

12 **EUROPEAN PATENT SPECIFICATION**

45 Date of publication of patent specification: **06.08.86**

51 Int. Cl.<sup>4</sup>: **G 09 G 1/16**

21 Application number: **82101013.9**

22 Date of filing: **11.02.82**

54 **Display system with multiple scrolling regions.**

30 Priority: **23.02.81 US 237316**  
**23.02.81 US 237318**

43 Date of publication of application:  
**08.09.82 Bulletin 82/36**

45 Publication of the grant of the patent:  
**06.08.86 Bulletin 86/32**

84 Designated Contracting States:  
**DE FR GB NL**

56 References cited:  
**EP-A-0 004 554**  
**US-A-4 075 620**  
**US-A-4 126 814**

**IBM TECHNICAL DISCLOSURE BULLETIN, vol.**  
**23, no. 6, November 1980, pages 2507-2508,**  
**New York (USA); H.E.BUSH et al.: "Data-**  
**display terminal buffer addressing for scrolling**  
**operations"**

**IBM TECHNICAL DISCLOSURE BULLETIN, vol.**  
**22, no. 11, April 1980, pages 4809-4810, New**  
**York (USA); K.R.DEMKE et al.: "Line attribute**  
**control"**

73 Proprietor: **TEXAS INSTRUMENTS**  
**INCORPORATED**  
**13500 North Central Expressway**  
**Dallas Texas 75265 (US)**

72 Inventor: **Watts, LaVaughn F.**  
**10410 Autumn Harvest**  
**Houston Texas 77064 (US)**  
Inventor: **Smith, Ronald L.**  
**7058 Shady Arbor**  
**Houston Texas 77040 (US)**  
Inventor: **Pandya, Yogendra C.**  
**12810 Grand Cross**  
**Houston Texas 77072 (US)**  
Inventor: **Wood, Paul B.**  
**9205 Statecoach**  
**Houston Texas 77041 (US)**

74 Representative: **Leiser, Gottfried, Dipl.-Ing. et al**  
**Patentanwälte Prinz, Bunke & Partner**  
**Ernsberger Strasse 19**  
**D-8000 München 60 (DE)**

58 References cited:  
**IBM TECHNICAL DISCLOSURE BULLETIN, vol.**  
**23, no. 4, September 1980, pages 1512-1513,**  
**New York (USA); D.A.STOCKWELL: "Display**  
**with partitioned slow scroll"**

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

**Description****Background of the Invention**

This invention relates generally to data display system as defined in the precharacterising part of claim

5 1.

Traditional region separation has been performed using large scale computers. In such cases, data to be placed into a desired region was computed by software resident within the main computer. The data was then transmitted to a display device as a new display frame. Heretofore, data in display systems were not organized for display in terms of regions or windows into a data block. Further, the segmentation of the display into vertical and horizontal regions was performed by the large computer in software, and not at the local display device. The prior art system could not provide smooth or pixel scrolling of a vertical split region because of the immense software and processing time requirements.

In accordance with the prior art — as disclosed in IBM Technical Disclosure Bulletin, vol. 23, No. 6, November, 1980, pages 2507—2508, H.E. Bush et al, a data display system has been advanced for forming a plurality of independently controllable display regions on a display device. This system has a data processor connected to a main memory. An input is connected to the data processor connected to the main memory display, and a data controller is connected to the data processor display memory wherein each row of characters to be displayed on the display device is assigned one or more memory location identifiers as defined by the display controller to indicate the display memory address of a subsequent row of characters to be displayed and wherein the display controller alters the sequence of rows displayed on the display device by modifying the memory location identifiers of the selected rows of characters.

It is an object of this invention to provide a display control system which permits the smooth or discrete scrolling of a plurality of horizontal and vertical split display regions.

Another object is to provide a system for controlling a display memory which minimizes the processing load on a system CPU.

Still another object is to provide a system for mapping the contents of a display memory without requiring the transfer of blocks of data therein during various editing and scroll functions.

Yet another object is to provide a display control system which may be formatted either locally or from a remote host computer.

Another object is to provide the system microprocessor with executable instructions at a high access rate, and to provide the microprocessor with instructions through a directly accessed memory whose capacity is less than the list of program instructions.

**Summary of the Invention**

In the data display system including the features of the characterising part of claim 1, which realize the above-mentioned objects, a microprocessor controls the system operation in accordance with the list of instructions stored in a random access main memory. The main memory stores a portion of the list of instructions, while the complete list of instructions is stored in a mass data storage memory in the form of one or more GROMs. An additional ROM contains a list of instructions for initializing the system and a non-volatile memory in the form of a CMOS RAM powered by a battery is included to store system configuration parameters in the event of a power loss.

A display device such as a CRT is controlled by a display control system formed of a CRT controller, a visual attribute generator, a display memory, and a character ROM. Alternate character sets are provided by a character RAM which is accessed by the display controller through the attribute generator. The display controller accesses the display memory asynchronously with the system microprocessor, thus decreasing the processing time and increasing the processor throughput. Segmentation of the display into multiple horizontal and vertical split regions which are independently scrollable either discretely or smoothly is provided by the linking of one or more row attribute bytes or "pointers" to each character and character attribute row stored in the display memory. In this manner, data can be manipulated on the display screen without changing the absolute address of the data in display memory. That is, only the pointers associated with each block of data (corresponding to a displayed row) needs to be modified by the display controller. Processing time is greatly reduced because whole blocks of data need not be transferred in the display memory, only the display memory addresses associated with the pointers are affected.

For network applications, input/output means are provided for communicating with a controller or "host" computer. For local control of the system, input means in the form of a keyboard are provided. In one embodiment, the keyboard is a self-contained unit having a dedicated microcomputer. The keyboard unit communicates with the system through a keyboard interface unit that converts the serial data into a parallel form which is processed by the system microprocessor. Discrete control of the display device contrast is achieved by a contrast register in conjunction with the attribute generator.

60

**Brief Description of the Drawings**

Other objects and many of the attendant advantages of the present invention will be readily apparent as the invention becomes better understood by reference to the following detailed description with the appended claims, when considered in conjunction with the accompanying drawings, wherein:

65

Figures 1a—1c form a block diagram of an electronic terminal including a display controller system according to the present invention;

Figure 2 is a system memory map;

Figure 3 is a diagram of the bit structure of an address byte used to access the system mass read-only program storage memory;

Figures 4 and 5 are block diagrams of the GROM interface unit;

Figure 6 is a pictorial representation showing the use of form descriptors and region data tables in establishing a display screen format;

Figure 7 shows the use of display memory mapping to scroll a selected region on a display screen segmented into a plurality of regions;

Figures 8a—8c form a block diagram of the display controller;

Figure 9 is a schematic diagram of the self-contained keyboard unit;

Figure 10 is a schematic diagram of the keyboard interface unit;

Figure 10a is a timing diagram showing the relationship of the signals in the keyboard interface unit.

#### Detailed Description of Invention

##### System Architecture

Referring now to Figures 1a—1c there is shown a block diagram of an electronic input/output terminal according to the present invention. A system microprocessor CPU 10 is coupled to a system control bus 12 and a system address bus 14. Control bus 12 and address bus 14 both extend to a counter time circuit CTC 18, a dual asynchronous transmitter/receiver DART 20, a serial input/output device SIO 22, a dynamic random-access memory (DRAM) timer 24 and a memory I/O decoder 26. System address 14 also extends to a system read-only memory (ROM) 27, a DRAM address multiplexer 28, a test ROM 29, a CMOS RAM 30, and a control I/O device 32. A battery 31 is coupled to CMOS RAM 30 to provide backup during power off conditions. CPU 10 is connected through a two-way buffer for transceiver (XCVR) 34 to a system data bus 36. Data bus 36 extends to a contrast register 38, and to CRTC 16, CRTC 18, DART 20, and SIO 22. Data bus 36 is connected through a transceiver 40 to a memory data bus 42 which in turn extends to system ROM 27 and a series of DRAMs 48a—48c. Each DRAM is also connected to DRAM timer 24 and to a DRAM address bus 50 which is connected to a DRAM multiplexer 28. Data bus 36 is also connected through a XCVR 44 to an I/O data bus 46 which in turn extends to test ROM 29, CMOS RAM 30, and control I/O 32.

