



**Europäisches Patentamt**  
**European Patent Office**  
**Office européen des brevets**

19

11 Publication number:

**0 121 070**  
**A2**

12

**EUROPEAN PATENT APPLICATION**

21 Application number: 84101729.6

51 Int. Cl.<sup>3</sup>: **G 09 G 3/28**  
**G 06 F 3/147**

22 Date of filing: 20.02.84

30 Priority: 07.03.83 US 472783  
 07.03.83 US 472784  
 07.03.83 US 472776

43 Date of publication of application:  
 10.10.84 Bulletin 84/41

84 Designated Contracting States:  
 DE FR GB

71 Applicant: **International Business Machines Corporation**  
 Old Orchard Road  
 Armonk, N.Y. 10504(US)

72 Inventor: **Emerson, Geoffrey Alan**  
 3 Spaulding Court  
 Saugerties New York 12477(US)

72 Inventor: **Ellis, Joseph J., Jr.**  
 5 Pleasant Ridge Drive  
 West Hurley New York 12498(US)

72 Inventor: **Apperley, Norman**  
 13 Maytree Road  
 Chandlers Ford Eastleigh Hampshire(US)

72 Inventor: **Edwards, Roger Joseph**  
 11 Livingstone Court  
 Woodstock New York 12498(US)

72 Inventor: **Foster, Raymond Lewis James**  
 "Wyedan" Partridge Hill  
 Landford Salisbury Wiltshire(GB)

72 Inventor: **Haigh, David Christopher**  
 11 Shelley Close  
 Winchester Hampshire(GB)

72 Inventor: **Haslam, Michael**  
 31 Keats Close  
 Winchester Hampshire(GB)

72 Inventor: **Verey, Peter**  
 Morland House Salters Lane  
 Winchester Hampshire(GB)

74 Representative: **Grant, Iain Murray**  
 IBM United Kingdom Patent Operations Hursley Park  
 Winchester, Hants, SO21 2JN(GB)

54 Plasma display management systems.

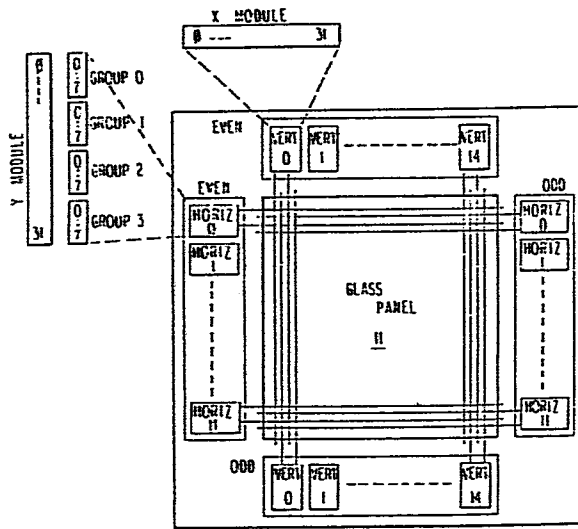
57 A display management system for one or more large plasma gas panel displays (11) is organised to split the display management operations between a host processor (20), a system microprocessor (21) and a picoprocessor (27). The picoprocessor (27) is the heart of a common internal bus (31) plasma display adapter (22). The program for the display is downloaded from the host processor (20) to the system microprocessor (21). The microprocessor generates op codes and initialises display parameters. The plasma display adapter provides control for the plasma panel interface, serialisation of character generator data, translates display position addresses for absolute cartesian coordinates to panel address, and computes the boundaries of display panel write and erase operations. Interaction between the system microprocessor and the plasma display adapter is minimised by a code list contained in the microprocessor memory and fetched by direct memory access. This code list contains high

level commands which the picoprocessor in the adapter decodes and translates into simple commands for the surrounding interface logic.

**EP 0 121 070 A2**

./...

FIG. 7a



0121070

PLASMA DISPLAY MANAGEMENT SYSTEMS

The present invention relates to plasma or gas panel display management systems.

Until recently, the predominant display technology has been the cathode-ray tube (CRT). When flat panel matrix display technologies made their debut, it was expected that this compact-type device would revolutionise the packaging and appearance of display terminals, but this has happened only to a limited extent. There are several matrix-addressed display technologies presently in use: liquid crystals, light-emitting diodes, vacuum fluorescents, a.c. and d.c. plasmas and to a lesser degree a.c. and d.c. electroluminescents. The subject invention is directed to a.c. plasma display technology that provides a large screen, multiple image-format capability. The use of higher information content displays is advantageous for applications requiring the scanning of multiple pages of reference material and for cross-referencing multiple pages or frames of stored information.

The a.c. plasma display technology is a memory technology. Because of this characteristic of the technology, the maximum size, or maximum information content of the screen, is not limited by the device's luminance-voltage characteristic as is the case for refresh display devices, but is limited only by manufacturability considerations. The specific display used is a gas panel having a plurality of horizontal and vertical wires divided into odd and even groups which are physically addressable from opposite edges of the panel. This arrangement allows closer spacing of the wires and the electronic components that generate the driving voltages for the individual wires. The gas panel is an all points addressable device in which the display cells, located between the orthogonal conductor arrays, are individually and selectively addressable. An example of this technology is disclosed in U.S. Patent No. 4,200,868 issued to Lamoureux et al and assigned to the assignee of the subject invention. A specific example of a gas panel which may be

used with the present invention is the model 581 Plasma Display Subassembly available to original equipment manufacturers (OEMs) from the International Business Machines Corporation.

It is an object of the present invention to provide a data management system that permits full utilisation of the advantages of the large screen display afforded by a plasma gas panel without substantially changing basic structure of the user.

These and other objects of the invention are accomplished by providing a plasma display adapter that controls the gas panel, together with a keyboard and other I/O units and a programmable character generator, if and when such are present.

The present invention provides a display management system comprising a host processor adapted in operation to execute application programs, including generating display related programmed symbols and peripheral commands; a system microprocessor, connected as a peripheral unit to the host processor and provided with associated storage to receive from the host processor, and to store, display related commands and programmed symbols, the system microprocessor decoding and executing commands from the host processor including those relating to management of display data; and a display device, interfaced with the system microprocessor, for displaying data as determined by the host processor; characterised in that the display device is a plasma or gas panel display; and the system microprocessor interfaces with the display via a plasma display adapter providing high level control of the native interface of its associated display, together with serialisation of character data and computation of the boundaries of panel write and erase operations.

The plasma display adapter is, preferably designed around a common internal bus and contains a plurality of logic "macro"s, a read-write random access memory (RAM) and a read only storage (ROS). The plasma

display adapter in the described embodiment implements these logic macros with programmable logic arrays (PLAs). The plasma display adapter interfaces with a display system microprocessor having its own associated RAM. The program for the display is downloaded from a local control unit (which is attached to a host system) to the microprocessor memory. System logic, driven by the microprocessor and the adapter, generates all memory addresses and control signals for accessing of memory by the processor during execution of code and by the adapter for transferring data to and from memory. The plasma display adapter has a register mapped I/O control and a picoprocessor which controls data flow to and from a row buffer, the gas panel, and the character generator. Operation of the display is thus split between the host, microprocessor and picoprocessor. The host provides application program operations and downloads programmed symbols. The microprocessor carries out decoding and execution of commands including management of the display data. The plasma display adapter provides a control for the plasma panel interface, serialisation of both character generator and non-coded data, translates display position addresses from absolute cartesian coordinates to panel address, and computes the boundaries of display panel write and erase operations.

Because the display panel is all points addressable and has inherent storage characteristics, updating of the display and partitioning of the display are rather unique when compared to the same functions performed as a conventional CRT display. In the described embodiment, these functions are performed primarily on the plasma display adapter picoprocessor. Since the gas panel cannot generally be updated as quickly as data is changing, unnecessary updates are eliminated by the microprocessor by associating with each row of data in the display buffer an update list containing flags indicating which rows have changed. The replace mode can be used to update characters or insert, delete or scrolling operation. In this mode, all pels within the boundary of the new character are first erased and then selected pels are written based on the chosen character. The microprocessor does

not have to know the screen contents in an update area or specify the erasure of an individual pel. The adapter can write or erase a horizontal stripe the width of the gas panel screen and, in the specific embodiment disclosed, up to sixteen pels high. Thus, for a sixteen pel high character, only one or two erase cycles are used, providing improved performance. The adapter further allows the screen to be partitioned to provide multiple viewing windows, with previously generated data displayed in one window adjacent to updated data in another window or, alternatively, with the simultaneous display of a plurality of data processing sessions on a single screen.

Interaction between the microprocessor and the plasma display adapter is minimised by a code list contained in the microprocessor memory and fetched by direct memory access (DMA). This list consists of high level commands, and while each code is limited to one type of operation, single codes can be linked in the microprocessor memory to provide macro results. The picoprocessor in the adapter decodes these high-level commands and translates them into a sequence of simple commands for the surrounding interface logic.

