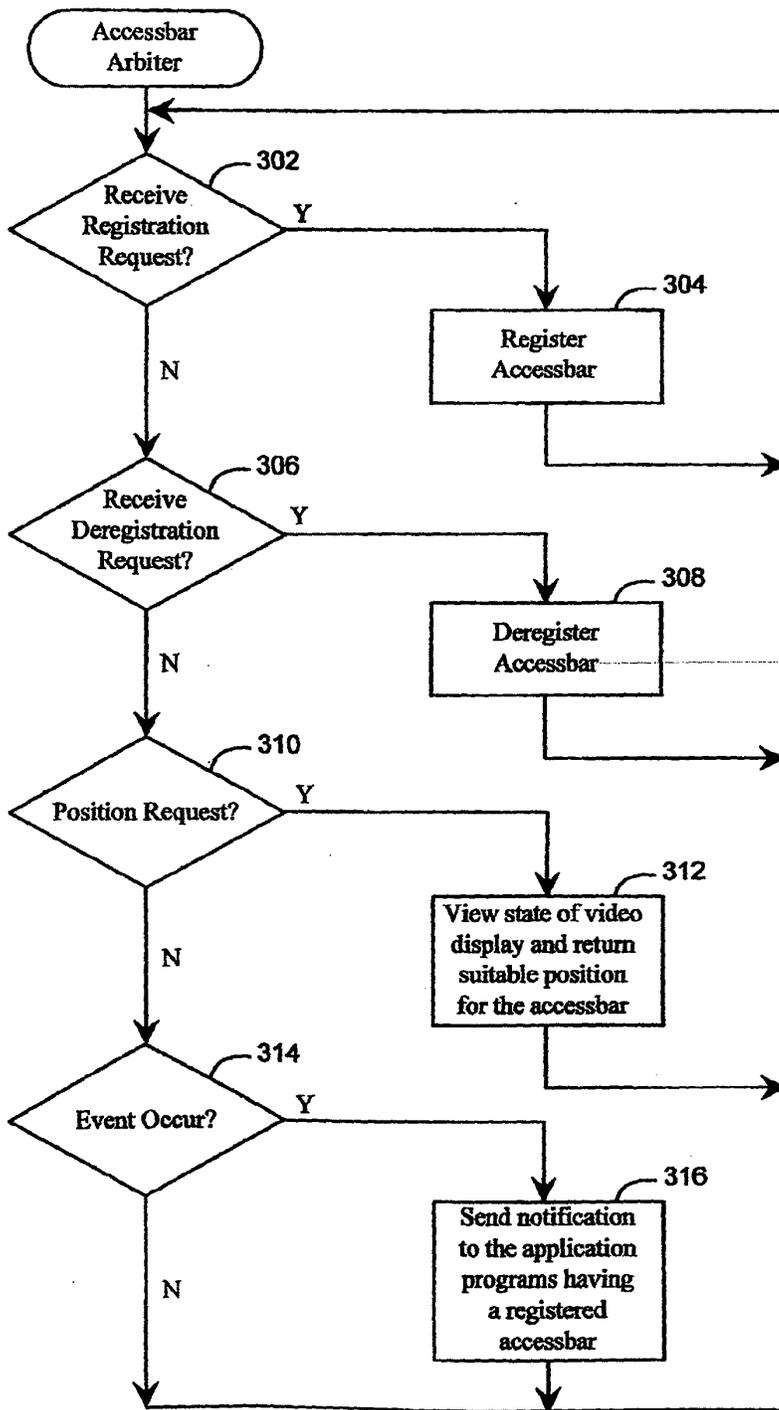


FIG. 3

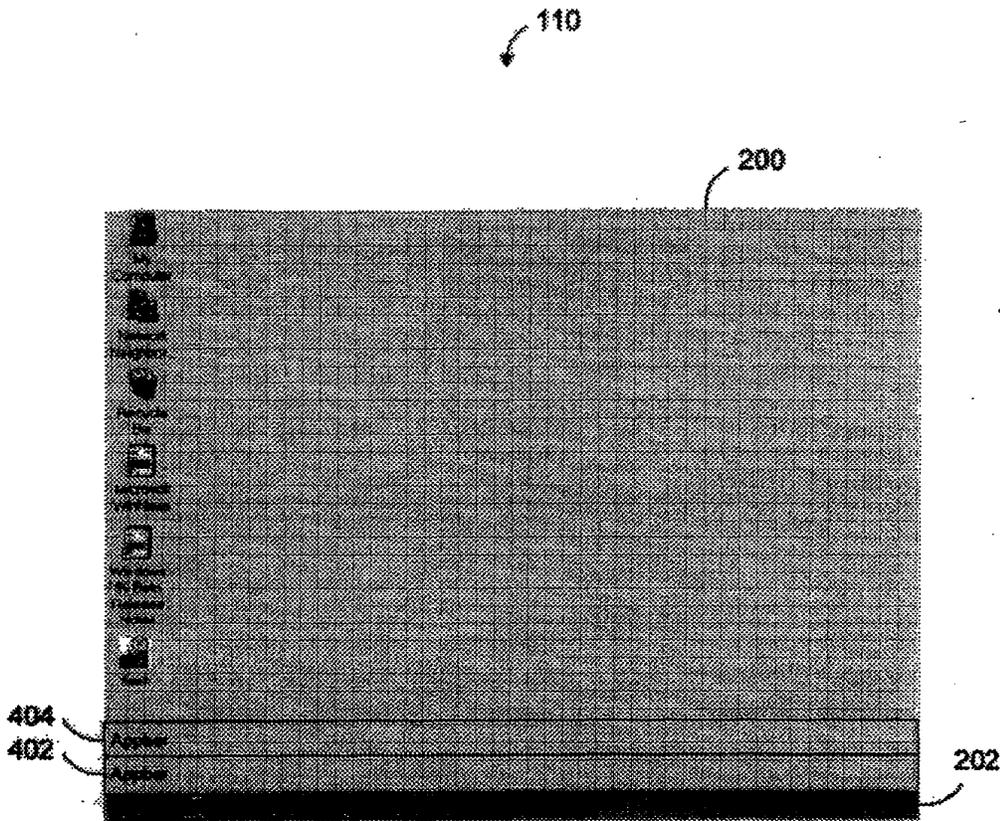


FIG. 4A

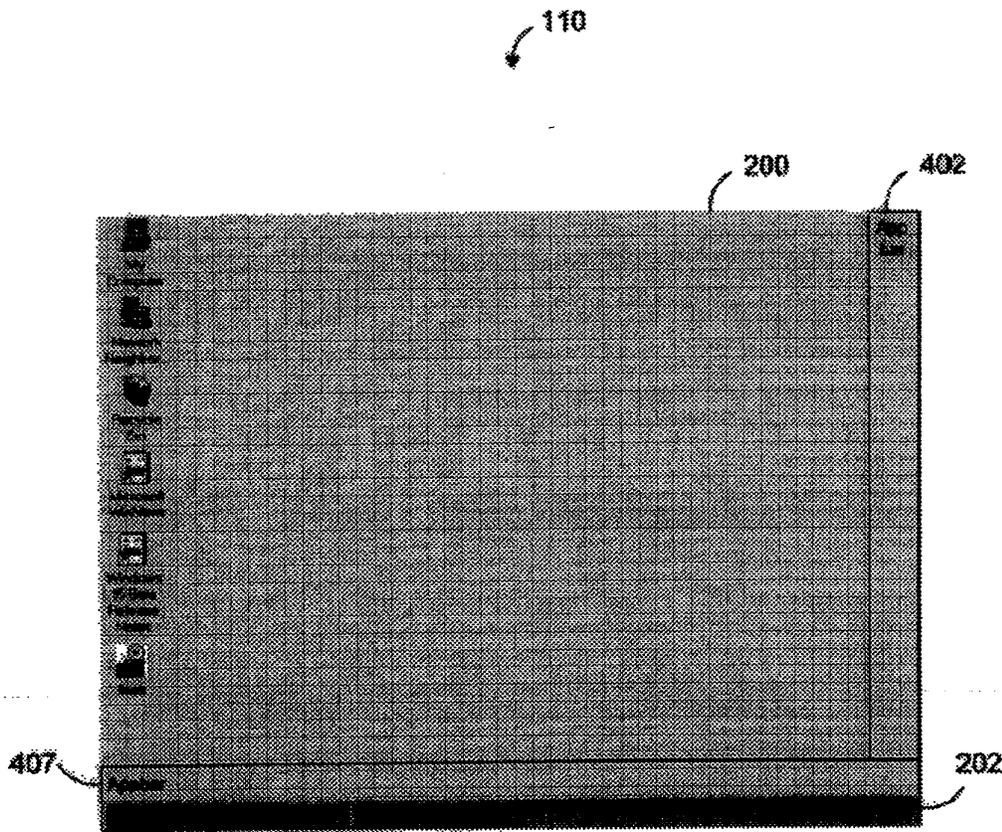


FIG. 4B