Program storage is provided by large scale mass read-only memory in the form of GROMs 52—1 to 52—8. The GROMs are connected to a common GROM bus 54 which is in turn connected to I/O data bus 46 through a GROM I/F 56.

An operator interfaces with the present system by means of a keyboard 58 which is connected to a keyboard I/F 60 through a telephone type multiconductor cable 62. Keyboard 58 contains its own computer 64 which permits keyboard 58 to process instructions with a minimum computing load on CPU 10.

A series of test multiplexers 66, 68, and 70, are connected respectively to DART 20, and channels B and A of SIO 22, to provide loop back testing of the terminal in a local operating mode. A set of communication buses 72, 74, and 76, are connected to control I/O 32 and, respectively, to test multiplexers 66, 68, and 70. Communication to peripheral devices such as printers is provided from communication buses 72 and 74 through EIA driver-receivers 78 and 80, which are connected respectively to auxiliary ports AUX 2 and AUX 1. Communication with a host computer is provided by communication bus 76 which is connected through a PM interface 82 and an EIA driver-receiver 84 to an input/output port labeled HOST in the drawing. Longer distance communication is provided by means of an internal modulator-demodulator (MODEM) 85 which is also connected to communication bus 76. PM I/F 82 may be connected to a "personality module" (not shown) which permits the terminal communications to be compatible with various host computers. A baud rate multiplexer 86 is connected to communication bus 76 and to control I/O 32 to provide synchronization and a selection of baud rates with the host computer. The timing of baud multiplexer 86 is provided by timer 18—0 of CTC 18. Baud rate timing for the AUX 2 and AUX 1 ports is provided by timers 18—2 and 18—1 of CTC 18. Timer 18—3 functions as a system timer.

A communications option unit 87, shown by dashed lines in Figure 1c, is connected to the HOST port, the system address bus 14 and through an option I/F 88 to communication bus 76. Unit 87 is used to provide local and current-loop communications by methods known in the art, and need not be described in detail here.

The present terminal communicates with an operator by video display means such as a CRT (not shown) which is driven by a controller CRTC 16. The generation of characters with various character attributes will be described in more detail below. Briefly, however, a set of characters is stored in a character ROM 90 connected to CRTC 16 by a character bus 92. The character, along with selected visual "attributes" generated by a video attribute generator VAG 94 are stored in RAM display memory 96, which is connected to CRTC 16 through a register/buffer 98. In the present terminal an "attribute" is defined to mean any of the following in relation to individual characters:

high/normal intensity or background highlight, blinking, underline, reverse video, double high characters, non-display double wide characters, and double high/wide characters. VAG 94 is connected to CRTC 16 and to display memory 96 by an attribute bus 100, and to CRTC 16 only through a control bus 102.

The display memory 96 is expandable by additional RAMS 104 connected as shown in Figure 1a. Additional

characters may be generated by including a character generator RAM 106 connected to character bus 92 and to VAG 94 through a dot bus 107. The CRT intensity is controlled by programmable contrast register 38 through a video amplifier 108, which is also connected to VAG 94.

#### 5 System Memory Map and Architecture

Referring now to Figure 2, there is shown a memory map of the present system illustrating the assigned memory of the various memory devices in hexadecimal form. In the basic 8-bit bytes in memory locations 0000—7FFF, and is expandable to 64K bytes in memory locations 8000—FFFF by the addition of DRAMs 48b and 48c.

10 Memory locations 0000—1FFF are assigned to system ROM 27, which contains 2K bytes (expandable to 8K) of program instructions to control the system initialization and diagnostics on power up, as well as instructions for loading DRAM 48a from the program storage memory resident in the GROMs. DRAM 48a is used by CPU 10 for execution of instructions to control the terminal.

Locations 2000—3000 are assigned to test ROM 29 (48K byte maximum) which is used primarily during 15 manufacture for test and diagnostic aids such as signature analysis.

Address 3000 is a write-only memory location that stores data in contrast register 38 to control the intensity level of the CRT as set by the operator. The attributes video on/off, background select, and highlight select are also controlled by register 38.

20 A read at memory location 3A00 inputs the data sent by keyboard 58 and resets its associated interrupt. A write to this location sends data to the keyboard for processing.

Read and write commands to memory locations 3A80—3A87 communicate the terminal configuration from control I/O 32 to CPU 10, that is, which of the basic and expandable devices are installed, as well as configuring the operational mode of the various communication devices are selected by the operator. Location 3B00 is reserved for communication option unit 87, which functions as described above.

25 Locations 3E00—3EFF are assigned to the 256 byte CMOS RAM 30, which is powered by a battery 31 to preserve its contents during power off conditions. CMOS RAM 30 stores terminal configuration information such as communications baud rates so that the terminal can readily communicate with the host computer when power is restored.

30 Display memory 96, which stores characters and associated attributes for display on the CRT screen, is accessed through CRT controller 16 as an I/O device. Display memory 96 contains 2K 16-bit bytes and is expandable to 8K in 2K increments.

The GROM storage memory is accessed by addressing the GROM I/F 56 at memory location 3B80. A read from this location inputs 4 bits of data from the selected GROM, and a write to this memory location outputs address and control information for accessing the GROM. In the basic terminal configuration three 35 16K byte GROMs, 52—1, 52—2, and 52—3, form a 48K byte program storage memory. This is expandable to a total of 128K bytes in 16K increments by the addition of GROMs 52—4 to 52—8. Dashed line 110 in Figure 1a denotes the expanded portions of the operating and program storage memory. Basically, a GROM is a PMOS, metal gate read-only memory that provides low cost mass data storage. In the present terminal, program instructions are down-loaded from the GROM to the DRAM operating system memory as needed 40 for execution by CPU10. In this manner, a 16K DRAM can be used for execution of 48K of program instruction. A similar method of overlaying program instructions into the DRAM is described in a co-pending patent application with publ. Nr. EP—A—48816 by Skelton, et al entitled "Virtual Memory Microcomputer Architecture."

45 Referring now to Figures 3—5, the GROM accessing procedure is shown in greater detail. Figure 3 shows the bit structure of a 20-bit byte used to access the GROM. The byte is loaded into GROM I/F 56 in 4-bit blocks or "nibbles" N1—N5. Bits 0—13 are used to select any one of 16K byte addresses in a particular GROM, while bits 14—17 are used to select one of up to eight GROMS. One embodiment of the GROM I/F is shown in Figure 4 and 5, illustrating circuitry for accessing a 16K byte GROM through a 4 bit data/address line D0—D3.

50

#### Display Control System

The display control system of the present terminal is formed of CRT controller 16, video attribute generator 94, display memory 96, character ROM 90, contrast register 38 and the associated circuitry and expandable memories 104 and 106. Briefly, CRTC 16 provides horizontal and vertical timing for the CRT, 55 addresses display memory 96 for CRT refresh, provides the scan line addresses for character ROM 90 and character generator 106, controls the cursor on the CRT, controls data transfers between CPU 10 and display memory 96 or character generator 106, and controls video attribute generator VAG 94. VAG 94 converts parallel dot data into serial video data, modifies dot data according to attributes and control inputs, generates a character clock from a video dot clock, and provides a data path for character generator 60 access. The CRT is configured to display information in a 25 row by 80 or 132 column format, with the line 25 reserved for terminal status messages. Each character cell defined by a row/column address is 9 × 11 dots or pixels, with a character size of 7 × 9 pixels.

65 The present terminal, as configured by either the operator or the host computer, allows for dividing the display memory into partitions. Each partition is called an editing boundary. The partition is defined in terms of rows of display, each row having 80 or 132 columns of display information. Additionally, the

editing boundary may be defined to be an unformatted display containing no fields, but containing display attributes on a character-by-character basis. The formatted display, called a FORM, contains at least one field and may contain display attributes on a character-by-character basis.

Only one editing boundary is displayed at a time. If more than one editing boundary is resident in the terminal, the length of each editing boundary must be equal to or greater than 24 rows of display information. That is, if 12 rows of double high/wide characters are used, then the 12 rows of double high/wide characters are 24 rows of display information.