The present invention will be described further by way of example with reference to an embodiment thereof as illustrated in the accompanying drawings, in which:

FIGURE 1 is a perspective view of a display terminal and keyboard;

FIGURE 2 illustrates some of the possible partitionings of the screen display;

FIGURE 3 shows a specific application where displays are presented in each of four quadrants of the screen;

FIGURE 4 is a block diagram showing the relationship of the keyboard controlled Screen Manager to the host and the plasma display adapter;

FIGURE 5 is a functional block diagram showing the relationship of the plasma display adapter to the system microprocessor and the host;

FIGURE 6 is a more detailed block diagram showing the organisation of the plasma display adapter;

FIGURE 7, in sections A,B and C, is a diagrammatic representation of the gas panel showing the organisation of the X and Y addressing registers;

FIGURE 8, in sections A, B and C, illustrates X and Y addresses as calculated by the picoprocessor and stored in on-chip RAM;

FIGURE 9 is a block end logic diagram illustrating the operation of the serialiser register in the plasma display adapter;

FIGURE 10 illustrates the manner in which horizontal alignment of displayed data is attained;

FIGURE 11 is another diagrammatic representation of the gas panel showing the process of block erasure;

FIGURE 12 is a flow diagram of the replace character operation;

FIGURE 13 is a flow diagram of the variable width character feature;

FIGURE 14 is a block diagram of the use of registers in the plasma display adapter to insert and detect attributes;

FIGURE 15 is a flow diagram of the operation of the high level interface provided by the plasma display adapter;

FIGURE 16 is a block diagram illustrating the addressing of one of sixty-four registers within a plasma display adapter which may be connected to a single system microprocessor; and

FIGURE 17 is a block diagram illustrating attribute testing in the plasma display adapter.

Referring now to the drawings and more particularly to FIGURE 1, the plasma gas panel display is housed in a cabinet 10 that takes advantage of the inherent two dimensional structure of the gas panel 11. Basically, the cabinet 10 has the appearance of a framed panel of modest thickness with a smaller rectangular housing 12 in the back for some of the electronics and power supply. Advantageously, the framed panel may be mounted on a base 13 with the mounting being pivotal in a horizontal axis to permit tilting of the display. A separate keyboard 14 is provided and is typically connected to the display electronics by a cable (not shown).

The most common display applications today are written for 1920 (24x80) character displays typical of CRT displays. The high-capacity a.c. plasma display used in the described embodiment is capable of displaying 9920 characters. To obtain immediate benefits from a high-capacity display of this type, existing applications must be capable of being adapted to the display. Two features have been developed to achieve this objective. They are the display multiple copy screens and display multiple interactive screens. A third feature, multiple partitions, has been developed to allow modified and newly written applications to make full use of the capabilities of the high-capacity plasma panel display. By using a character cell size of 6 x 12 pels, it is possible to accommodate 1920 character "screens" of information simultaneously. Similarly, it is possible to display two



"screens" of other standard sizes either side by side or one above the other. Horizontal and/or vertical divider lines one pel in width are written to differentiate the "screens" displayed. FIGURE 2 illustrates these possibilities.

In the multiple copy screen feature, the user can make copies of one area or "screen" of the display in another without making any modifications to his software. The display is divided into four quadrants with the top left quadrant, for example, being designated as the "active area". This is the only area that the host software is aware of and appears, for example, as a 1920 character display to the host. The remaining areas are used as reference areas. The user can copy the entire display within the active area to any one of the reference area.

In the multiple interactive screen feature, the user can run several applications simultaneously. This feature is illustrated in FIGURE 3. The display is again divided in up to four quadrants, but each of the areas of the plasma panel is an active area. Each area defines a logical terminal and has a different device address. To the host, the plasma display terminal appears to be up to four separate display terminals; thus, the only impact to the host software is to modify tables of device addresses and characteristics. The host may interact with any of the active areas by addressing a data stream to it.

The multiple partitions feature allows a host application to divide the plasma display panel into a collection of up to sixteen non-overlapping rectangular areas. Certain characteristics may then be defined for each of these areas, such as character size, the format of the data streams returned to the host, its position on the display, and whether it is "scrollable".

The described embodiment follows the design philosophy of system network architecture (SNA). SNA freed mainframes and connecting lines

from device dependence so that a common physical link could service multiple applications and multiple device types. SNA defines architectural relationships between logical entities instead of physical devices. This creates an opportunity for produce developers to combine multiple logical entities into a single physical device, which in the case of the described embodiment, is realised with the multiple screen partitioning and multiple data base access just described. The gas panel technology is important to the multiple terminal-multiple data base concept for three reasons. First of all, the fine resolution of the gas panel display allows an unusually high number of characters (in the specific example given, 9920 characters) to appear on the screen at the same time. Secondly, gas panel technology is flicker-free and permits storage of data on the screen in a way that CRT technology has not yet allowed. Finally, all this can be done in an ergonomically enhanced package that fits easily into a user's work environment. Matching the plasma gas panel with SNA capabilities is achieved by means of a screen manager 16 as shown in FIGURE 4. The screen manager 16 communicates with the host interfact 17 and control the display on screen 11 through the plasma display adapter 22. The screen manager 16 allows user control via the keyboard 14 of the functions available from the host. It is the screen manager that allows the user to choose and rearrange screen formats.

The plasma gas display panel driven by the adapter has a more complex interface than had earlier plasma gas panel displays. The specific gas panel used in the preferred embodiment of the invention can write or erase a horizontal stripe the width of the screen (960 pels) and up to sixteen pels high. This permits updating the screen more quickly but at the cost of increased complexity in adapter design. For the particular panel used, panel addressing requires module selection, group selection and specification of the starting pel within a group of modules as will be better understood from the description which follows with reference to FIGURES 7A, 7B and 7C. However, the application program works with absolute X and Y coordinates. The translation

required is accomplished by the plasma display adapter. The design of the adapter is an example in the hardware/software trade-off which resulted in a reduced load on the display system microprocessor and achieve increased performance. The architecture chosen supports this hardware/software trade-off, and the actual circuit implementation preferably uses a Metal-Oxide-Silicon (MOS) technology to achieve high circuit density capabilities.

The interface system is represented by the block diagram of FIGURE 5. The host system 20 downloads display data to the display system microprocessor 21 which includes the microprocessor RAM and ROS, system logic and the microprocessor. The microprocessor preferably supports a sixteen bit address bus, a nine bit data bus (eight bits for data and one bit for parity), interrupt and an I/O interface bus. The system is controlled by logic which is driven by the microprocessor and the attached adapter 22. It is a function-driven according to demand and generates all memory address and control signals for accessing of memory by the processor during the execution code and by the adapter for transferring data to and from memory. To simplify hardware design and to speed up operation, the adapter 22 can access any part of the microprocessor memory via DMA. The adapter is given access to the buses when it requests them, and upon selection by the system logic, the adapter supplies a sixteen bit address, nine bits of data and a read/write control signal.

There is no unique communication between the system microprocessor 21 and the display panel adapter 22. In the preferred embodiment, the adapter 22 is mapped into a portion of the system microprocessor's 21 register space. Therefore, the processor simply executes a register access instruction. Access of the adapter or a typical register is determined by the address used in the register instruction. This scheme is called register mapped I/O (RMIO) and permits addressing more than one adapter although only one such adapter will be shown and described. The microprocessor has the capability of addressing 64K bytes of

register space by using a couple of register indirect instructions. When such an instruction is executed by the processor, a signal is sent to indicate a register operation. As shown in FIGURE 16, the system logic 99 senses this, and based on the sixteen bit address, it selects the appropriate adapter 22 by issuing a signal on an I/O select line which is used only by that adapter. Each adapter has 64 register addresses allocated in the system microprocessors 21 register space for its communication with the system microprocessor 21. The adapter can also initiate communications with the system microprocessor by interrupting at a specified I/O level. Each adapter which has need to store and retrieve information from the system microprocessor memory is interfaced to the system logic via two lines, the DMA request and the DMA select lines. When the adapter has need to access memory, it notifies the system logic via the DMA request line that it wants to access memory. The system logic then, based on priority scheme, issued a DMA select to the adapter and initiates the control signals to the memory.

The display of data is handled through a unique a code list contained in the system microprocessor memory and fetched by DMA. This list consists of high level commands such as "replace character", "load display parameters" and "load character generator". Each op code is limited to one type of operation and therefore is viewed as a "primitive". However, through the use of "chaining", single op codes may be linked in memory to create "macro" results.

As shown in FIGURES 5 and 16, communication between the adapter and the system microprocessor is handled by the RMIO control 23 and system logic 99. This control communicates with the keyboard I/O logic 24, the programmable timer 25, the programmable I/O port 26, and the picoprocessor 27. The high level interface provided by the adapter is made possible by the picoprocessor 27 which executed picocode from an on-chip ROS 32 shown in FIGURE 6. The picoprocessor decodes high-level commands from the system microprocessor and translates them into a

sequence of simple commands for the surrounding interface logic which includes the character generator I/O 28, the plasma panel I/O 29 and the DMA control 30. The picoprocessor can also vary the sequence of interface logic commands and adjust parameters used based on the adapter input parameters. An example of this operation is the sequence of logic for replace mode operations described hereinafter.

As is best shown in the block diagram of FIGURE 6, the picoprocessor 27 is the centre of a common bus architecture. All interface logic macros can both send and receive on the bus 31. The picocode for the picoprocessor 27 is contained in ROS 32, and the picoprocessor 27 communicates with an on-chip RAM 33 via the address bus 34 and data bus 31. In addition, separate control lines exist (not shown here) between the picoprocessor 27 and the interface logic macros. These provide sequence signals and indicate data on bus 31 to be loaded by the interface logic macros. The keyboard interface logic 24 performs a simple "data available", "acknowledge" handshake and an eight bit parallel data transfer. Keystroke data is loaded into a display panel adapter RMIO register and a microprocessor interrupt with keystroke complete status is generated by the adapter. Keystroking in particular and RMIO in general are totally asynchronous to DMA and display update activity. The programmable timer 25 is an eight bit timer whose operation is asynchronous with other adapter functions. The programmable I/O port 26 allows the display system microprocessor 21 to sense or control up to sixteen system external devices through eight input and eight output lines. The display panel adapter provides read (for display and verification), write (for initialisation) and refresh control for a 32,768 by nine bit character generator 100. This generator may contain up to 2048 different symbols, all accessible for display through different data stream and initialisation commands. The adapter supports the panel's unique addressing requirements by translating the binary representation of the display location into an X,Y coordinate driver selection and line selection within the driver.