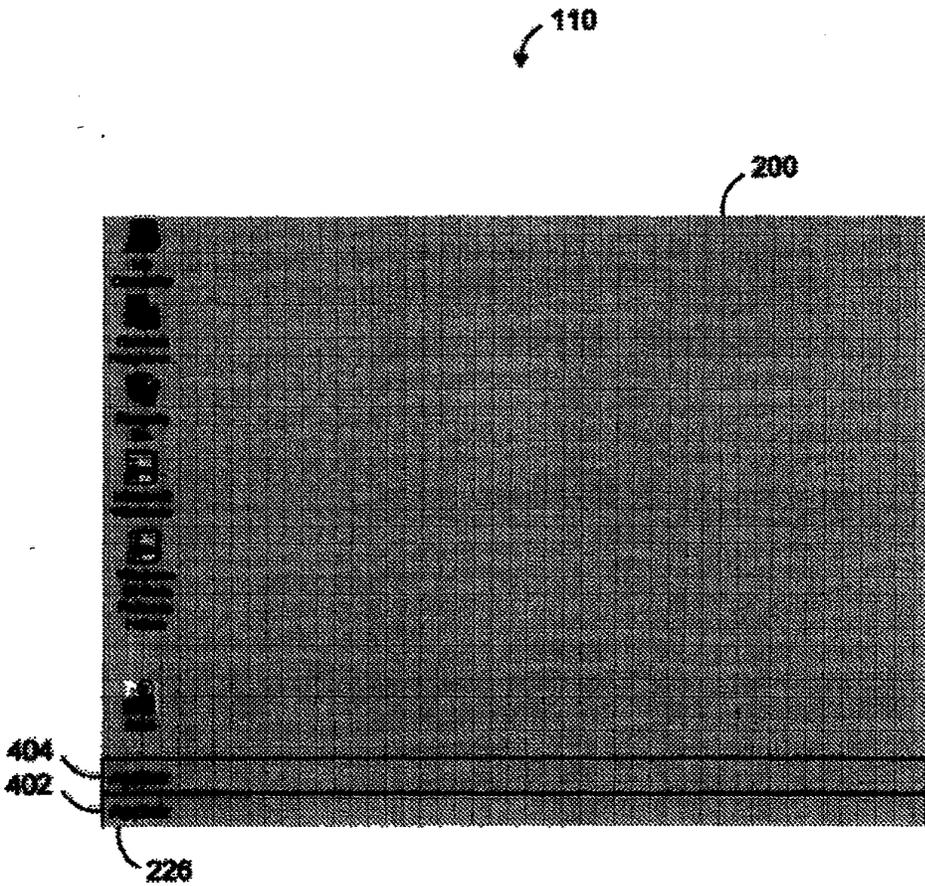


FIG. 4C

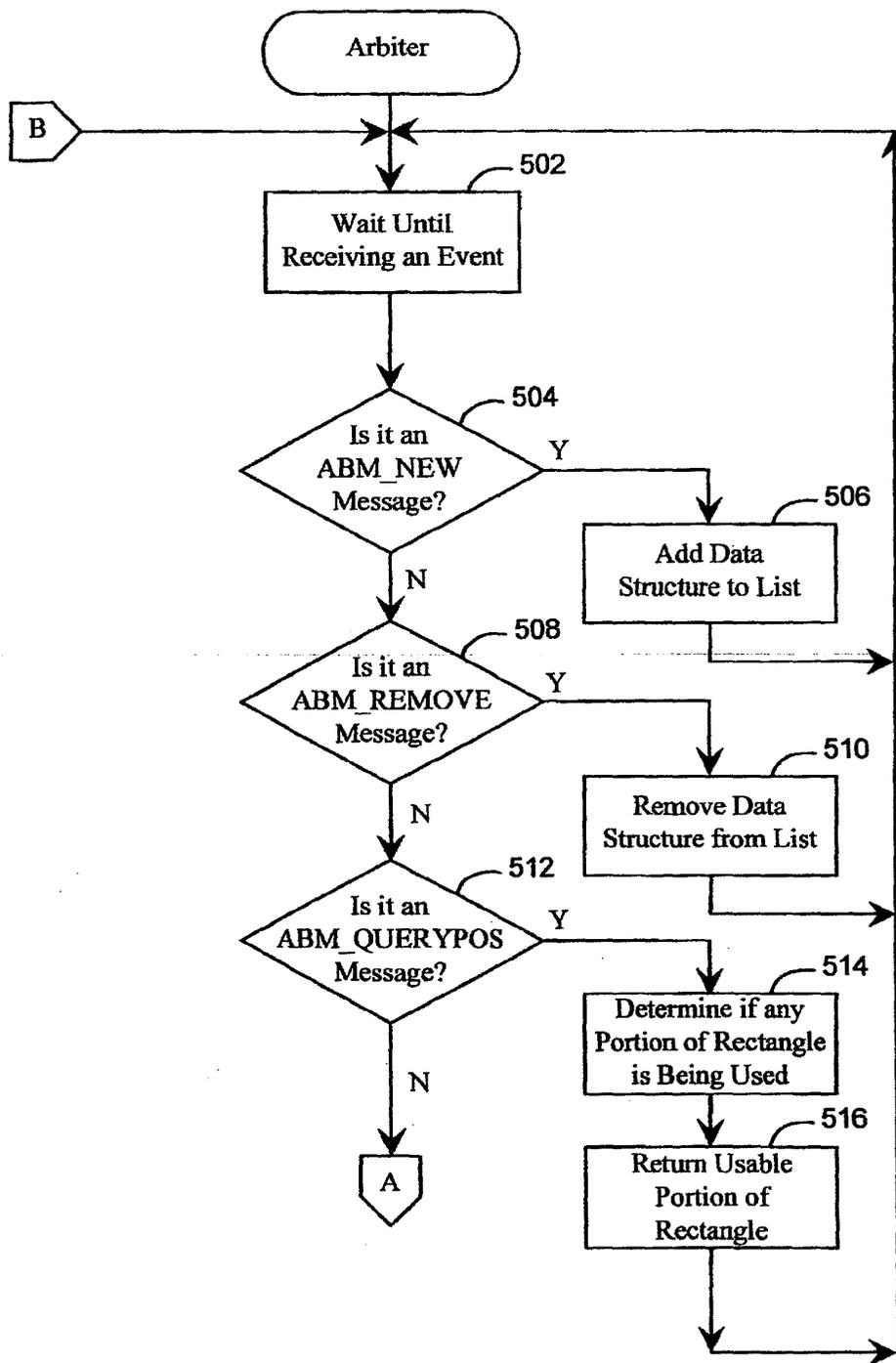


FIG. 5A

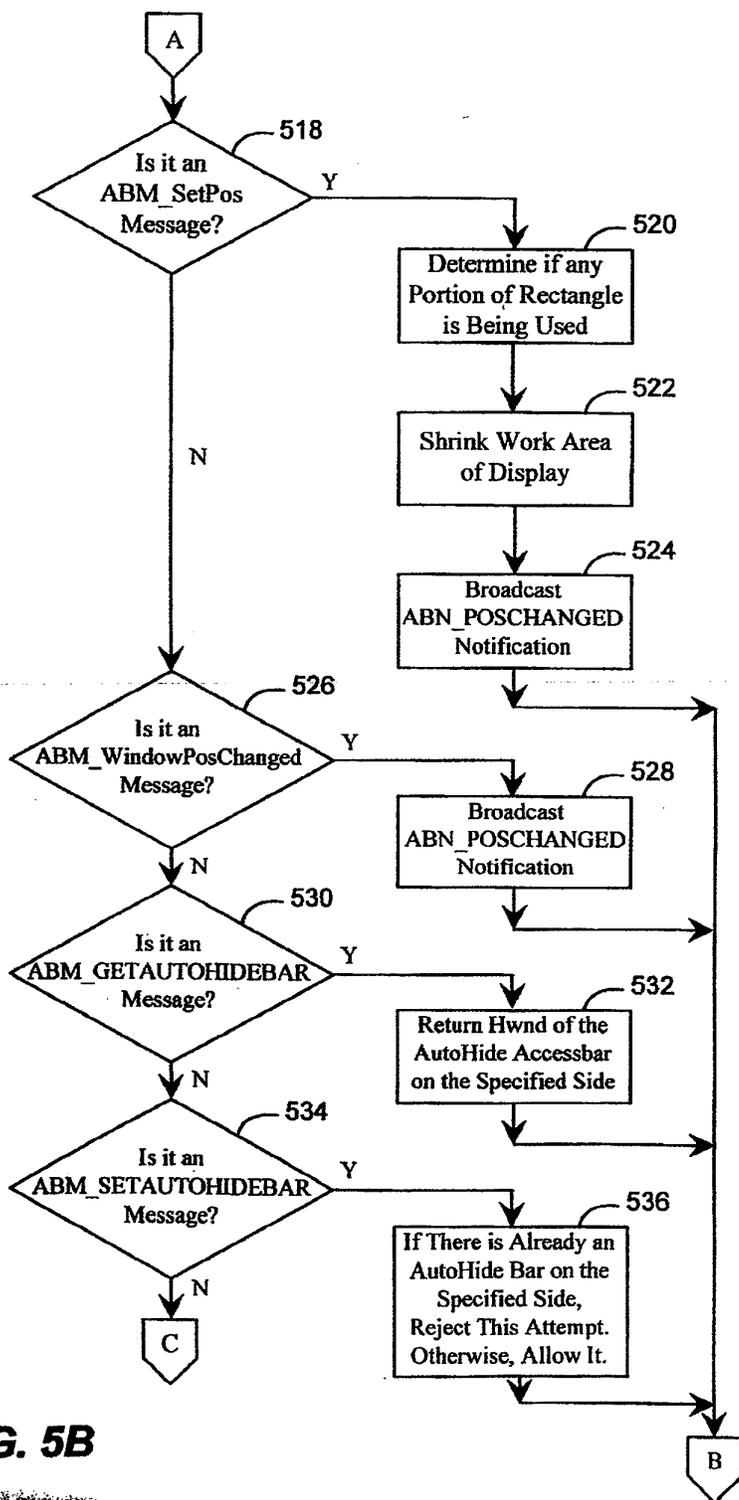


FIG. 5B