A formatted editing boundary consists of "regions." A region is defined as a consecutive number of display rows viewed by the operator. The maximum number of consecutive rows possible for a region is 24 and the minimum is 1. However, the region data may exceed the display rows allocated for the defined region. This is called a scrolled region. Additionally, a region is defined as a full partition, left partition, or right partition. A full partition region is a region displayed to the operator using all columns possible in the CRT display. A left partition region is a region displayed to the operator using all columns to the left of the vertical split defined by the FORM. A right partition is a region displayed to the operator using all columns from and including the vertical split to the right defined by the FORM. If a region is defined as display only, with no operator access to the region, an entire region may be defined as a "protected region." A protected region is always guarded for transmission. A vertical split may exist for any FORM, however, only one vertical split may exist for any one form. A vertical split is a selected column 1—80 or 1—132 which defines the starting column of any right partition region.

Each region is a "window." A window which shows all the data in the display at once is called a fixed region or a non-scrollable region. A window that shows only a part or a segment of the data assigned to the window is called a scrolled region. Every editing boundary contains at least one window. The window may consist of all of the editing boundary which in turn may consist of all terminal memory, but the window still exists. Each region can be individually scrolled either discreetly or smoothly. Thus, in effect, each region can function as an independent display.

These and other features are obtained by means of a "data-linking" procedure whereby two "line attributes" are generated and stored in display memory 96 for each row of characters and character attributes on the CRT screen. A line attribute is a byte of data which functions as a pointer, telling CRT controller 16 at which display memory location the first character in the next row of characters appears. In the case of a vertical split region, wherein a full partition region is followed by a left and a right partition region on the display, the last character of the last row of the full partition region will be followed in display memory by a right attribute (RA) and a left attribute (LA). The LA points to the display memory location of the first character in the first row in the left partition, and the RA points to the memory location of the first character in the first row of the right partition. In this manner, various display functions such as inserting/deleting lines and scrolling are achieved by merely modifying the pointers RA and LA to indicate the memory location of the next desired row of characters. This greatly increases the processing throughput of CPU 10 because only the affected pointers need to be modified, without rearranging character and character attribute data in the display memory.

By way of example, and to aid in the understanding of the form and region concept described above, reference is made to Figure 6 wherein the display screen as seen by an operator is partitioned into four regions. It is assumed for this example that the terminal has been configured by the host computer to define two editing boundaries or forms. The form descriptors as well as region data tables (RDT) are loaded into the terminal memory or DRAM. Form descriptor 2 contains the information for configuring the display screen as shown in the drawing, that is, there is a vertical split at column 40 of the display screen and the form starts at display memory location 10 and ends at display memory location 50. As shown in the display memory map each of the four regions is a scrollable region, that is, the number of rows in each region exceeds the available "window" for display. Since the terminal in this example is configured to have a two page display memory, there are 48 lines of information available for the region definitions, although only 24 lines may be displayed at one time. Form 2 has four associated region data tables (RDT), one for each region on the display screen. Each RDT contains 8 bytes of information. Byte 0 is a predetermined region number, for example, to indicate whether a left or a right partition region will be displayed on the screen. Byte 1 indicates the starting row on the display screen for the associated region. Byte 2 is the current memory display starting row, that is, the display memory address of the currently displayed row. Byte 3 indicates the memory starting row number in the region, and byte 4 is the display ending row. Byte 5 is the current memory display ending row, byte 6 is the region memory ending row number in display memory, and byte 7 is the maximum relative column number for the region on the display screen. For example, referring to the RDT for region 1 in Figure 6, region 1 character and character attribute data are stored in display memory locations 10—24. The currently displayed information from region 1 on the display screen, however, is formed of 8 lines that are stored in display memory locations 12—19, and appear on the display screen rows 0—7. That is, memory locations 12—19 comprise the "window" of region 1 that is currently displayed. Since region 2 is a right partition split region occupying the same display screen rows as region 1, the display window for region 2 is the same size as the window for region 1. It should be noted that 48 rows of display memory have been used for form 2, although only 24 rows of information appear on the display screen at a time. Regions 3 and 4 are similarly defined by their associated RDTs.

Although in Figure 6 the rows of information in display memory corresponding to the rows on the

**0 059 349**

display screen are shown in sequential order, they may in fact be located at any display memory location. The RA and LA pointers associated with each row of screen information, as described above, will map the location of the next characters in display memory to be displayed on the screen.

Referring now to Figure 7, the use of these pointers in scrolling and individual region on the display screen will now be described by way of example. The display screen is shown divided into four regions, with the letters in each region indicating the first and last rows of characters in that particular region, for example, line B in region 1 is the first row of characters and line C represents the last row of characters in region 1. Similarly, line J represents the first row of characters in region 3 while line K represents the last row of characters. The "after" view of the display screen show region 3 scrolled up one row of characters, that is, row K is now the first displayed row of characters and row L is the last displayed row. Also shown in FIGURE 7 is a map of the RA and LA pointers showing the action of the pointers both before and after the scroll function has been completed. Before scrolling is initiated pointer LA of the status line points to row B in region 1 and pointer LA of row B points to row C. Since region 2 is a left partition split region, pointer LA of row C points to row F in region 2, while pointer RA of row C points to row J in right partition region 3. Pointer LA of row G in region 2 points to row N in region 4. Because region 3 is a right partition region, only the associated RA pointers are used to indicate the next successive row in that region. Region 3 is scrolled up by one row merely by changing pointer RA in row C of region 1 so that it points to row K in region 3 instead of row J. Thus, scrolling and other editing functions are achieved by changing the pointers only, rather than rearranging whole blocks of data in display memory as was done in prior art systems. Further, the use of controller 16 and its associated circuitry to assign and modify the various pointers greatly reduces the processing time required to perform these functions and, since CRTC 16 operates asynchronously with CPU 10, the processing load on CPU 10 is reduced.

Smooth scrolling, i.e., the scrolling of a line one pixel or scan line at a time, may be performed independently in any of the vertical or horizontal split regions under the control of CRTC 16, which will now be described in greater detail.

TABLE 1

Select Line (A0—A2) Write Assignments

$\overline{CE}$	$\overline{WR}$	A2	A1	A0	Register Loaded
1	X	X	X	X	Address decoding inhibited
0	0	0	0	0	Control Reg.
0	0	0	0	1	Command Reg.
0	0	0	1	0	Character Data Reg.
0	0	0	1	1	Attribute Data Reg.
0	0	1	0	0	Cursor Address Reg. (Upper 6 bits)
0	0	1	0	1	Cursor Address Reg. (Lower 8 bits)
0	0	1	1	0	Vertical Split Reg.
0	0	1	1	1	Not used

TABLE 2

Select Lines (A0—A2) Read Assignments

$\overline{CE}$	$\overline{RD}$	A2	A1	A0	Register Read
1	X	X	X	X	Address decoding inhibited
0	0	0	0	0	Status Reg.
0	0	0	0	1	Not used
0	0	0	1	0	Character Data Reg.
0	0	0	1	1	Attribute Data Reg.
0	0	1	0	0	Cursor Address Reg. (Upper 5 bits)
0	0	1	0	1	Cursor Address Reg. (Lower 8 bits)
0	0	1	1	0	Not used
0	0	1	1	1	Not used

TABLE 3

## Control Register Bit Assignments

Control Bit	Name	Description
7		Not used
6		Not used
5	TSTMD	Test mode
4	COU <sub>T</sub>	Control out
3	ENLA	Enable row attributes
2	CBLINK	Blink cursor
1	CURON	Enable cursor
0	DFMS	Select alternate display format

## Display Controller

In one embodiment, display controller CRTC 16 is a single integrated circuit device, which generates timing signals for a standard or a nonstandard raster-scan CRT monitor incorporating a non-interlaced format. CRTC 16 controls the horizontal and vertical display formatting, the independent display refresh memory 96, and the cursor address. Internal character and attribute data registers synchronize attribute and character data allowing controlled transfers between CPU 10 and display memory 96 or video attribute generator 94. As described above, controller 16 and its associated circuitry provide smooth scrolling and split screen capabilities as well as determining the number of characters per row, the number of scan lines per row, and the number of rows per display frame. In conjunction with VAG 94, the various character attribute features are also provided.