Control line synchronisation and two bit data serialisation are provided by the display I/O logic 29 and the data stream control serialiser 35.

As stated, the application program or the local terminal intelligence works with absolute X,Y screen coordinates, but the gas panel used requires module selection, group selection and specification of the starting pel within a group or module as illustrated in FIGURES 7A, 7B, 7C. The first area of translation is the Y address which is specified by the system microprocessor 21 as an absolute coordinate. The Y address is loaded into a register in the picoprocessor 27 where it is shifted and rotated until the Y group/module is assembled as represented by FIGURE 8A. This byte is then stored in RAM 33 for later use. The picoprocessor will recalculate this byte only if the Y address changes to a value outside the current group/ module range. It will be observed from FIGURES 7A and 7B that a pair of even and odd Y modules (32 bits) appear to be 64-bits wide to the adapter and there are four sixteen bit groups within each module pair. To efficiently use this 16-bit group, Y start/stop byte is assembled as shown in FIGURE 8B. This specified on which Y line within the group the write or erase will start and on which line it will stop. For single line operations, these two values will be equal. For block erase operations, the adapter takes advantage of the gas panel's ability to erase multiple Y lines within the same group. First, the picoprocessor determines the Y address range of the block erase by adding the height to the current Y value. Then it performs modulo-16 calculations to determine the number of Y group boundaries crossed as indicated by block 42 in the flow diagram of FIGURE 12. If multiple Y groups need to be accessed, then several erase cycles will be required as indicated by blocks 43 and 44 in FIGURE 12. The purpose is to erase as many liens in as few cycles as possible. When multiple accesses are required, the Y group/module byte will be recalculated by the picoprocessor. The following illustrates by way of example a block erase which involves three Y groups.

Block Erase Example

Starting Y = 60 (base 10)

Height = 32 (base 10)

## First Erase Cycle

Y group/module = 1 1 X X 0 0 0 0 Module  $\emptyset$ , Group 3

Y start/stop = 1 1 0 0 1 1 1 1 Start  $12_{10}$ ,  
Stop  $15_{10}$

## Second Erase Cycle

Y group/module = 0 0 X X 0 0 0 1 Module 1, Group  $\emptyset$

Y start/stop = 0 0 0 0 1 1 1 1 Start  $\emptyset$ , Stop  $15_{10}$

## Third Erase Cycle

Y group/module = 0 1 X X 0 0 0 1 Module 1, Group 1

Y start/stop = 0 0 0 0 1 0 1 1 Start  $\emptyset$ , Stop  $11_{10}$

X = Don't Care

At the display I/O logic 29, the Y group/module and start/stop data is transmitted by both serial and parallel means. The Y module data is driven from four parallel output pins. The Y group and start/stop data is clocked out serially in twenty significant bits, sixteen for line selection, and two for odd module group selection.

The X module address calculation is identical to that of the Y module. The result is the X module byte as represented by FIGURE 8C. This data is driven from the same four parallel outputs as the Y module data. The steering of this data to either X or Y address logic at the gas panel is determined by a fifth interface line controlled by the

adapter 22. As is the case with the Y modules, a pair of even and odd X modules (32-bits) appears to the adapter 22 as a 64-bit wide module, as may be appreciated by reference to FIGURES 7A and 7C. The address resolution within the 64-bit X module is provided by padding the display data. This is necessary due to the unique requirements of the interface. If the starting X address specified by the system microprocessor 21 is not exactly divisible by sixty-four (base ten), then a pre-pad of the screen data is necessary. The pre-pad is the number of non-displayable data pels to be shipped serially to the gas panel before valid data begins. It is equal in value to the six least significant X address bits provided to the adapter 22 by system microprocessor 21 and is used to provide proper horizontal alignment of the data. However, the data is clocked out serially two bits at a time, so different boundary conditions will exist throughout the transmission. When highlighting is added on a character basis, this adds to the complexity of the situation. If the character is odd in width, then characters will alternately start on even and odd pel boundaries. This, and the case of starting on an odd X address is handled by steering logic in the data serialiser 35. As shown in more detail in FIGURE 9, the data is either loaded directly or offset by one pel address into the serialiser register 36 by means of steering logic 37. The same logic that controls the steering of data into the data serialiser 36 also maintains flags for mixed character boundaries and mixed starting and ending conditions. This handles the following conditions:

- Start X Address Odd - only the second pel of the two pel data shift is valid.
- Ending X Address Even - only the first pel of the two pel data shift is valid.
- Mixed Character Boundary - the first pel belongs to character N; the second to character N + 1. This is particularly significant if different highlighting is used on each character.



While data is shifting, another counter counts the number of pels transmitted across the interface (modulo-64). For proper horizontal alignment, any X module pair that is accessed must be completely filled with data since these pairs appear to the adapter to be a 64-bit shift register. If valid data runs out before this point, the logic will continue to shift using non-displayable data until the modulo-64 counter has cycled. This excess data is called post-pad. FIGURE 10 illustrates the pre-pad and post-pad with the display data in the 64-bit shift register.

Because the gas panel retains previously written data, a replace mode is used to provide selective, high performance character updates. This permits operations like scrolling, insert and delete. An advantage of this approach is that the display controller does not have to know the screen contents in the update area or specify the erasure of individual pels. The plasma display adapter handles the update operation by performing a high speed erase and all pels within the boundary of the new character(s) before writing the appropriate pels from the character generator. As previously described, the high speed erase utilises a feature of the gas panel that allows erasure of up to sixteen scan lines in a single erase cycle rather than one scan line per erase cycle. This permits a sixteen pel high character to be erased in one or at most two erase cycles as compared to sixteen erase cycles using single scan line erasure techniques. Again, this function is performed by the picoprocessor 27,

When a replace operation is detected, the current Y location is saved in RAM 33. This is necessary because each scan line will be accessed twice, once for the block erase and once for the draw operation. The picoprocessor 27 then adds the character height to the starting Y value to determine the Y dimension of the block erase. With reference now to FIGURE 11, the gas panel's single cycle, sixteen line (one group) erase is limited to fixed modulo-16 boundaries. But the range of lines to be erased may exceed sixteen and in most cases will

not start on one of the modulo-16 boundaries. The picoprocessor 7 resolves this by performing modulo-16 arithmetic to determine the number of Y groups accessed as indicated by block 42 in the flow diagram of FIGURE 12. A group boundary is crossed as between lines 15 and 16 and lines 31 and 32 in the example illustrated in FIGURE 11, so the picoprocessor must determine the proper starting location in the first group and the ending location in the last group. These groups will then be accessed on separate erase cycles until the block erase is complete.

The logic paths in the plasma display adapter are the same for both the erase and the draw portions of the operation. The replace character operation is set forth in the flow diagram of FIGURE 12. The first step in the operation is to detect the replace character op code as indicated in block 40. The picoprocessor 27 then calculates the ending Y address by adding the character height to the current Y address and saves the current Y address in RAM 33 as indicated by block 41. The picoprocessor then calculates the number of Y groups accessed in block 42 and sets the block erase flag in block 43. The flag forces all display data (FIGURE 10) to "ones" so when the panel 11 erase command is issued, all pels within the range will be erased. Also in block 43, the display logic 29 is started with a range of Y addresses not exceeding sixteen to erase. Then in decision block 44 the picoprocessor determines whether there are remaining Y groups to be accessed. If so, the display logic is again started with a range of Y addresses; otherwise, the block erase is complete in which case the picoprocessor 27 resets the block erase flag and restores the original Y address as indicated in block 45. Then in block 46 the picoprocessor starts the draw operation.

The plasma display adapter makes possible multiwidth character display. The nominal width-height ratio for the display of alphanumeric data is 9x16 pels. A typical character displayed uses only seven of the nine horizontal pels for information. The other two pels are used for spacing or the creation of "box" in which the information bits will reside. Because of the finer resolution exhibited by the gas panel,

smaller character boxes are possible without sacrificing "readability". Also, as pel densities increase, the number of bits displayed per character must also increase to maintain the original 9x16 aspect ratio. This is handled in the adapter by allowing the system microprocessor 21 to specify any box width between four and thirty-one pels for both characters and non-coded information (NCI). In the case of characters, the character generator RAM 100 holds 9 pels of horizontal information. For widths less than nine, this information is truncated to the specified width. For widths greater than nine, the information is padded with additional pels to the right of the character. These pels follow the highlighting of the character box (i.e. for normal highlighting, blank pels are inserted, but for reverse highlighting, pels that are lighted are inserted).

With reference now to FIGURE 13, the plasma display adapter fetches parallel data (8 bits) from either the character generator 28 for coded data or from RAM 33 for NCI as indicated in block 50. The data is then serialised in serialiser 35 for transmission to the gas panel as indicated in block 51. The variable width feature is implemented by designing the serialiser 35 around a byte (eight bit) wide data bus. The logic resolves the five bit width field from the system microprocessor 21 into a two bit modulo-8 count that determines how many times the serialiser 35 will iterate. Data is loaded into the serialiser eight bits at a time. When the serialisation is complete, the two bit count is checked. If it is not zero, then it is decremented, more data is loaded in block 55 and another pass through the serialiser begins as indicated by decision block 52. This continues until the count equals zero. Also, portions of the data load may be suppressed if loading the full eight data bits would cause the specified width to be exceeded as indicated by blocks 53 and 54 and by block 56. Widths less than nine pels can be used on the gas panel to provide a compressed character display. Widths greater than nine pels can be used to insert additional inter-character spacing if all nine of the character generator bits are used for display information. The latter

case may be used to create an enlarged character display or to maintain the current aspect ratio on higher density displays. It should be noted, however, that to maintain this aspect ratio with widths greater than nine pels, heights greater than sixteen pels must be generated. The plasma display adapter supports heights of one to 255 scan lines. "Pad" scan lines that follow the character by highlighting are automatically inserted beyond scan line sixteen.