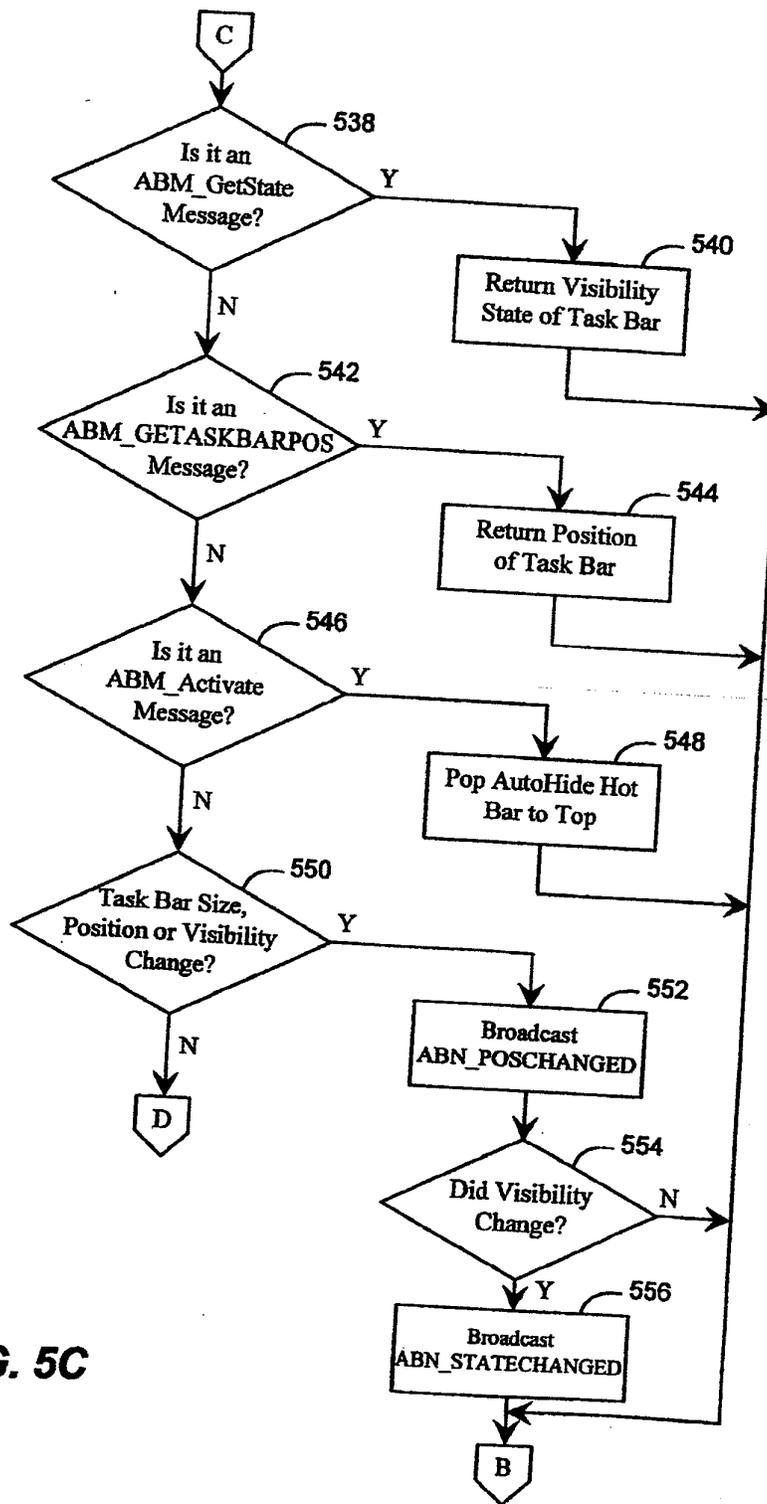


FIG. 5C

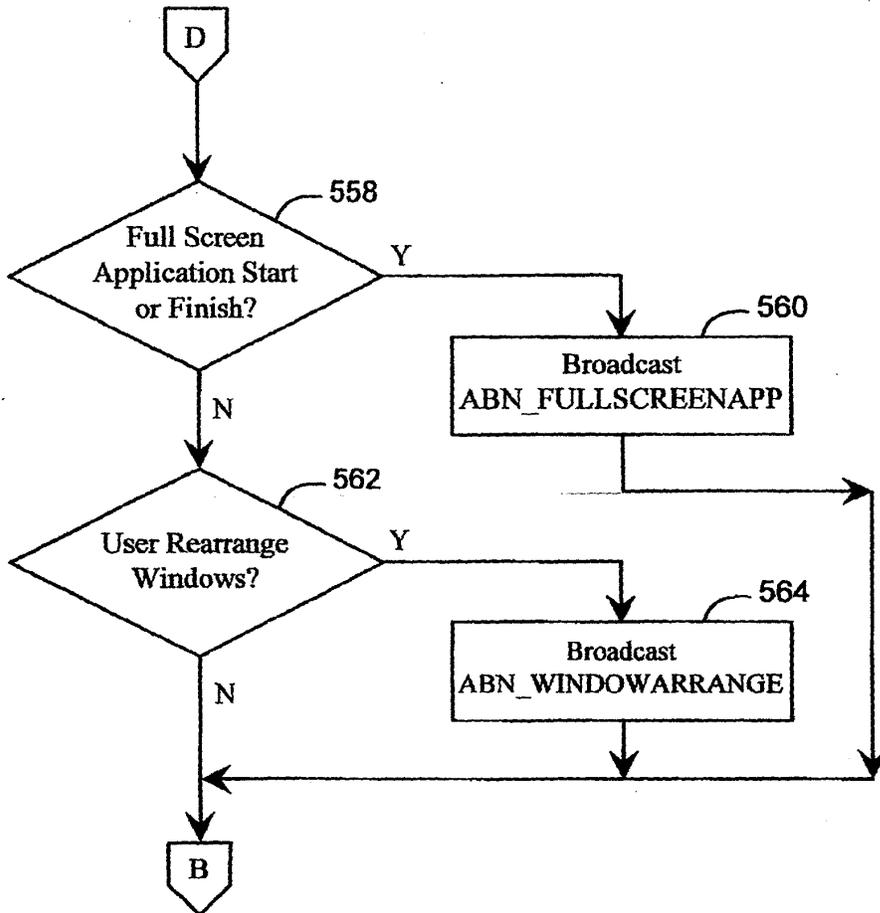


FIG. 5D

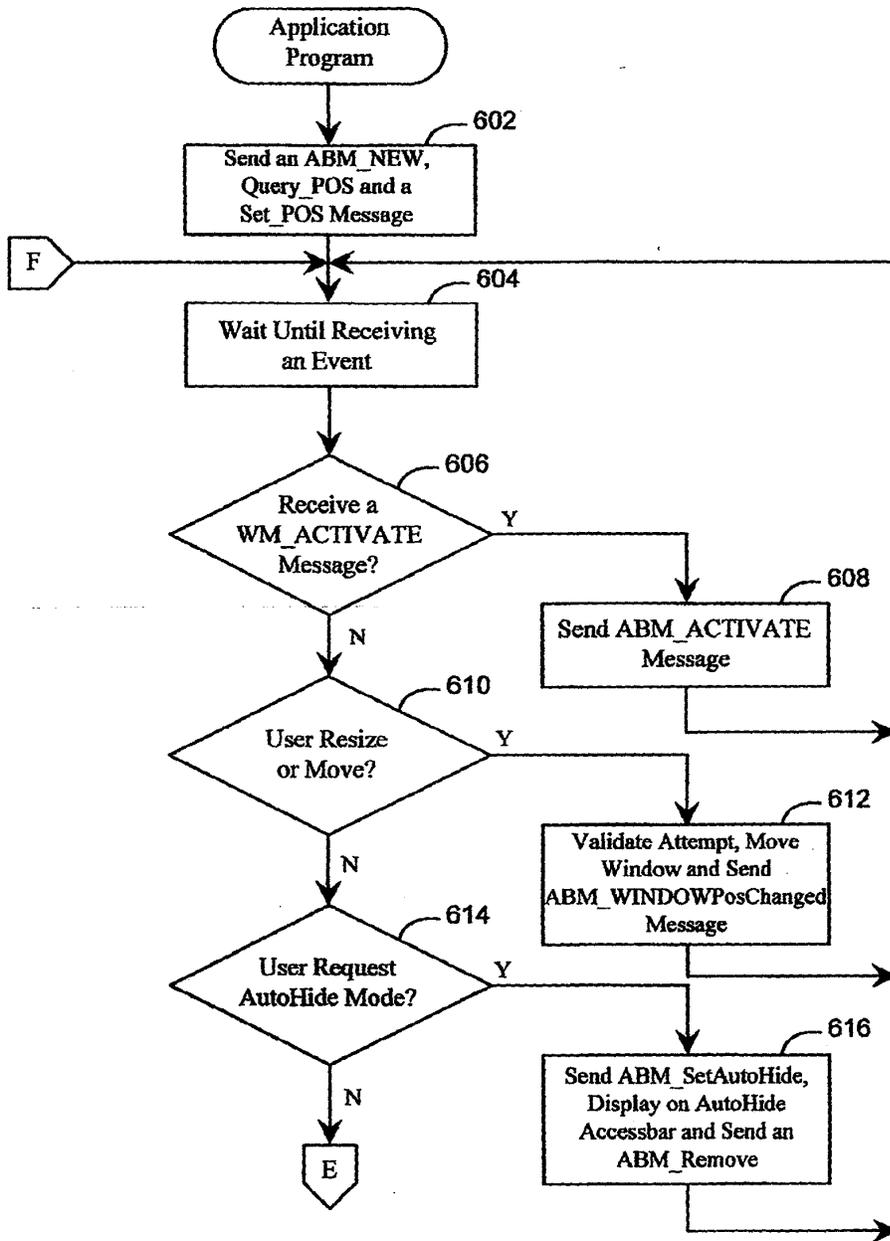


FIG. 6A