Referring now to Figures 8a—8c, CRTC 16 is shown in block diagram form. Controller 16 interfaces with CPU 10 through system address bus 14 (select lines A0—A2), system data bus 36 (data lines D0—D7), and system control bus 12 (RD/, WR/, and CE/). Lines A0—A2 are connected to an address decode logic circuit 112. Lines D0—D7 are connected to a data buffer 113 which is in turn connected to an internal data bus 114. The signal lines from control bus 12 are connected to an I/O control logic circuit 116. Address decoder 112 controls the various internal registers connected to data bus 114 by decoding the three address inputs together with the I/O control logic 116 outputs. Tables 1 and 2 below list the write and read assignments for the three select lines, A0—A2. I/O control 116 gates the control signals from CPU 10 and controls data buffer 113 and the bus precharge.

A control register 118 connected to data bus 114 is a write-only register that contains the control bits listed in Table 3. Bit 0 (DFMS) selects either a standard or an alternate display format. The logical value of bit 0 is output as the ADF signal shown in Figure 8c. Bit 1 (CURON) enables the cursor output. Setting control bit 1 to zero inhibits the cursor output unconditionally. Bit 2 (CBLINK) selects either a blinking or nonblinking cursor. Bit 3 (ENLA) sets the mode of operation for controller 16. Enabling the line attributes allows for a display memory format with row attributes following each row. When disabled, the display format is assumed to be conventional, that is, each row is sequentially followed by the next displayed row with no row attributes present. The logical value of control bit 4 (COU<sub>T</sub>) appears as signal (COU<sub>T</sub>/) during a read or write to display memory 96. During active display this signal is inactive (High) and during retrace the signal is active (Low). Setting control bit 5 (TSTMD) reduces the display format to 16 characters per row and 2 scan lines per character row. Resetting this bit enables normal device operation. This is primarily used for testing and inspection purposes.

A command register 120 connected to data bus 114 is an eight-bit write-only register which allows CPU 10 to issue display memory read or write commands, scroll commands, or interrupt commands. The register bit assignment is shown in Table 5 wherein an "x" indicates a "don't care" value.

A character register 122 connected to data bus 114 is an eight-bit read/write register used by CPU 10 for transferring data to and from display memory 96. The lower seven bits from register 122 appear on an internal data bus 122a. Bus 122a is connected to a character data buffer 123 which is in turn connected to character data bus 92. An attribute register 124 is an eight-bit read/write register similar to character register 122. Register 124 is connected to an internal attribute data bus 124a which is in turn connected to an attribute data buffer 125. The eight attribute bits and the most significant character bit are connected to external attribute bus 100 as shown in Figures 1a and 1b. A cursor address register 126 connected to data bus 114 is a thirteen-bit read/write register containing the absolute display memory address of the displayable cursor. The contents of this register are also the address used for display memory transfers.

A status register 128 connected to data bus 114 is a 6-bit read-only register with a bit assignment as shown in Table 4. Bit 0 (SVBLK) indicates the start of the vertical blanking interval, i.e., the beginning of the status row. Bit 1 (RWCOM) indicates that a read or write operation to display memory 96 has been completed. Bit 2 (SCRCOM) indicates that a smooth scroll operation has been completed. This will occur at the beginning of the status row on the last frame of the scroll. Bit 3 (INTP) indicates that an interrupt

**0 059 349**

condition has occurred. This bit is reset or enabled by means of the command register 120. Bit 4 (RWBSY) is set following a read or a write command from CPU 10, indicating that the desired operation has not been completed. Upon completion of the read/write, this bit is reset. Bit 5 (SCRBSY) is set following a smooth scroll command from CPU 10 and is reset upon receipt of a scroll complete signal.

5

**TABLE 4**

Status Register Bit Assignments

10

Status Bit	Name	Description
7		Not used
6		Not used
15	5	SCRBSY
	4	RWBSY
	3	INTP
	2	SCRCOM
	1	RWCOM
20	0	SVBLK

**TABLE 5**

Command Register Bit Assignment

25

Command Reg. Bit								Description	
7	6	5	4	3	2	1	0		
								<u>Read/Write Command</u>	
30	0	0	X	0	0	X	0	X	No operation
							1	0	Read at cursor address
							1	1	Write at cursor address
35	0	0	X	0	1	0	1	0	Read and post-decrement at cursor address
					1	1	1	0	Read and post-increment at cursor address
					1	0	1	0	Write and post-decrement at cursor address
					1	1	1	1	Write and post-increment at cursor address
	0	0	X	0	1	0	0	X	Enable post-decrement at cursor address
40	0	0	X	1	0	X	0	X	Enable post-increment at cursor address
									Post increment/decrement at cursor address
								<u>Scroll Command</u>	
45	0	1	X	X	X			0	Scroll up
								1	Scroll down
								0	Offset counter internal clock mode
								1	Offset counter external clock mode
								0	No operation
50								1	Offset counter external increment clock
								<u>Interrupt Control Command</u>	
55	1	0	X					0	Disable vertical blank interrupt mask
								1	Enable vertical blank interrupt mask
								0	Disable read/write interrupt mask
								1	Enable read/write interrupt mask
								0	Disable scroll interrupt mask
								1	Enable scroll interrupt mask
60								0	No operation
								1	Reset interrupts
								0	Disable interrupts
								1	Enable interrupts
65	1	1	X	X	X	X	X	X	No operation

A vertical split register 130, connected to data bus 114, is an 8-bit write-only register containing the column position of a vertical split and is zero based. That is, if register 130 is loaded with zero, on a split region the left partition would contain one character and the right partition would contain the remaining number of characters that are displayable.

5 The overall display format is controlled by a series of cascaded free-running counters. Two independent formats are mask programmable, each having two mask programmable formats for providing separate refresh rates. A horizontal sync register 132 is an 8-bit counter that controls horizontal display timing and is incremented at the character rate by a CCLK signal from video attribute generator 94. A programmable logic array (PLA) decodes the current register value and outputs signals corresponding to  
10 the total number of displayed characters, the start of horizontal sync, the end of horizontal sync and the end of horizontal retrace which clears the register. The operation of PLA's are known in the art and need not be described in detail. A scan line/vertical adjust counter 134 is a 4-bit counter that is incremented at the end of horizontal retrace, thus counting the number of scan lines per character row. Its count is decoded by means of a PLA and when the total number of scan lines is reached it is cleared and the counter in a vertical sync  
15 register 136 is incremented. When the register 136 counter reaches its total number of character rows count, register 134 enters a vertical adjust mode which permits a resolution of the refresh rate to within one scan line. Upon completion of vertical adjust, register 134 is cleared and begins counting scan lines in the upper-most character row.

Vertical sync register 136 as described above counts character rows and is clocked by register 134.  
20 Register 136 is a 7-bit counter which is PLA decoded for total displayed character rows, status row position, total character rows (including vertical retrace), and vertical sync position. The vertical sync pulse is set at vertical sync position and reset on the total character rows count.

An address counter 138 is a 4-bit counter that counts character scan lines as does register 136. Register 136 is used for normal characters while address counter 38 is used for double high and smooth scroll characters. Counter 138 tracks register 136 until one of these special character regions are detected. For  
25 double high characters counter 138 toggles between increment and inhibit increment states. For the scrolling function, counter 138 is preset with an offset signal from an offset counter 140 and then is incremented as before. Offset counter 140 is a four-bit counter which is incremented every time the status row position is reached, or once every frame. This increases the smooth scrolled offset one pixel (or scan line per region) per display frame. Offset counter 140 is initialized when a smooth scroll command is  
30 received from CPU 10 by means of command register 120. This either presets or clears counter 140 based upon the direction of the scroll. A blink rate register 142 is a three stage ripple counter that is clocked by the carry output of the most significant stage of offset counter 140. It is used to establish the cursor blink rate and the blink rate output.