In handling a field oriented data stream, the situation can arise where a display update is required within a field but a complete field rewrite is not desired. An example of this is where a character is to be inserted on a line, simply by writing the new character and rewriting the now shifted characters to the right of it. However, in the data stream used in the preferred embodiment of the invention, field qualifiers exist which specify the highlighting, colour, character generator font, intensity and display/non-display of all characters in their field. Some of these can be over-ridden and others cannot. In the case where the individual character attribute specifies a default to the field, these field parameters must be present. The plasma panel 11 provides a unique challenge over prior art CRT's. CRT's provide a sequential raster refresh of the display. As a memory device, the plasma panel can be used in random-access mode. It is in just this mode where the situation described here arises. In order to provide specification of these field parameters without rewriting the entire field, the plasma display adapter can interpret an artificial initialisation attribute. With reference to FIGURE 14, in the normal processing of a character row, the adapter reads a current data stream character into one of its registers 60 and updates another register 61 if an extended and/or field attribute is detected. Field information detected in this way is normally used for subsequent characters until the next field information is detected. However, the system microprocessor 21 may selectively write this register 61 before starting the adapter 22 on a screen update operation. This means that a character or characters may be inserted in mid-field simply by writing

the proper field attributes into the adapter registers. While conventional field attributes use a position on the display, these register based attributes require no such location. In other words, access by the system microprocessor to the adapter registers provides the ability to specify a field attribute outside the data stream. The key to this operation is steering logic 62 that sends the attribute information from register 61 to the highlighting logic 63 for the first character of every row. This will be over-ridden, however, if the first character position on the row contains a field attribute (in this case register 60 is directed to the highlighting logic 63).

There are certain data stream handling requirements that can easily overburden the system microprocessor, particularly with a data stream with both field and character attributes. The problem becomes more acute with the gas panel display environment because the display adapter cannot automatically handle blinking and underscoring of characters and fields. Therefore, the microcode must be aware of all blink and underscore locations and handle them separately. This could be done with a search through the display buffer, performing tests along the way to detect the different attributes. But this approach does not have very good performance due to the fetch and test loop required for each character in the buffer. This is compounded with the large screen display panel used because the display buffer can be as large as 10,000 characters (20,000 bytes). Another approach would be for the microcode to build and maintain an attribute position list. This has problems of performance, storage required and complexity associated with it.

This problem is solved by performing the attribute testing in the plasma display adapter 22 by checking the data while it is being loaded into the adapter by DMA. After the system microprocessor 21 requests a character row write, it can read adapter register 101 (FIGURE 17), to determine if further screen update is required as, for example, underscore. The microcode can also read back the field characteristics that were active at the end of the row (e.g., non-display) through the

field and extended field attribute detect registers 61 in the adapter. This field information will be used on the next row unless the first character of that row is a field attribute. The following attributes and characters are tested for:

- Any NULL Characters in the Display Data
- Any Blinking Characters in the Display Data
- Any underscore Characters in the Display Data
- Any Field Attributes in the Display Data
- Any Field Intensify Attributes in the Display Data
- The Last Field Attribute Detected
- The Last Extended Field Attribute Detected

The operation of the data management system just described will be briefly summarised with reference to FIGURE 15. In block 70, the system microprocessor 21 creates adapter op codes in system microprocessor RAM, initialises parameters in the adapter and issues the command to start to the adapter 22. The adapter 22 then fetches the op code from the microprocessor RAM by DMA and decodes as indicated in block 71. The adapter 22 next fetches display data and stores it in RAM 33 in block 72. Once the op code and display data have been fetched, the picoprocessor 37 calculates display parameters and initialises the display I/O logic 29 (FIGURE 6) as indicated in block 73. Based on these calculations, the line buffer addresses and character generator RAM 100, data is serialised and highlighted in serialiser 35 and then displayed as indicated in block 74. Once the data in serialiser 35 has been outputted by the display I/O logic 29 and displayed on the plasma panel, the display I/O logic 29 flags completion to the picoprocessor 27 as indicated on block 75. The clean-up performed by picoprocessor 27 returns the adapter 22 to a base state, ready to execute additional op codes. In decision block 76, if a chained op code is being executed, the operation return to block 71; otherwise, the adapter interrupts the system microprocessor 21 with completion status and the operation stops as indicated in block 77.

## CLAIMS

1. A display management system comprising

a host processor (20) adapted in operation to execute application programs, including generating display related programmed symbols and peripheral commands;

a system microprocessor (21), connected as a peripheral unit to the host processor and provided with associated storage to receive from the host processor, and to store, display related commands and programmed symbols, the system microprocessor decoding and executing commands from the host processor including those relating to management of display data; and

a display device (10), interfaced with the system microprocessor, for displaying data as determined by the host processor;

characterised in that

the display device is a plasma or gas panel display (11); and

the system microprocessor interfaces with the display via a plasma display adapter (22) providing high level control of the native interface (29) of its associated display, together with serialisation of character data and computation of the boundaries of panel write and erase operations.

2. A system as claimed in claim 1, wherein the adapter includes a picoprocessor (27), having direct memory access to the microprocessor storage, the microprocessor compiling, in response to display related symbols from the host processor, a picocode list defining the current sequence of operations to be performed by the picoprocessor and hence by

the adapter as a whole, such picocode being immediately executable by the picoprocessor.

3. A system as claimed in claim 2, wherein the picocode list includes picoprocessor executable op codes and initialising display parameters,

the display adapter having a common internal bus(31), to which common internal bus are connected

the picoprocessor, it being adapted to calculate the required boundaries,

adapter memory means (26) for storing the picocode accessed from the microprocessor memory, together with the results of calculations performed by the picoprocessor in response thereto,

control means (30) for fetching the op codes and display parameters from the microprocessor memory by direct memory access and storing the op codes and display par-meters in the adapter memory , and

character generator and serialiser (28) for serialising character data for transmission to the attached display according to the boundaries calculated by the picoprocessor.

4. A system as claimed in claim 3, wherein the character generator includes registers (61, for example), accessible by the system microprocessor for storing afield attribute for insertion into the data stream formatted by the character generator.

5. A system as claimed in claim, in which the adapter includes attribute detect means (62) for detecting and storing a field attribute in the display data stream, such field attribute being used for the next display row unless the first character of that row is, itself, a field attribute.



6. A system as claimed in any preceding claim, wherein the adapter is arranged to translate display position addresses from absolute cartesian coordinates specified by system microprocessor to panel addresses.

7. A system as claimed in claim 6, wherein the display is of the type having a plurality of horizontal and vertical wires divided into odd and even groups which are physically addressable from opposite edges of the panel (see Fig. 7), each group having the same number of wires, the groups being aggregated into modules of a constant group size, absolute cartesian coordinates being translated into module, or module and group identities, for vertical and horizontal wires, respectively.

8. A system as claimed in claim 7, wherein the groups of wires of each module are selectable by means of the currently loaded state of a shift register (see Fig. 7) of the kind which is loaded serially by pairs of bits, corresponding to an even and an odd group of wires, but read in parallel, the registers corresponding to one wire orientation being loaded before those corresponding to the other wire orientation.

9. A system as claimed in claim 8, wherein for writing, as opposed to erasing, the shift registers of at least one orientation of wires are pre- and/or post-padded to obtain correct display element alignment.

10. A system as claimed in any of claims 7, 8 or 9, wherein the number of groups between the start and the finish of an erase operation is calculated.

11. A system as claimed in claim 9, wherein the adapter includes a character generator (28) and an associated character memory (100) storing character information of a predetermined pel width, the system processor being capable of specifying character widths greater or less than the stored predetermined pel width, the picoprocessor calculating the position of data to be displayed within the module specified for such display, the picoprocessor being responsive to the character width

specified by the system microprocessor to cause the character generator to, either, truncate the character information from the character generator memory if the specified width is less than the specified pel width, or to pad the character information from the character generator memory with additional pels if the specified character width is greater than the predetermined pel width, for insertion into the wire selecting shift registers.