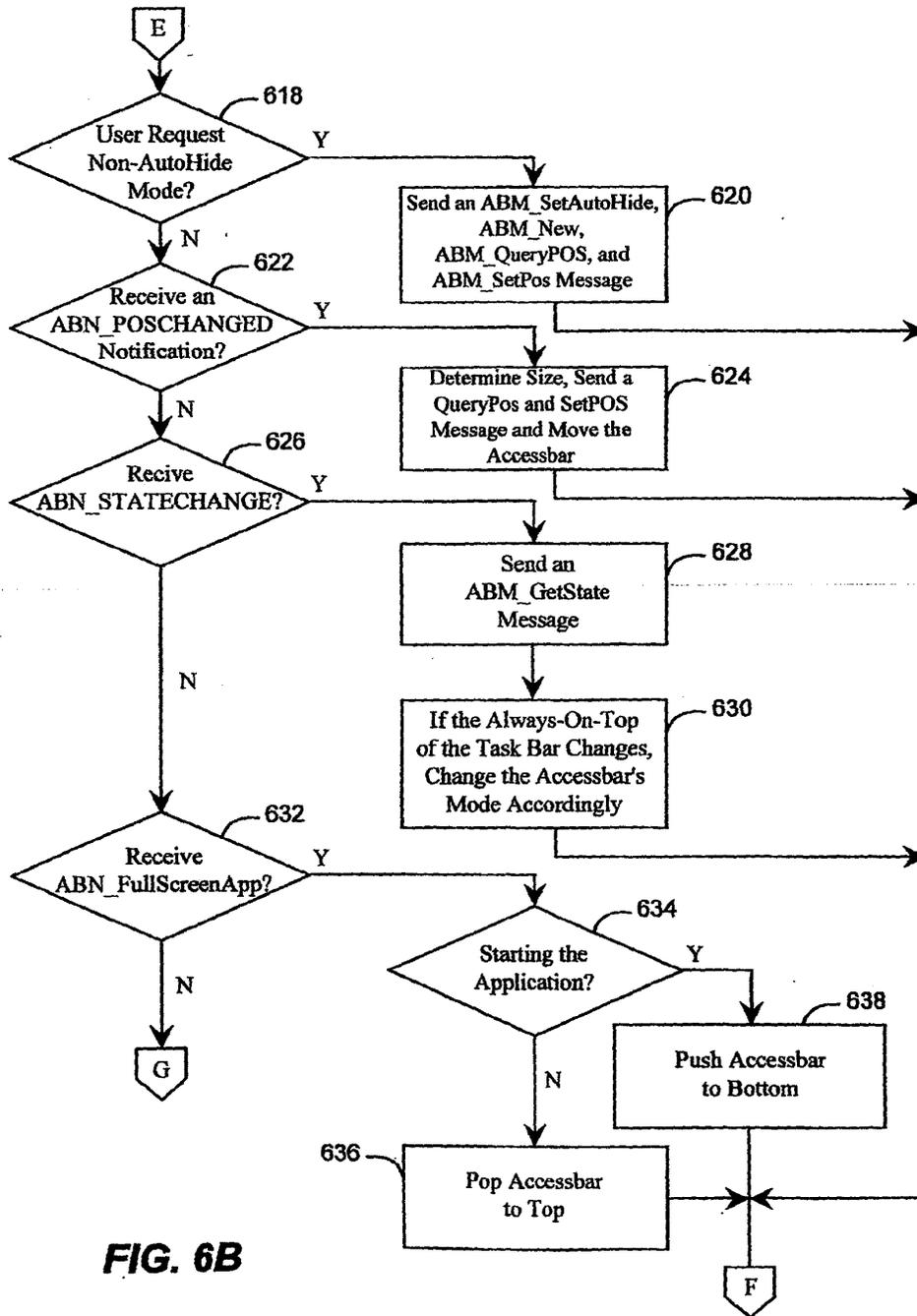


FIG. 6B

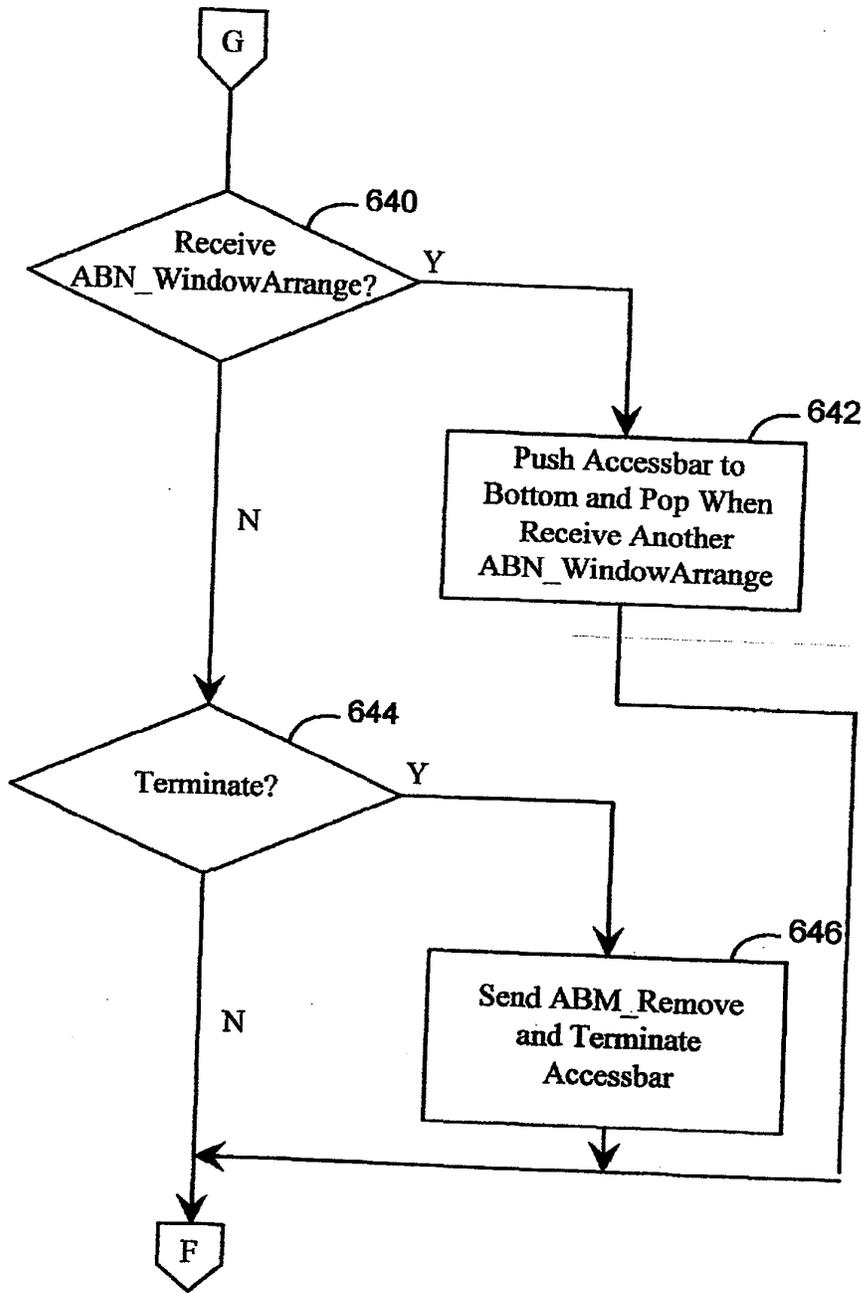


FIG. 6C

ACCESSBAR ARBITER

TECHNICAL FIELD

This invention relates generally to data processing systems and, more particularly, to the management of accessbars on a video display of a data processing system.