35 The display memory 96 is addressed by the thirteen bit display memory address (DMA) outputs (DMA0—DMA12), from a buffer 143 connected to an internal DMA bus 143a. Associated control signals include write control (DMWR/), select memory or character generator (CGA), and bank select control (COUT). A display memory address counter 144 generates the DMA for active, or visible, display. Counter 144 is connected to bus 143a and contains three registers: a region 1 register 146, a region 2 register 148,  
40 and a split register 150. Register 146 contains the starting address of a character row and is loaded into the memory address counter 144 at the beginning of every scan line of that particular row. After the last scan line of a row has been displayed, register 146 is updated with the starting address of the next row. This may be done either by: loading memory address counter 144 contents plus 1 into register 146 when scan line attributes are disabled; loading the address following the row attribute in register 146 when the link  
45 address is not valid and row attributes are enabled; or, loading the thirteen bit address stored in memory 96 at the row attribute location into register 146, when the link address is valid and row attributes are enabled.

Region 2 register 48 is the equivalent of register 146 for a split region. That is, register 146 contains the starting address for the left region, while register 148 contains the starting address for the right region.  
50 Register 148 is loaded from either the memory address counter 144 or from display memory 96 through the internal attribute and character data buses 124a and 122a.

Split register 150 is loaded from memory address counter 144 for a row falling into two regions with the display memory address of the row attribute of the left region. During the retrace state times, the address in register 150 is loaded into the display address counter 144 to fetch data at that location. When  
55 not on the last scan line of a split row, register 150 contains a refresh address that is loaded into memory address counter 144 following the state time throughout horizontal retrace for the purpose of refreshing dynamic memory devices. When a read or a write to display memory command is given to the controller 16, it is synchronized in command register 120 and decoded. During the actual read or write operation, which occurs during the state time of retrace, the contents of cursor address register 126 are placed onto  
60 display memory address bus 143a. If the cursor is enabled, the cursor output will be active since a cursor address register comparator 152 always compares the contents of register 126 with what is on the DMA bus 123a. In this case it is comparing the register 126 contents with itself.

The character scan line outputs CSL0—3 from a scan line control logic circuit 154 may also be considered to be a part of the display memory address. These outputs indicate which scan line of the  
65 character the display memory address corresponds to. The four bits may be either from counter 138,

counter 134 or the four least significant bits of the cursor address register 126. The selection between these counters depends upon the type of row being displayed, i.e., normal, smooth scrolled, or double high. The four cursor bits are always selected during the state times and are used in accessing the external character generator RAM 106. A PLA follows a multiplexer in circuit 154 for the purpose of decoding the underline position based on the selection of a standard or an alternate display format.

A state timing logic circuit 156 is a seven bit shift register with recirculation capability. At the beginning of horizontal retrace, a one is shifted into this register and serially propagates through all stages. The outputs of the stages represent state times which are used to control a register control logic circuit 158. All row attribute fetches and display memory reads and writes occur during the state times with the row attribute fetch having priority over the command oriented memory accesses. The register control logic embodies control of the character 122 and attribute 124 registers with respect to the character 92 and attribute 100 external data buses. Also controlled by logic circuit 158 are the register transfer of memory address counter 144, the command register 120 synchronization and control logic, the precharging of character bus 122a and attribute bus 124a, and the control of buffers 123 and 125. Following the last displayed character row until the beginning of vertical adjust, the state timer recirculates shifting the "one" from the last stage back to the first, thereby increasing the bandwidth of the processor (CPU 10) access to display memory 96.

A row attribute logic circuit 160 provides an input to the register control logic 158 based upon the row currently being displayed and the row to be displayed next. The outputs of the row attribute logic 160 provide for the following: controlling counter 138 by presetting it with the offset counter 140, by clearing it, or by enabling it to count double high; signalling the register control logic 158 when a left and/or a right row attribute fetch should occur; and determining when register 146 or register 148 should be loaded from the character and attribute buses 122a, 124a. A set of memory control buffers 162 is connected to row attribute logic 160 and to register control logic 158 in order to buffer the inputs and outputs to display memory 96 and VAG 94.

Table 6 below summarizes the various signals and I/O ports on controller 16. The registers, PLA's and counters described above are known in the art and the electronics circuitry of each need not be described in detail.

30

35

40

45

50

55

60

65

TABLE 6

Signature	Pin	I/O	Description
D7 (MSB) D6 D5 D4 D3 D2 D1 D0 (LSB)		I/O I/O I/O I/O I/O I/O I/O I/O	— D7 through D0 constitutes the data bus. The data bus serves as an input for register contents to the controller and output for buffer/cursor address read data and attribute/character data.
A2 (MSB) A1 A0 (LSB)		I I I	— A2 through A0 are select lines, used to select the internal registers of the controller.
$\overline{RD}$		I	— Read — a negative true signal indicates that the controller should gate data onto the data bus.
$\overline{WR}$		I	— Write — a negative true signal indicates that the controller should gate data bus into internal register.
$\overline{CE}$		I	— Chip Enable — a negative true signal indicates that the controller is being addressed.
$\overline{RESET}$		I	— Reset — a negative true signal initializes the controller, and is active for 1 $\mu$ s minimum.
$\overline{INT}$		O	— Interrupt — a negative true signal indicates that at least one of the interrupt conditions has occurred. $\overline{INT}$ is an open drain output.
50HZ		I	— 50/60 Hz Control Input — this input indicates to the controller to switch to second set of vertical timing parameters in the current display format.
HSYNC		O	— Horizontal Sync — a positive true signal is used to drive the horizontal deflection circuits of a CRT.
$\overline{VSYNC}$		O	— Vertical Sync — a negative true signal is used to drive the vertical deflection circuits of a CRT.
CBLANK		O	— Composite Blanking — a positive true signal is used to blank the video input of a CRT during Horizontal and Vertical Retrace.
DMA12 (MSB) DMA11 DMA10 DMA9 DMA8 DMA7 DMA6 DMA5 DMA4 DMA3 DMA2 DMA1 DMA0 (LSB)		O O O O O O O O O O O O O	— DMA12 through DMA0 constitutes the Display Memory Address. The controller outputs the address of the Display Memory location which it is currently accessing.

0 059 349

TABLE 6 (continued)

Signature	Pin	I/O	Description
VCC 1		I	— Supply Voltage (+5V NOM).
VCC 2		I	— Supply Voltage (+5V NOM).
VSS		I	— Ground Reference Voltage
CD6 (MSB)		I/O	— CD6 through CD0 constitutes the character data bus. This data bus defines the character the controller stores or reads from the Display Memory.
CD5		I/O	
CD4		I/O	
CD3		I/O	
CD2		I/O	
CD1		I/O	
CD0 (LSB)		I/O	
AD8 (MSB)		I/O	— AD8 through AD0 constitutes the Attribute Data Bus. This data bus defines the Visual Attributes the controller stores or reads from the Display/Memory.
AD7		I/O	
AD6		I/O	
AD5		I/O	
AD4		I/O	
AD3		I/O	
AD2		I/O	
AD1		I/O	
AD0 (LSB)		I/O	
CCLK		I	— Character Clock — this establishes the basic character clock rate and is used to synchronize the controller internal timing.
CSL3 (MSB)		O	— Character Scan Line — CSL3 through CSL0 indicate the binary number of the scan line for the character to be displayed.
CSL2		O	
CSL1		O	
CSL0 (LSB)		O	
CURSOR		O	— Cursor — a positive true signal indicating the location in the Display Memory defined to have the cursor.
UNPOS		O	— Underline Position — a positive true signal indicates that the underline attribute should be displayed if applicable.
BLINK		O	— Blink rate — this signal determines the rate at which characters should be blinked when applicable.
$\overline{\text{DMWR}}$		O	— Write Display Memory — A negative true signal indicates that the data gated onto the character and attribute data bus will be stored in the Display Memory at the address provided by the Display Memory Address Bus.
CGA		O	— Character Generator Access — a positive true signal indicates that the controller is accessing the Character Generator Circuitry instead of the Display Memory.
ADF		O	— Alternate Display Format — a positive true signal indicates that the controller has been configured to operate in the Alternate Display Format instead of the standard format.
$\overline{\text{COUT}}$		O	— Control Output — this output signal is the inverse of Bit 6 of the Control Register of the controller.

## Keyboard and Keyboard Interface Unit

Keyboard unit 58 is self-contained, with its own microcomputer 64, for scanning and decoding the keys, assigning various functions thereto and transmitting and receiving data and instructions serially via I/F 60 to CPU 10. Various character formats are selectable, e.g., for different countries as stored in character RAM 106. The advantages of separate keyboard unit include a reduced processing load on CPU 10 and thus a faster throughput.