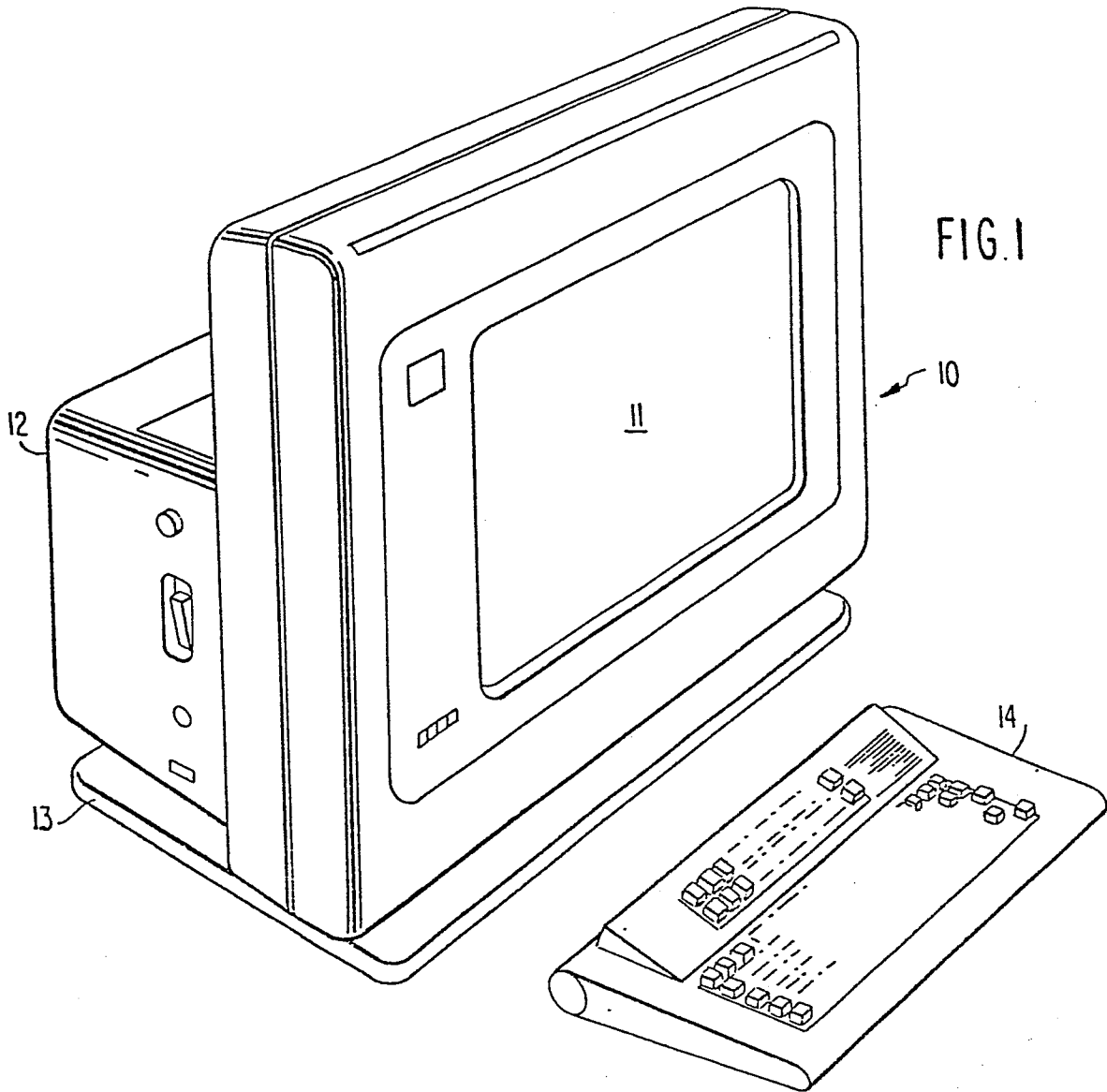


FIG. 2

1920 (24 x 80)	1920 (24 x 80)
1920 (24 x 80)	1920 (24 x 80)

1920 (24 x 80) OR 2520 (32 x 80) OR 3440 (43 x 80)	SAME AS AT LEFT
---	--------------------

1920 (24 x 80) OR 3514 (27 x 132)
SAME AS ABOVE

FIG. 3

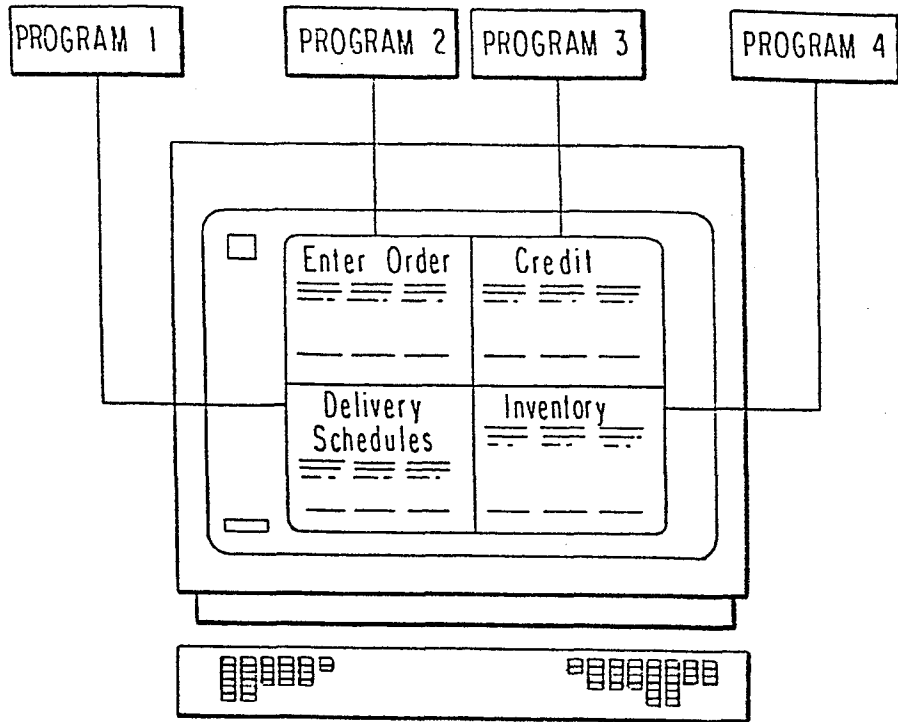


FIG. 4

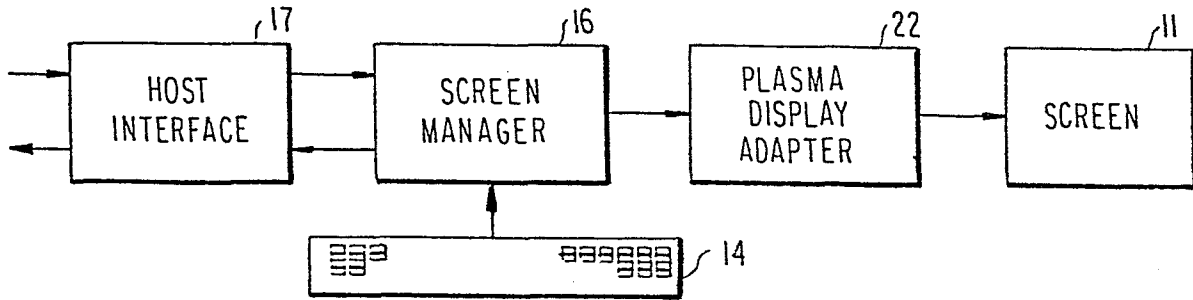


FIG. 5

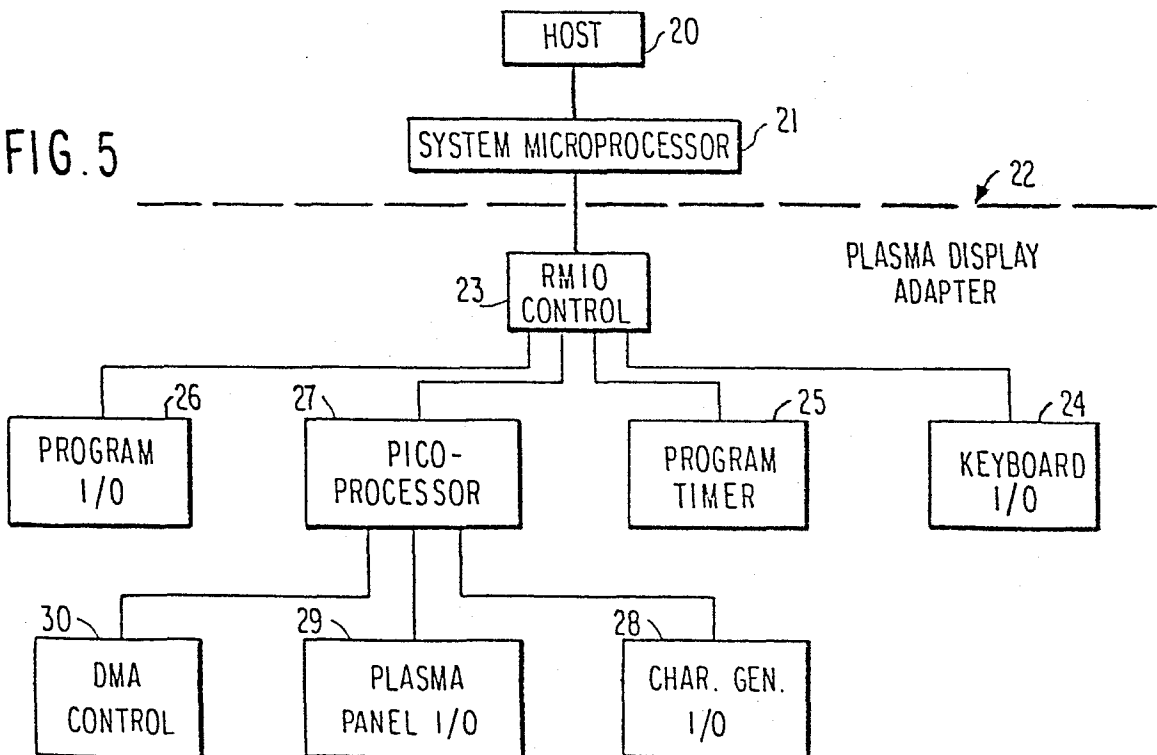
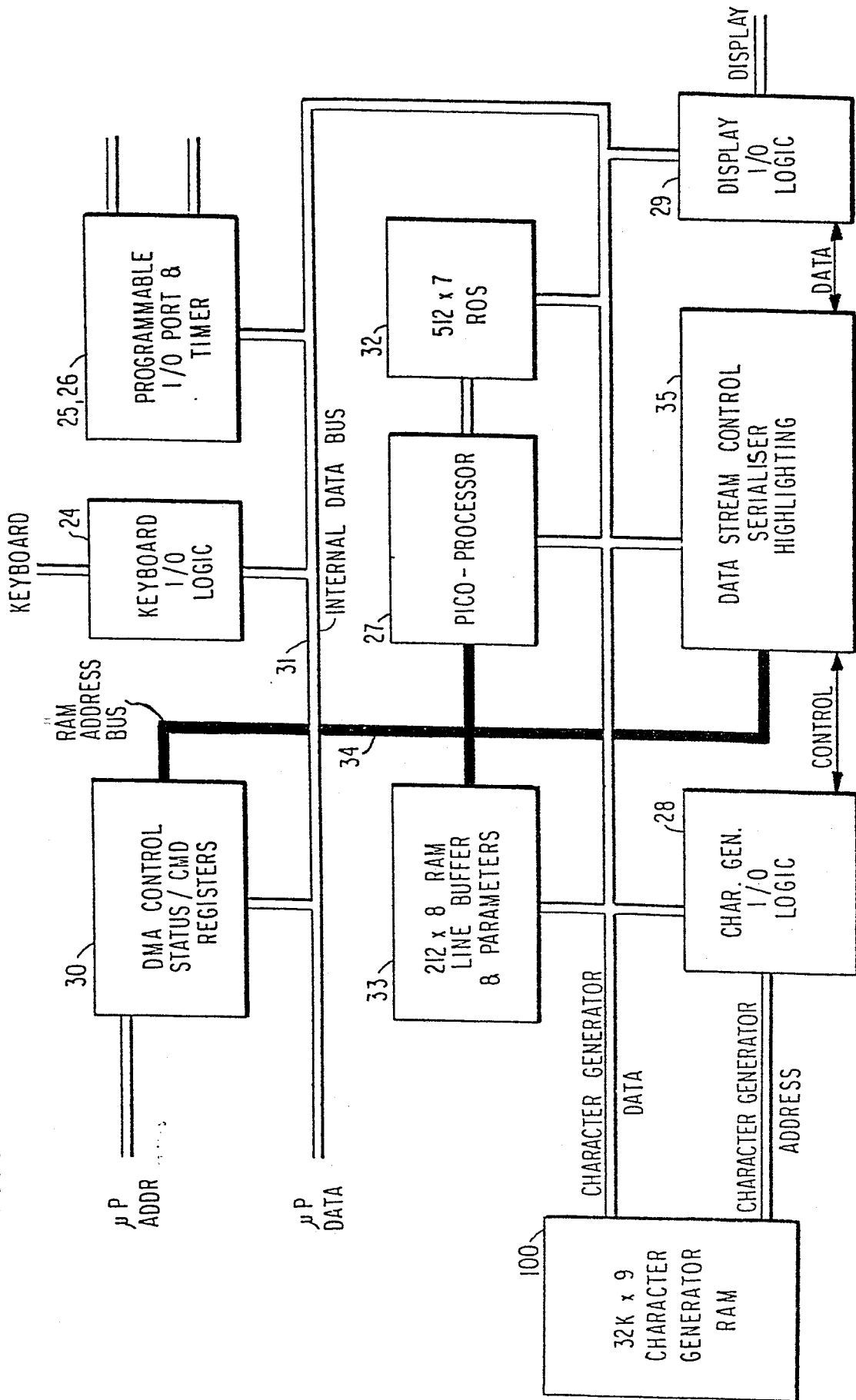