BACKGROUND OF THE INVENTION

Today's computer systems typically display screen objects on the video display of the computer system. A screen object is a user-visible object displayed on the video display that conveys information to the user. It is common for more than one screen object to be displayed on the video display at a time. Thus, conflicts arise when two screen objects wish to be displayed at the same location on the video display. In conventional computer systems, these conflicts are left unresolved.

There are other types of conflicts that arise between screen objects displayed on a video display. For example, one screen object may perform functionality that can have an effect on another screen object. Such functionality includes when one screen object moves so as to overlap a second screen object, thereby preventing the second screen object from being visible. These conflicts are left unresolved by conventional computer systems.

SUMMARY OF THE INVENTION

An accessbar arbiter is described herein that resolves conflicting requests from screen objects for locations on a video display. In addition, the accessbar arbiter governs the behavior of screen objects so as to prevent one screen object from affecting another screen object. An example of these screen objects includes accessbars. An accessbar is a consistently visible user interface element that provides a user with access to computer programs. In a computer system, accessbars are typically anchored at the edge of the display, and there may be many accessbars that can appear on the display at a given time, thus potential conflicts may arise that require arbitration. For example, more than one accessbar may wish to be displayed at a given location. In addition, the functionality of one accessbar may affect the processing of other accessbars. Thus, for the accessbars displayed on a computer display to operate together, a centralized mechanism for governing accessbars' location and behavior is desirable. The system described herein provides such a mechanism, known as an accessbar arbiter. The system described herein governs accessbars' location by receiving requests for proposed locations and by granting the requests if the proposed locations would not conflict with another accessbar. If such a conflict would occur, the system described herein provides an alternate location. The system described herein governs accessbars' behavior by providing a number of notifications to the accessbars whenever an event occurs that may affect the processing of the accessbars. Such an event includes changes to another accessbar on the same side of the display or the start of a fullscreen application, such as a screen saver.

In accordance with a first aspect of the present invention, a method is practiced in a computer system having a video display for displaying screen objects, a computer program for causing the video display to display one of the screen objects, and an arbiter for determining a location on the video display for the displaying of the screen objects. In accordance with this first aspect of the present invention, a request is received by the arbiter from the computer program

containing a proposed location of the screen object on the video display. The arbiter determines whether to display the screen object at the proposed location. When it is determined that the screen object is to be displayed at the proposed location, the computer program displays the screen object at the proposed location on the video display. However, when it is determined that the screen object is not to be displayed at the proposed location, the arbiter determines an alternative location, and the computer program displays the screen object at the alternative location on the video display.

In accordance with a second aspect of the present invention, a data processing system is provided that includes a video display for displaying a plurality of screen objects and a memory. The memory further comprises an arbiter and a computer program. The arbiter has a receive component for receiving a request to display a screen object at a requested location. The arbiter also has a determination component for determining whether to display the screen object at the requested location. Further, the arbiter has a location component for determining an alternative location when the determination component determines not to display the screen object at the requested location and the arbiter has an indication component for returning an indicated location. The indicated location is the requested location when the determination component determines to display the screen object at the requested location. The indicated location is the alternative location when the determination component determines not to display the screen object at the requested location. The computer program sends the request to the arbiter to display the screen object at the requested location, receives the indicated location from the arbiter and causes the video display to display the screen object at the indicated location.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 depicts a computer system suitable for practicing the preferred embodiment of the present invention.

FIG. 2A depicts a taskbar displayed on the video display of the computer system of FIG. 1.

FIG. 2B depicts an appbar displayed with the taskbar of FIG. 2A.

FIG. 2C depicts a window displayed after the invocation of a taskbar button on the taskbar of FIG. 2A.

FIG. 2D depicts the movement of the taskbar of FIG. 2A from the bottom edge of the video display to the left-hand side edge (relative to the user) of the video display and the resulting displacement of the icons on the video display from their original location.

FIG. 2E depicts the taskbar of FIG. 2A as it appears in autohide mode.

FIG. 2F depicts the displaying of the taskbar of FIG. 2E in autohide mode after the user has touched the hotbar with the mouse cursor.

FIG. 3 depicts a flowchart of a functional overview of the accessbar arbiter of the preferred embodiment of the present invention.

FIG. 4A depicts the state of the video display of the computer system of FIG. 1 containing a taskbar and two appbars.

FIG. 4B depicts the state of the video display after one of the appbars of FIG. 4A has been moved from the bottom edge of the video display to the right-hand side edge of the video display.

FIG. 4C depicts the state of the video display after the taskbar of FIG. 4A has been placed in autohide mode.

FIGS. 5A, 5B, 5C, and 5D depict a flowchart of the steps performed by the accessbar arbiter of the preferred embodiment of the present invention.

FIGS. 6A, 6B, and 6C depict a flowchart of the steps performed by an application program utilizing the accessbar arbiter of the preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The preferred embodiment of the present invention utilizes an accessbar arbiter for resolving conflicts between screen objects competing for a location on a video display. In addition, the accessbar arbiter governs the functionality of screen objects to prevent the functionality of one screen object from negatively affecting another screen object. An example of these screen objects includes accessbars. An "accessbar" is a general term that encompasses both taskbars and appbars. A "taskbar" is a consistently visible user interface element that informs a user of which tasks are active and have an active window. An active task is a computer program that is loaded into primary memory. The taskbar is constructed so that it is not obscurable by open windows, unless the user so selects to obscure the taskbar. The taskbar also displays indicators or buttons used for non-minimizing currently active task windows. The taskbar includes a button for each active task. Users merely need to single click on a taskbar button provided for a task to open the task window. These buttons may also be used to switch tasks. The taskbar usually remains anchored at a fixed location on the user interface (see FIG. 2A) which may be changed by a user. The taskbar runs typically as part of the operating system and provides other functionality that is more clearly described in U.S. patent application Ser. No. 08/354,916, entitled "Taskbar With Start Menu," which is hereby incorporated by reference.

An "appbar" shares many of the same characteristics as a taskbar with the exception that an appbar provides access to computer programs that are both loaded into memory as well as not loaded into memory. An appbar serves as a shortcut to run related computer programs. Typically an appbar has a button for each related computer program to which the appbar provides access. These buttons, unlike the comparable taskbar buttons, are permanently visible. For example, a company that develops multiple computer programs may implement an appbar with a button that provides access to each computer program so that users may have easy access to the company's computer programs.