Referring now to Figure 9, keyboard unit 58 includes an array or matrix 166 of momentarily-closed single action switches arranged in rows and columns. Microcomputer 64, for example, a model 3870 available from MOSTEK Corporation and from Motorola, Inc., includes internal ROM and RAM. Input/output ports, PO and P4 are connected to a series of open collector buffers U2—U4, used for strobing matrix 166. A feedback return is produced via a row path into port P5 of U1. Four control keys, shown in the left hand portion of Figure 9, are connected to U1 through input port P1. Bits 5 and 6 of port P0 are connected to a bidirectional data line E3 and a clock line E4 which are connected in turn to keyboard I/F 60 for serial transmission of signals thereto and therefrom. Lines E1 and E2 provide power and ground connections for keyboard unit 58. The four logic gates denoted as U5 in Figure 9 form a bell or tone circuit that is controlled by microcomputer 64 upon commands from CPU 10 and from internal error codes.

Referring now to Figure 10, I/F 60 converts serialized data from keyboard data 58 to a parallel form in a tri-state output shift register 168 for output onto I/O data bus 46. Flip-flop circuits FF1—FF4 control the data transfer to shift register 168. FF1 determines whether I/F 60 is in a read or a write mode and controls the data path. FF2 signals microcomputer 64 that CPU 10 is still processing data and is not ready to read. FF3 sends an interrupt signal to CPU 10 to indicate that the data in register 168 is available for reading, and FF4 establishes a time-delay to stabilize the data transfer into register 168 from the I/O data bus 46. The signals that appear on the left hand portion of Figure 10 are from CPU 10. This arrangement reduces the processing load on CPU 10, while expanding the versatility of keyboard unit 58.

Figure 10a is a timing diagram showing the relationship of various signals in I/F 60 to the clock and data signals. The clock signal is generated by microcomputer 64, and the serial data in the read and the write modes is bi-directional.

## Claims

30

1. A data display system for forming a plurality of independently controllable display regions on a display device of the type having: data processor means (10), main memory means (48a—c) connected to said data processor means (10), input means (58, 60) connected to said data processor means and to said main memory means (48), display controller means (16) connected to said data processor means (10), display memory means (96, 104) connected to said display controller means (16) wherein each row of characters to be displayed on said display device is assigned one or more memory location indicating pointers as defined by said display controller means (16) to indicate the display memory address of a subsequent row of characters to be displayed and wherein said display controller means (16) alters the sequence of rows displayed on said display device by modifying the memory location indicating pointers of the selected rows of characters, characterized by character memory means (90, 106) connected to said display controller means for storing a predetermined set of character data codes, whereby the data code corresponding to a selected character is loaded into said display memory means (96, 104) in response to a command from said display controller means (16), visual attribute generator means (94) connected to said display memory means (96, 104) and said character memory means (90, 106), means for loading character and visual attribute data into selected display memory locations from a data bus (92), wherein said character data is stored in said character memory means (90, 106) and said attribute data is generated by said visual attribute generator means (94), and wherein a plurality of character and attribute data bytes form a block of data in the display memory means (96, 104) corresponding to a row of characters as seen on the display device, means for assigning one or more memory location indicating pointers to each displayed row of characters, corresponding to said data blocks in the display memory (96, 104), wherein each of said pointers contains the display memory address of the first character in the subsequent row of characters to be displayed on the display device, means (130) for storing a data byte corresponding to a preselected column on said display device wherein a vertical split region will begin, means (160) for storing a data byte corresponding to the starting address of a character row, means (162) for storing a data byte corresponding to the starting row address of a vertical split region, and means (136, 138) for determining which scan line of a character a particular display memory address corresponds to, whereby a plurality of rows of characters appearing on the display device are scrolled one scan line at a time.

2. Data display system according to claim 1, further including means (16) for assigning within said memory location indicating pointers selected region row visual attribute information, and means (118) for presenting the next row of characters on said display device in a selected manner corresponding to said selected visual attributes.

3. Data display system according to claim 2 wherein said selected region row visual attribute generator means (94) comprises double high character generating means (138) whereby each scan line of region character row information is repeated on said display device before the next character row scan line is accessed from said display memory means (96, 104).

65

**Patentansprüche**

1. Datenanzeigesystem zur Bildung mehrerer unabhängig steuerbarer Anzeigebereiche auf einer Anzeigevorrichtung, die folgendes enthält: eine Datenverarbeitungsvorrichtung (10), eine Hauptspeichervorrichtung (48a—c), die an die Datenverarbeitungsvorrichtung (10) angeschlossen ist, eine Eingabevorrichtung (58, 60), die an die Datenverarbeitungsvorrichtung und an die Hauptspeichervorrichtung (48) angeschlossen ist, eine Anzeigesteuervorrichtung (16), die an die Datenverarbeitungsvorrichtung (10) angeschlossen ist, eine Anzeigespeichervorrichtung (96, 104), die an die Anzeigesteuervorrichtung (16) angeschlossen ist, wobei jeder auf der Anzeigevorrichtung wiederzugebenden Zeichenreihe ein oder mehrere von der Anzeigesteuervorrichtung (16) definierte Speicherplatzhinweiszeiger zugeordnet sind, die die Anzeigespeicheradresse einer nachfolgenden, wiederzugebenden Zeichenreihe angeben, und wobei die Anzeigesteuervorrichtung (16) die Folge der auf der Anzeigevorrichtung wiederzugebenden Reihen durch Modifizieren der Speicherplatzhinweiszeiger der ausgewählten Zeichenreihen verändert, gekennzeichnet durch Zeichenspeichervorrichtung (90, 106), die an die Anzeigesteuervorrichtung angeschlossen sind und einen vorbestimmten Satz von Zeichencodes speichern, wodurch der einem ausgewählten Zeichen entsprechende Datencode in Abhängigkeit von einem Befehl aus der Anzeigesteuervorrichtung (16) in die Anzeigespeichervorrichtung (96, 104) geladen wird, einen an die Anzeigespeichervorrichtung (96, 104) und an die Zeichenspeichervorrichtung (90, 106) angeschlossenen Sichtmerkmalsgenerator, Mittel zum Laden von Zeichen- und Sichtmerkmalsdaten in ausgewählte Anzeigespeicherplätze von einem Datenbus (92), wobei die Zeichendaten in der Zeichenspeichervorrichtung (90, 106) gespeichert und die Merkmalsdaten durch den Sichtmerkmalsgenerator (94) erzeugt werden, und wobei mehrere Zeichen- und Merkmalsdatenbytes in der Anzeigespeichervorrichtung (96, 104) einen Datenblock bilden, der einer auf der Anzeigevorrichtung sichtbaren Zeichenreihe entspricht, Mittel zum Zuordnen eines oder mehrerer Speicherplatzhinweiszeiger zu jeder wiedergegebenen Zeichenreihe entsprechend den Datenblöcken in der Anzeigespeichervorrichtung (96, 104), wobei jeder der Zeiger die Anzeigespeicheradresse des ersten Zeichens in der nachfolgenden Zeichenreihe enthält, die auf der Anzeigevorrichtung wiederzugeben ist, Mittel (130) zum Speichern eines einer ausgewählten Spalte auf der Anzeigevorrichtung entsprechenden Datenbytes, wodurch ein vertikal geteilter Bereich beginnt, Mittel (160) zum Speichern eines Datenbytes, das der Startadresse einer Zeichenreihe entspricht, Mittel (162) zum Speichern eines Datenbytes, das der beginnenden Reihenadresse eines vertikal geteilten Bereichs entspricht, und Mittel (136, 138), die bestimmen, welche Abtastzeile eines Zeichens einer bestimmten Anzeigespeicheradresse entspricht, wodurch mehrere Zeichenreihen, die auf der Anzeigevorrichtung erscheinen, mit einer Abtastzeile pro Zeiteinheit verschoben werden.