FIG. 6



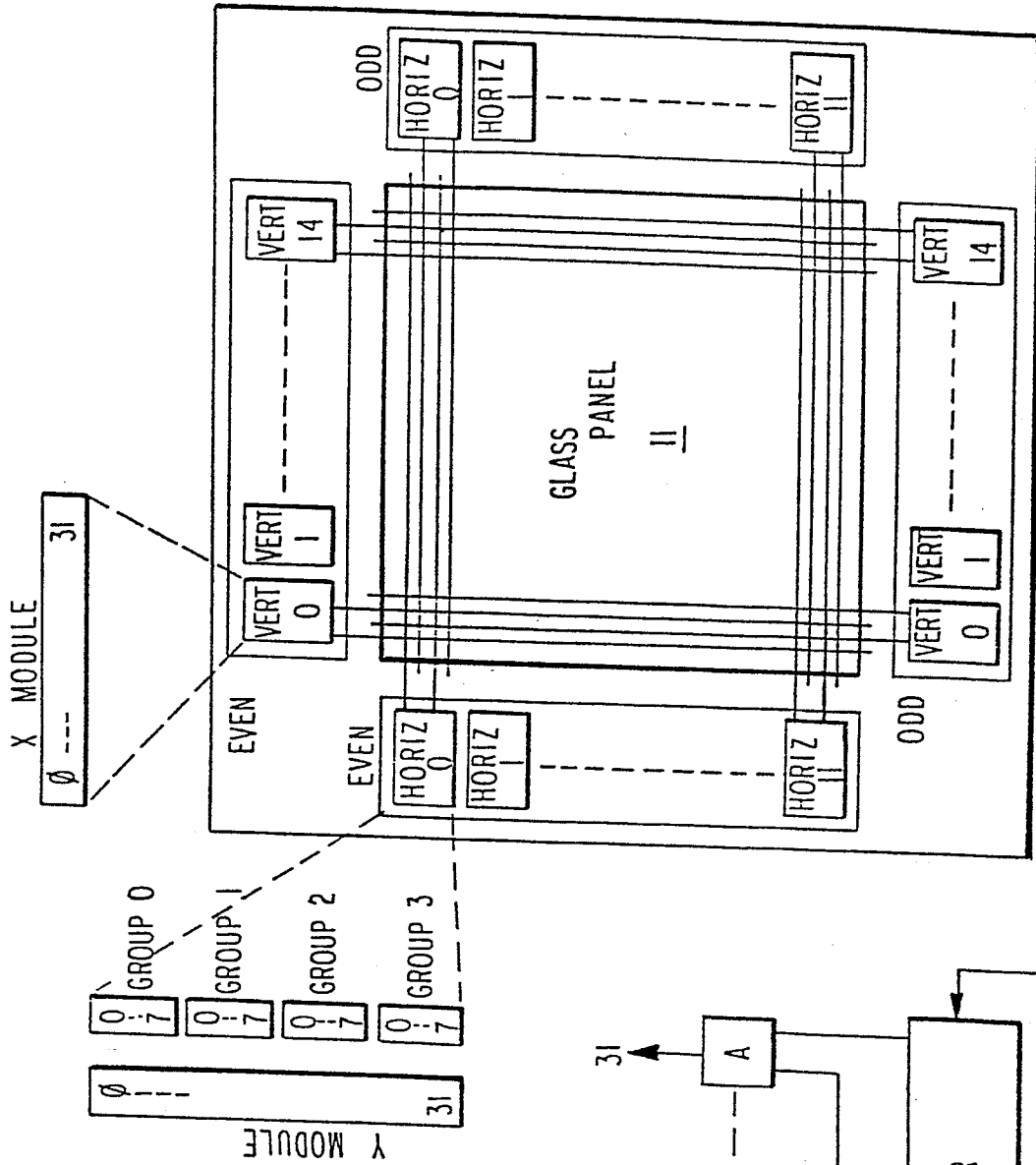
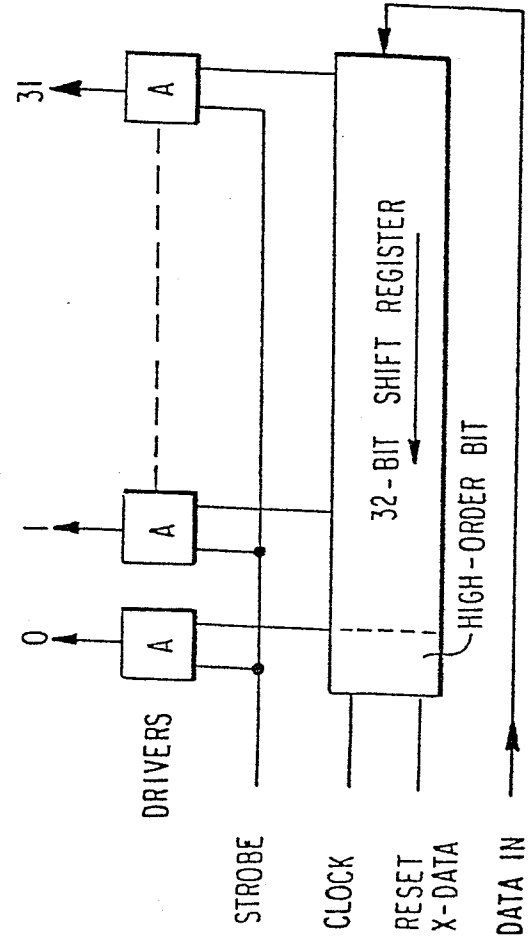
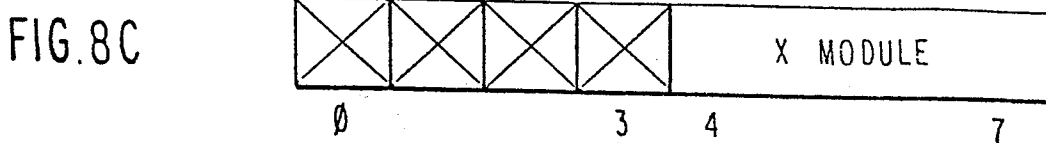
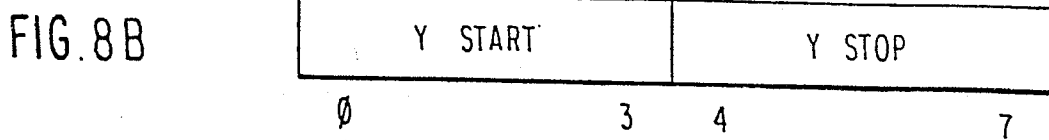
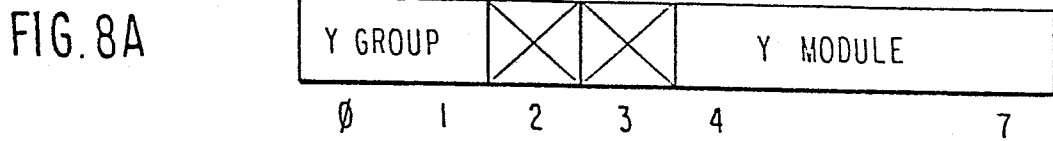
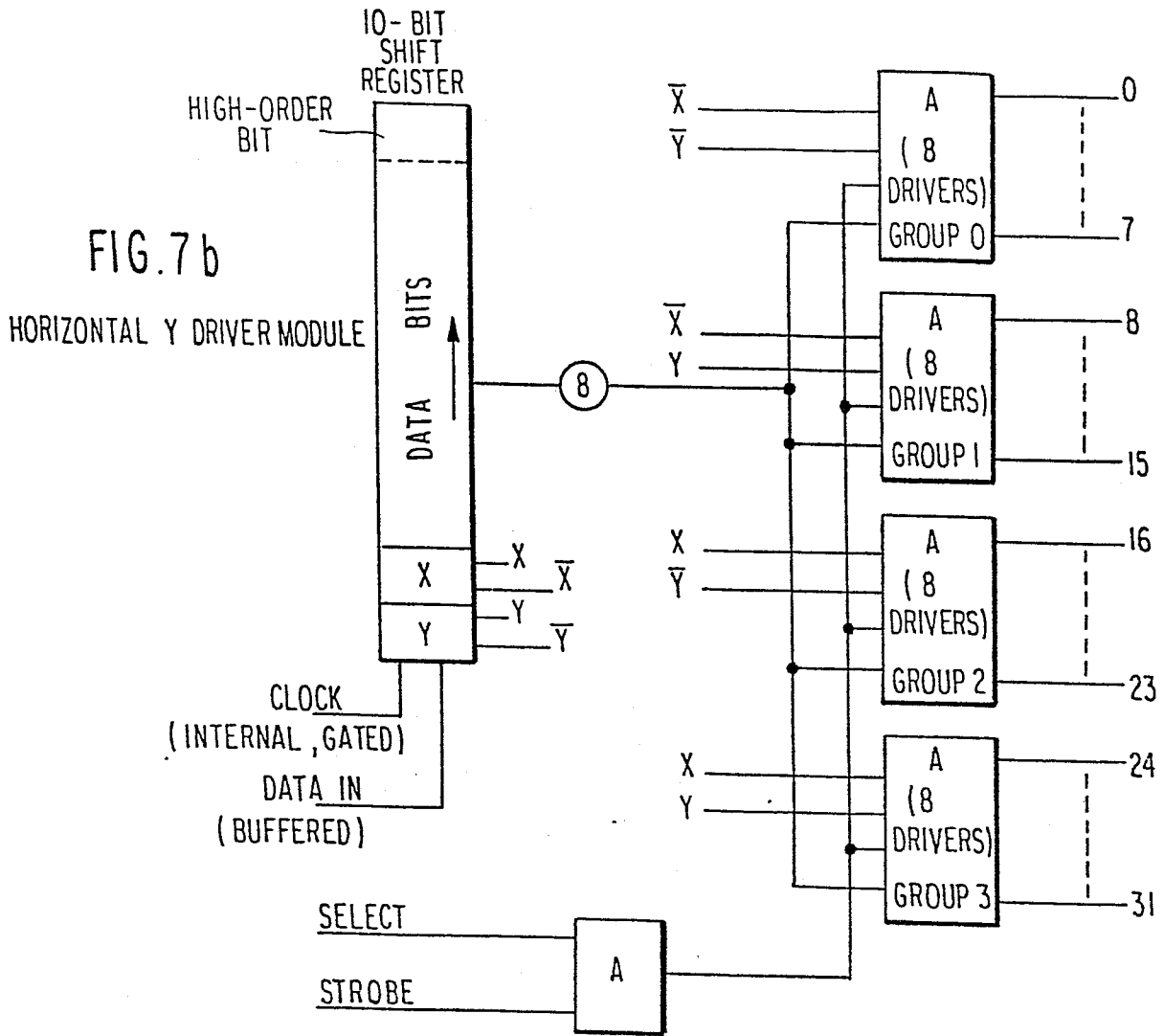