In a computer system, there are usually a number of accessbars that can appear on the display at a given time. Since the accessbars are typically anchored at a fixed location, such as the edge of the display, problems may arise. For example, more than one accessbar may wish to be displayed at a particular location. Thus, the functionality of one accessbar may affect other accessbars. Thus, for the accessbars displayed on a computer display to operate together, a centralized mechanism for governing accessbars' location and behavior is desirable. The preferred embodiment of the present invention provides such a mechanism, known as an accessbar arbiter.

FIG. 1 is a block diagram of a computer system 100 that is suitable for practicing the preferred embodiment of the present invention. The computer system 100 includes a central processing unit (CPU) 102, which has access to several peripheral devices, including a mouse 104, a keyboard 108, and a video display 110. The mouse 104 includes at least a left mouse button 120 and a right mouse button

122. The CPU 102 also has access to memory 112 and secondary storage 106. The memory 112 holds copies of an application program 116 and an operating system 114. The operating system 114 contains the accessbar arbiter 118 of the preferred embodiment of the present invention. The accessbar arbiter 118 provides an application program interface 119 so as to allow access to the arbiter. Those skilled in the art will appreciate that the accessbar arbiter 118 of the present invention is not limited to being part of an operating system; rather, the accessbar arbiter 118 may also be implemented in an application program. For purposes of the discussion below, it is assumed that the operating system 114 is the "MICROSOFT" "WINDOWS" 95 Operating System from Microsoft Corporation. Those skilled in the art will appreciate that the present invention may also be practiced with other operating systems.

Accessbars are typically implemented by either an application program 116 or the operating system 114 as a window. A window is created based upon a window class. The window class identifies a window procedure that processes messages that are sent to the window. The operating system 114 is a message-based operating system such that operations are achieved by passing messages between windows. The operating system 114 supports a RegisterClass() function that enables an application to register window classes. The window procedure is a function that receives and processes messages that are sent to the window.

In order to more fully understand the processing of the accessbar arbiter of the preferred embodiment of the present invention, the functionality of an accessbar requires explanation. Thus, the following description is presented in two parts: first, an overview of the functionality of accessbars is presented and second, the details of the accessbar arbiter are presented.

Accessbar Overview

FIG. 2A depicts an example of the graphical user interface provided by the operating system 114. A taskbar 202 is positioned at the bottom of a virtual desktop 200 on the video display 110. In the example shown in FIG. 2A, the taskbar 202 is a rectangle that extends from the left-hand side (relative to the user) of the video display to the right-hand side of the video display and is anchored to the bottom edge of the video display. The taskbar 202 includes a start menu button 204 that provides access to programs so that the programs may be invoked. The taskbar 202 also indicates the tasks that are active by displaying taskbar buttons such as buttons 206, 208, 210, and 212. A user displays an active task window, for example, from the taskbar 202 of FIG. 2A by positioning a mouse cursor 214 to point at a taskbar button using the mouse or other suitable input device and then single clicking the mouse button to open the window associated with the task to display it in a visible, non-minimized state. For example, FIG. 2C depicts a window 216 displayed in non-minimized form after the user has depressed taskbar button 210 of the taskbar of FIG. 2A. The window 216 displays the user interface of the Schedule+ scheduling program available from Microsoft Corporation. However, with reference to FIG. 2B, on an appbar 203, the buttons 205, 207 refer to both computer programs that are loaded into memory and computer programs that have not been loaded into memory. Thus, clicking on appbar buttons 205, 207 launches the computer program if it has not been loaded into memory and then displays the window associated with the computer program in a non-minimized form, regardless of whether the computer program had been loaded into memory. In the following, the functionality of an accessbar is explained.

taskbar, however, it should be appreciated that an appbar performs similar functionality.

As previously stated, although an accessbar is usually anchored to a side of the video display 110, it can be moved to another side of the video display. The movement of an accessbar occurs in response to user input. The type of user input required depends on the implementation of the accessbar. For example, in order to move the taskbar 202 shown in FIG. 2A, the user moves the mouse cursor 214 to refer to a location on the taskbar without a taskbar button, depresses the left mouse button and then moves the mouse to position the mouse cursor near a different edge of the video display 110. FIG. 2D depicts the state of the video display 110 after the mouse button has been released and a move of the taskbar 202 has occurred. In general, when an accessbar has been moved, any screen objects (e.g., windows or icons) 218 currently being displayed within the area to be occupied by the accessbar are moved out of the way of the accessbar in a direction toward the center of the video display. In this manner, the screen objects 218 remain visible and accessible to the user.

Each accessbar on the video display 110 typically has properties associated therewith. These properties can be accessed in a number of ways, such as clicking the right mouse button when the mouse cursor refers to an area of the accessbar without a taskbar button. Two properties of particular relevance are the "always-on-top" property and the "autohide" property. The always-on-top property refers to the accessbar being placed in a mode where it is always displayed at the top of the Z order. The "Z order" refers to the depth of the screen objects displayed on the video display 110. That is, the video display is typically viewed as a two dimensional plane of points that are referenced using Cartesian coordinates. A third axis, the 'Z' axis, may also be designated that identifies the logical depth of screen objects on the video display. The "Z order" refers to the order of the screen objects along the Z axis. The top of the Z order refers to a screen object that appears closest to the user and is unobstructed by another screen object. Consequently, the bottom of the Z order refers to a screen object that appears furthest away from the user and can be obstructed by many screen objects.

The always-on-top property has a particular effect when a user attempts to resize or move a window on the video display. When attempting to resize a window, the user is prevented from resizing the window so as to encroach upon an area of the video display 110 occupied by an accessbar. In other words, the "work area" of the video display 110 is reduced when an accessbar is displayed in always-on-top mode. When an accessbar is placed in always-on-top mode and a window is moved so as to encroach upon the area occupied by the accessbar, the accessbar is displayed on top of the encroaching window (i.e., at the top of the Z order). Conversely, when an accessbar is not in always-on-top mode, the user may resize the window so that it does overlap the accessbar. Thus, the work area of the video display includes the area of the window occupied by the accessbar. Similarly, when the accessbar is not in always-on-top mode, the user may move the window so that it overlaps the accessbar.

Autohide mode refers to when an accessbar is initially presented to the user such that the accessbar is invisible, as shown in FIG. 2E. When invisible, instead of displaying the accessbar, a "hotbar" 226 is displayed that is only a couple of pixels in width so that it is barely visible to the user. The hotbar 226 acts as a mechanism for displaying an accessbar in a user visible manner. This is done by moving the mouse

cursor 214 so as to touch the hotbar 226 and thereupon (FIG. 2F) the accessbar (e.g., taskbar 202) reappears and becomes visible to the user.