2. Datenanzeigesystem nach Anspruch 1, mit Mitteln (16) zum Zuordnen ausgewählter Bereichs-Reihensichtmerkmalsinformationen innerhalb der Speicherplatzhinweiszeiger und Mitteln (118) zum Darstellen der nächsten Zeichenreihe auf der Anzeigevorrichtung in einer ausgewählten Weise entsprechend den ausgewählten Sichtmerkmalen.

3. Datenanzeigesystem nach Anspruch 2, dadurch gekennzeichnet, daß der Sichtmerkmalsgenerator (94) für ausgewählte Bereichsreihen Generatormittel (138) für doppelt hohe Zeichen enthält, wodurch jede Abtastzeile aus Bereichszeichen-Reiheninformationen auf der Anzeigevorrichtung wiederholt wird, bevor die nächste Zeichenreihen-Abtastzeile aus der Anzeigespeichervorrichtung (96, 104) erfaßt wird.

**Revendications**

1. Système d'affichage de données pour former plusieurs zones d'affichage contrôlables indépendamment sur un dispositif d'affichage du type comprenant un processeur de données (10), une mémoire principale (48a à c) connectée audit processeur de données (10), des dispositifs d'entrée (58, 60) connectés audit processeur de données et à ladite mémoire principale (48), un contrôleur d'affichage (16) connecté audit processeur de données (10), une mémoire d'affichage (96, 104) connectée audit contrôleur d'affichage (16), un ou plusieurs pointeurs d'emplacement de mémoire définis par ledit contrôleur d'affichage (16) étant affectés à chaque rangée de caractères à afficher sur ledit dispositif d'affichage pour indiquer l'adresse, dans la mémoire d'affichage, d'une rangée suivante de caractères à afficher, et ledit contrôleur d'affichage (16) modifiant la séquence des rangées affichées sur ledit dispositif d'affichage en modifiant les pointeurs d'emplacement de mémoire des rangées de caractères sélectionnées, caractérisé en ce qu'il comprend une mémoire de caractères (90, 106) connectée audit contrôleur d'affichage pour stocker une série prédéterminée de codes de données de caractères, le code de données correspondant à un caractère sélectionné étant ainsi chargé dans ladite mémoire d'affichage (96, 104) en réponse à une commande dudit contrôleur d'affichage (16), un générateur d'attributs visuels (94) connecté à ladite mémoire d'affichage (96, 104) et à ladite mémoire de caractères (90, 106), des moyens pour charger des données de caractères et d'attributs visuels dans des emplacements sélectionnés de la mémoire d'affichage à partir d'un bus de données (92), lesdites données de caractères étant stockées dans ladite mémoire de caractères (90, 106) et lesdites données d'attributs étant engendrées par ledit générateur d'attributs visuels (94) et plusieurs multiplats de données de caractères et d'attributs formant un bloc de données dans la mémoire d'affichage (96, 104) correspondant à une rangée de caractères visualisée sur le dispositif d'affichage, des moyens pour affecter un ou plusieurs pointeurs d'emplacement de mémoire à

chaque rangée de caractères affichée, correspondant auxdits blocs de données dans la mémoire d'affichage (96, 104), chacun desdits pointeurs contenant l'adresse dans la mémoire d'affichage, du premier caractère de la rangée de caractères suivante à afficher sur le dispositif d'affichage, des moyens (130) pour stocker un multiplet de données correspondant à une colonne présélectionnée sur ledit  
5 dispositif d'affichage à laquelle doit commencer une zone à fractionnement vertical, des moyens (160) pour stocker un multiplet de données correspondant à l'adresse de début d'une rangée de caractères, des moyens (162) pour stocker un multiplet de données correspondant à l'adresse de la première rangée d'une zone à fractionnement vertical, et des moyens (136, 138) pour déterminer à quelle ligne d'analyse d'un caractère correspond une adresse particulière de la mémoire d'affichage, plusieurs rangées de caractères  
10 apparaissant sur le dispositif d'affichage étant ainsi décalées d'une ligne d'analyse à la fois.

2. Système d'affichage de données selon la revendication 1, caractérisé en ce qu'il comprend en outre des moyens (16) pour affecter, dans ladite mémoire des informations d'attributs visuelles de rangée du zones sélectionnées par des pointeurs d'indication d'emplacements et des moyens (118) pour présenter la rangée de caractères suivante sur ledit dispositif d'affichage, d'une manière sélectionnée correspondant  
15 auxdits attributs visuels sélectionnés.

3. Système d'affichage de données selon la revendication 2, caractérisé en ce que ledit générateur d'attributs visuels de rangée sélectionnés (94) comprend des moyens de génération de caractères de hauteur double (138) au moyen desquels chaque ligne d'analyse des informations d'une rangée de caractères d'une zone est répétée sur ledit dispositif d'affichage avant que la ligne d'analyse de la rangée  
20 suivante de caractères ne soit extraite de ladite mémoire d'affichage (96, 104).