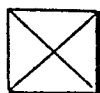
FIG. 7a

FIG. 7c  
VERTICAL X DRIVER MODULE



DRIVERS  
STROBE  
CLOCK  
RESET  
X-DATA  
DATA IN



 DON'T CARE

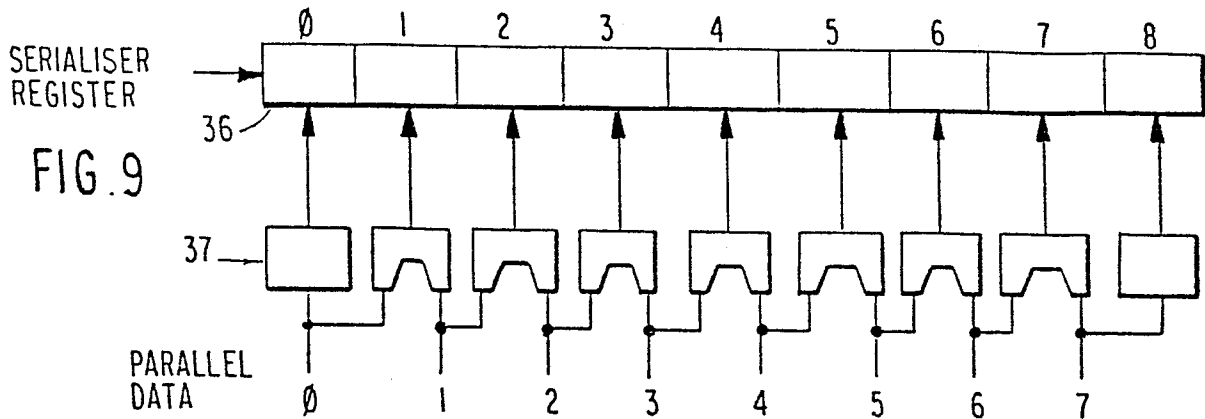


FIG. 9

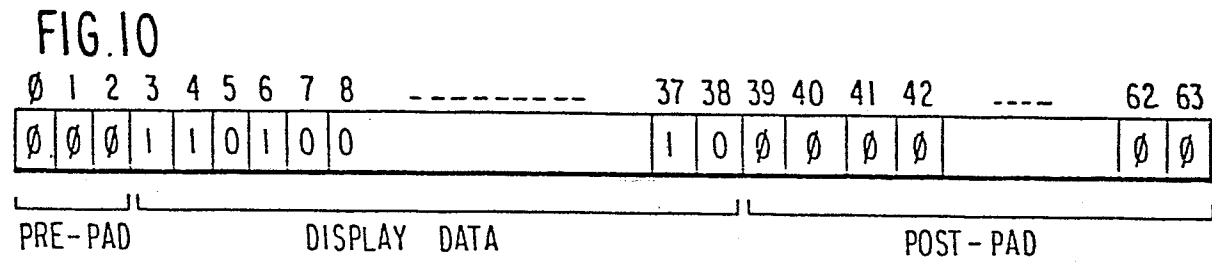