The Accessbar Arbitrator

The preferred embodiment of the present invention provides an accessbar arbitrator to ensure that accessbars work properly together and work properly in response to outside events, such as when a fullscreen application is invoked. A "fullscreen application" is an application program that takes total control of the video display by displaying information at all portions of the video display. Examples of fullscreen applications include video games as well as screen savers. Thus, when a fullscreen application is invoked, all accessbars should be informed so that they can relinquish control of their area of the display. The accessbar arbitrator ensures that accessbars work properly together by acting as a centralized mechanism that centrally manages all accessbars. This centralized management (or arbitration) is facilitated by application programs that display an accessbar sending messages to the accessbar arbitrator and by the accessbar arbitrator sending notifications to the application programs when an event occurs. The messages, notifications and the events for which the accessbar arbitrator sends the notifications are further discussed below.

FIG. 3 depicts a flowchart of a functional overview of the accessbar arbitrator of the preferred embodiment. During its processing, the accessbar arbitrator may receive registration requests, deregistration requests, and position requests from application programs having an accessbar. In addition, the accessbar arbitrator detects if an event has occurred that may affect the processing of the accessbars displayed on the video display. The first step performed by the accessbar arbitrator is to determine if it received a registration request (step 302). If the accessbar arbitrator received a registration request, the accessbar arbitrator registers the accessbar by storing information pertaining to the accessbar (step 304). This information is further described below. If the accessbar arbitrator did not receive a registration request, the accessbar arbitrator determines if it received a deregistration request (step 306). In this case, the accessbar arbitrator deregisters the accessbar by deleting the stored information (step 308). Upon registering an accessbar with the accessbar arbitrator, an application program can make a request for a position for the accessbar on the video display. Thus, the accessbar arbitrator may receive a position request (step 310). The accessbar arbitrator, after receiving the position request, views the state of the video display, determines if there are any other accessbars currently displayed at the requested position and returns a response to the application program containing a suitable position for the accessbar (step 312). The accessbar arbitrator, in performing this processing, provides preferential treatment to the taskbar. That is, in the preferred embodiment, the taskbar does not request a position from the accessbar arbitrator. Instead, the taskbar merely moves to the position that it prefers and the accessbar arbitrator ensures that all other accessbars accommodate the preferences of the taskbar. It should be appreciated that another embodiment of the present invention could have the taskbar communicate with the accessbar arbitrator.

In addition to the functionality described above, the accessbar arbitrator detects if an event has occurred that may affect the functionality of the accessbars displayed on the video display (step 314). Such events include when a fullscreen application has taken control of the video display, when an accessbar anchored on the same side as the application program's accessbar has been moved or repositioned, or when an accessbar anchored on the same side as the

application program's accessbar has become autohidden. If such an event occurs, the accessbar arbiter sends a notification to the application programs having a registered accessbar that indicates the event that occurred (step 316). Notifications are discussed in more detail below.

To illustrate the need for notifications, consider FIG. 4A, which depicts three accessbars on the video display 110. One of the accessbars is the taskbar 202 and two of the accessbars are appbars 402, 404. If appbar 402 were moved to another edge of the video display, the application program displaying appbar 404 needs to be notified so that it can reposition the appbar closer to the taskbar 202. Otherwise, the area previously occupied by the appbar 402 would become unusable space on the video display 110. FIG. 4B depicts the state of the video display 110 after appbar 402 has been moved to the right-hand edge of the video display in accordance with the preferred embodiment of the present invention. In the preferred embodiment, the application program displaying appbar 404 is notified by the accessbar arbiter of the movement of appbar 402 and in response, the application program repositions the appbar 404 to prevent any wasted space on the video display 110.

Another example, referring back to FIG. 4A, is if the taskbar 202 were placed in autohide mode. If the appbar 402 and 404 were not notified, the area of the video display that was previously used for displaying the taskbar 202 would become unusable space. FIG. 4C depicts the state of the video display 110 after the taskbar 202 has been placed in autohide mode in accordance with the preferred embodiment. The application programs that display the appbars 402, 404 have been notified by the accessbar arbiter of the taskbar 202 being placed in autohide mode. In response, the application programs have repositioned the appbars 402, 404 to prevent any area of the video display 110 from becoming unused (i.e., closer to the bottom edge of the video display).

In order for application programs to interface with the accessbar arbiter the accessbar arbiter provides an application program interface (API) 119. This API is facilitated by sending messages between an application program that displays an accessbar and the accessbar arbiter, and by sending notifications from the accessbar arbiter to the application programs having a registered accessbar.

This message and notification sending is performed by the SHAppBarMessage function that is provided by the operating system. One parameter of the SHAppBarMessage function is an identifier of the message or notification being transferred. This parameter contains information pertaining to both messages and notifications. Another parameter is the AppBarData structure, which is depicted in code Table No. 1.

CODE TABLE 1

```
typedef struct AppBarData
{
    DWORD    cbSize;
    HWND     hWnd;
    UINT     uCallbackMessage;
    UINT     uEdge;
    RECT     rc;
    LPARAM  lParam;
} APPBARDATA, *PAPPBARDATA;
```

The "cbSize" element of the AppBarData structure refers to the size of the AppBarData structure. The "hWnd" element of the AppBarData structure is the window handle of the accessbar, which is a reference to the window that displays the accessbar. The "uCallbackMessage" element of

the AppBarData structure is an identifier that the application program specifies and that is returned to the application program when the accessbar arbiter sends a notification to the application program. The "uEdge" element refers to one of the edges of the video display, including bottom, left, right, and top. The "rc" parameter contains the bounding rectangle, specified in screen coordinates, of an accessbar. The bounding rectangle specifies the perimeter or boundary of the accessbar. The use of the "lParam" element depends upon the particular message and is discussed below.

As previously stated, the SHAppBarMessage function has a parameter that specifies the message or notification being transferred. These messages fall into various categories that include: accessbar registration, accessbar position, autohide accessbars and taskbar information. The messages, according to these categories, are depicted below in Table No. 2. These messages and notifications will be described in more detail below.

TABLE NO. 2

```
Accessbar Registration
    ABM_NEW
    ABM_REMOVE
Accessbar Position
    Messages
    ABM_ACTIVATE
    ABM_QUERYPOS
    ABM_SETPOS
    ABM_WINDOWPOSCHANGED
Notifications
    ABN_FULLSCREENAPP
    ABN_POSCHANGED
    ABN_WINDOWARRANGE
Auto-Hide Accessbars
    ABM_GETAUTOHIDEBAR
    ABM_SETAUTOHIDEBAR
Taskbar Information
    Messages
    ABM_GETSTATE
    ABM_GETTASKBARPOS
Notification
    ABN_STATECHANGE
```

The ABM_NEW message is sent from the application program to the accessbar arbiter to register an accessbar and to specify the message identifier that the accessbar arbiter should use when sending notification messages to the accessbar. After receiving this message, the accessbar arbiter stores the data found in the AppBarData structure into a list that it maintains for all registered accessbars. The ABM_REMOVE message is sent by the application program to indicate to the accessbar arbiter that it no longer wishes to receive notifications or perform processing associated with an accessbar. In response to receiving this message, the accessbar arbiter removes the information relating to the indicated accessbar from its internal list.

The ABM_ACTIVATE message is sent by the application program to the accessbar arbiter in response to receiving the WM_ACTIVATE message from the operating system. A WM_ACTIVATE message is received by the application program from the operating system when a user has activated (e.g., clicked on) the window of the application program. The ABM_QUERYPOS message is sent by the application program to the accessbar arbiter to request a proposed location for an accessbar on the video display. The