FIG. 10

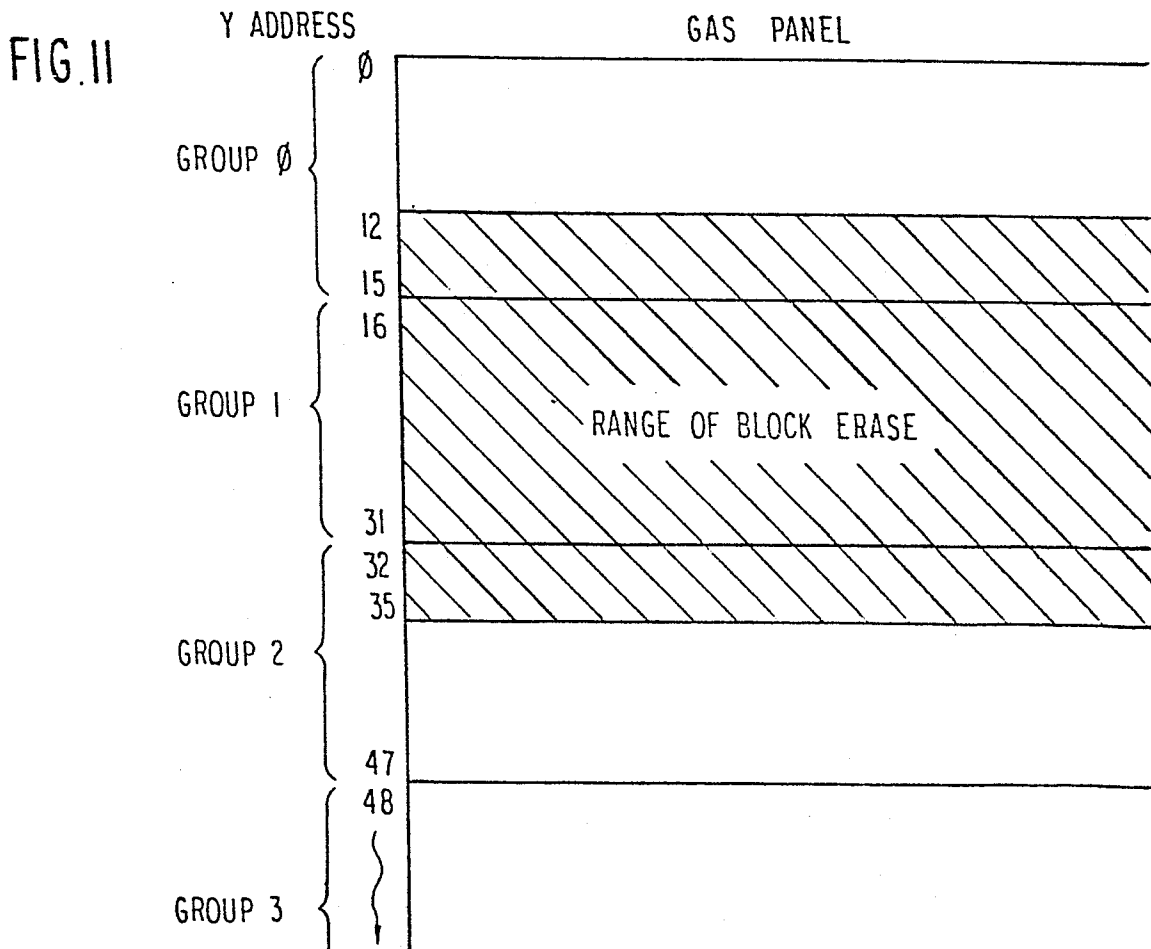


FIG. 11



7/9

FIG. 12

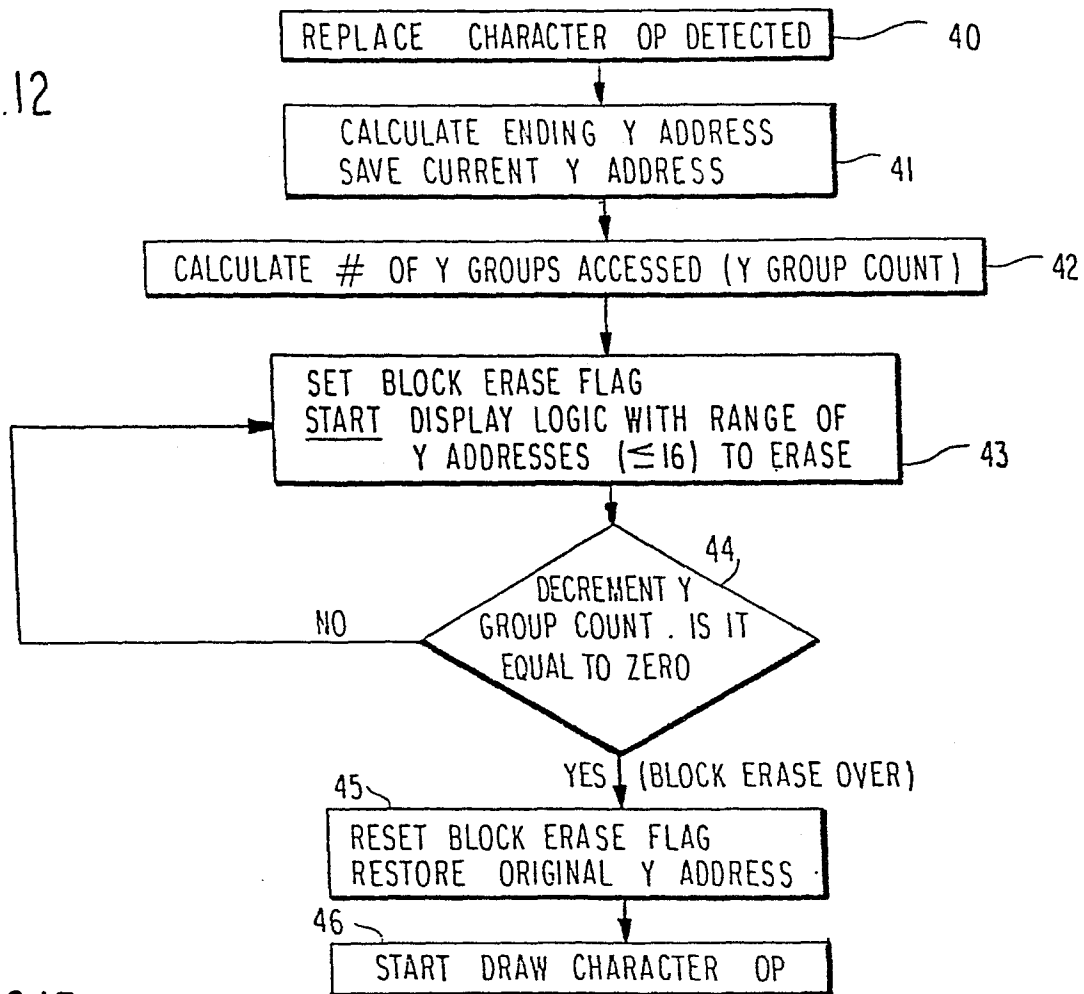
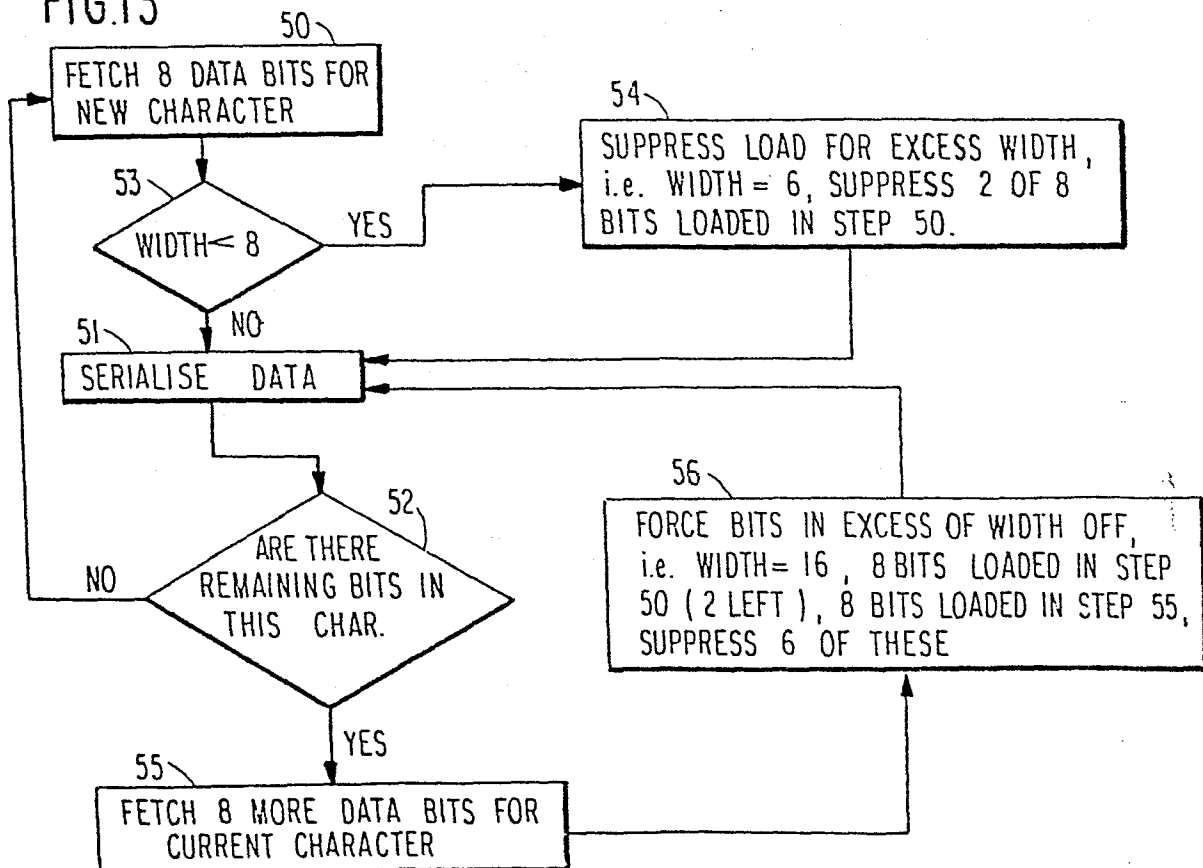


FIG. 13



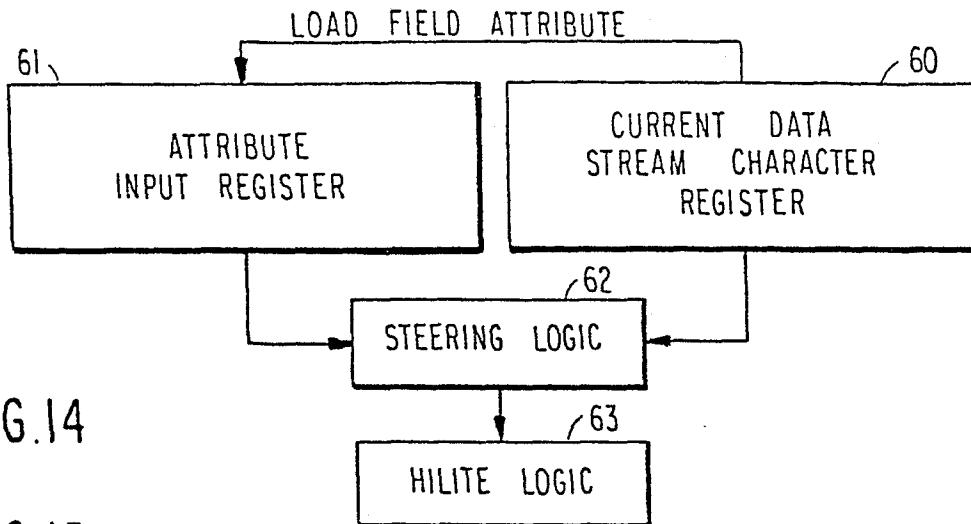


FIG.14

FIG.15