25

30

35

40

45

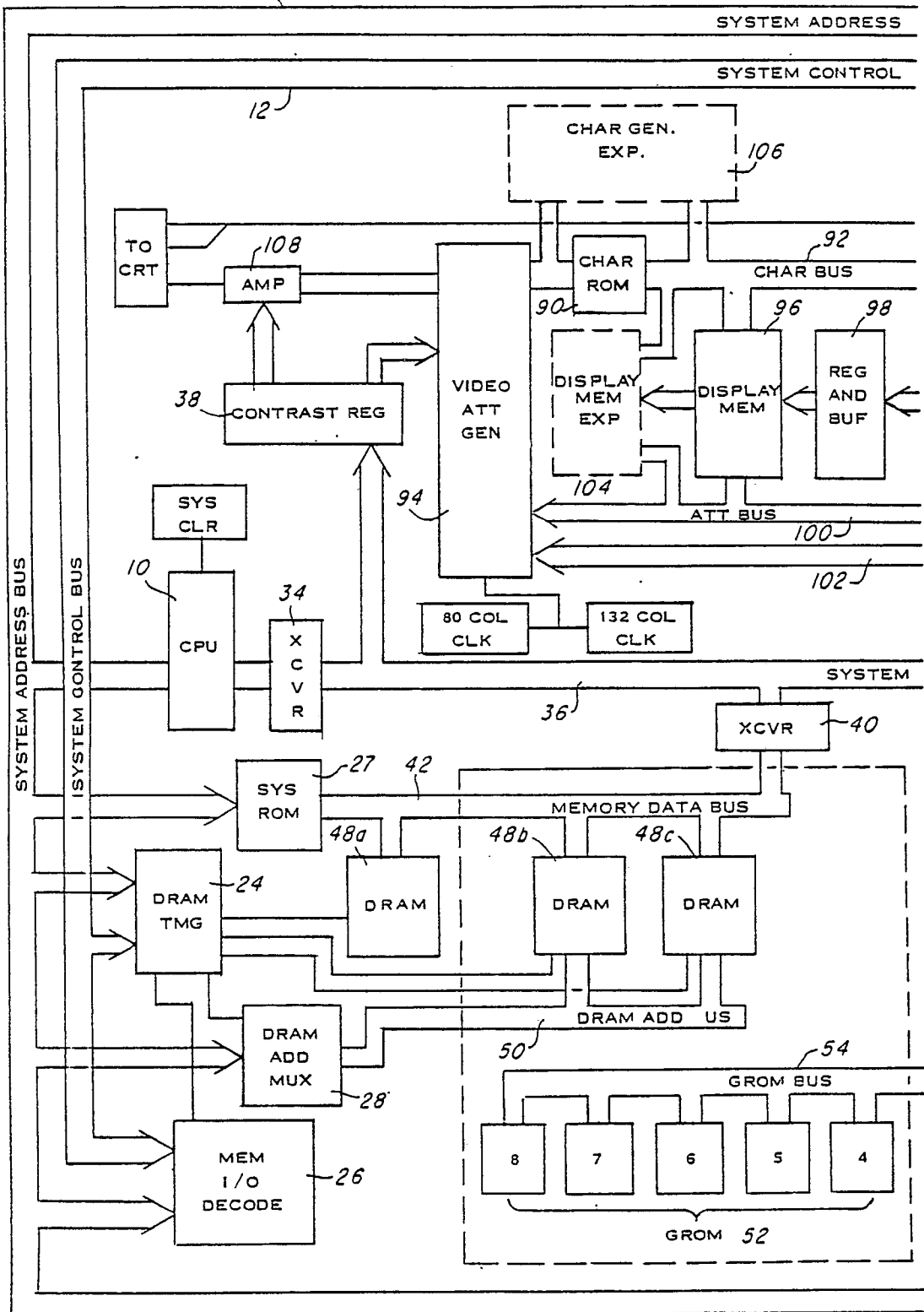
50

55

60

65

Fig. 1a



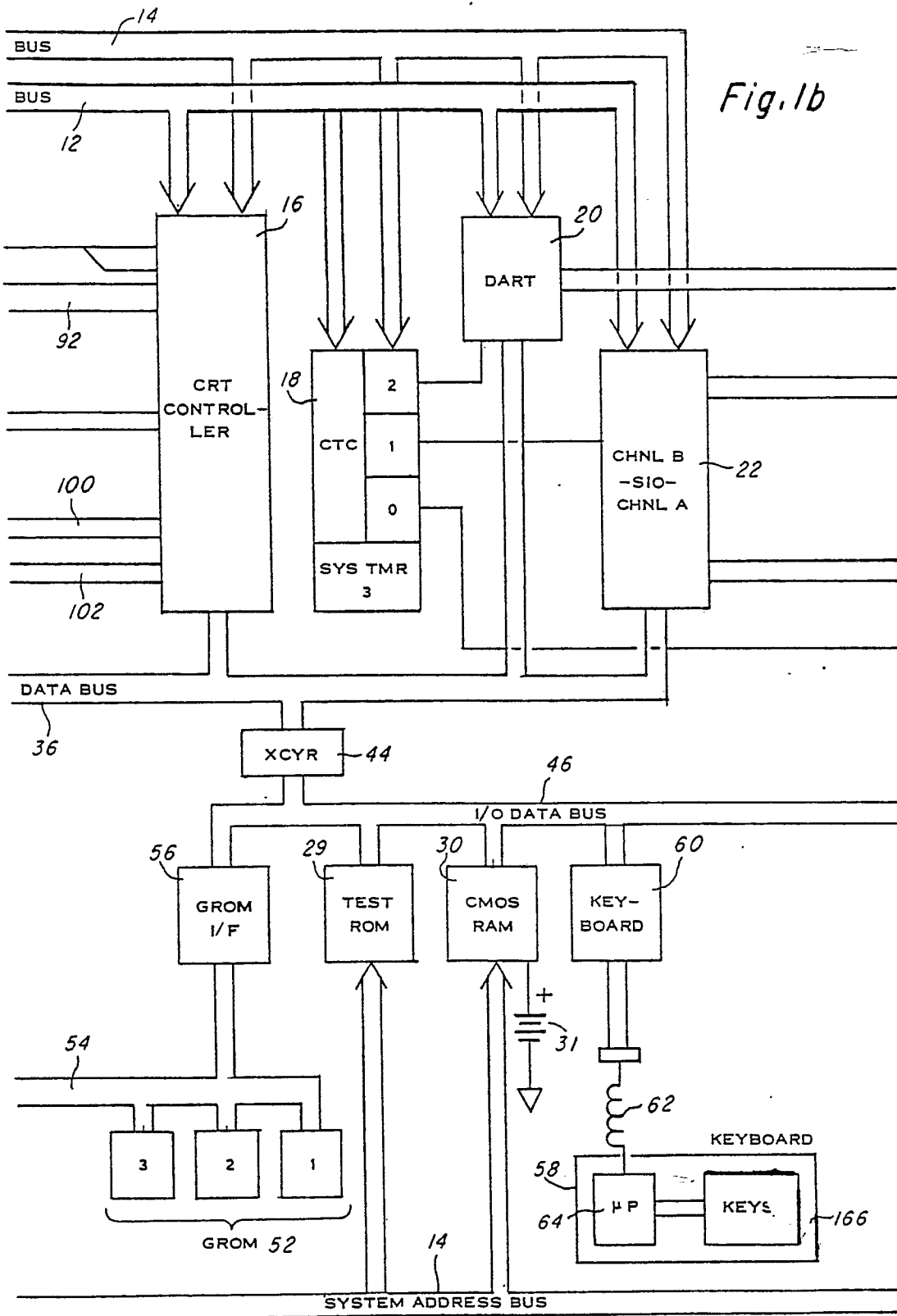
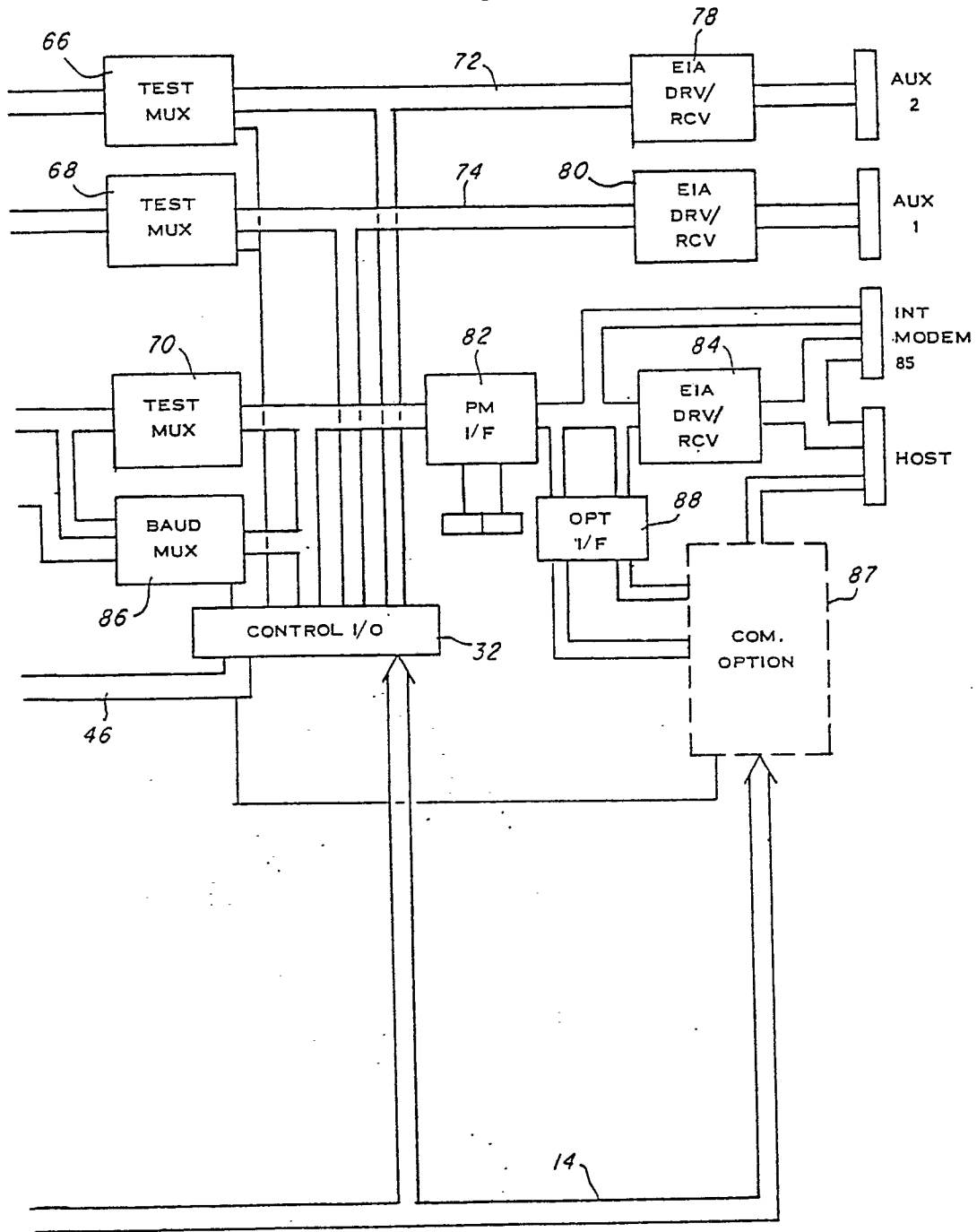
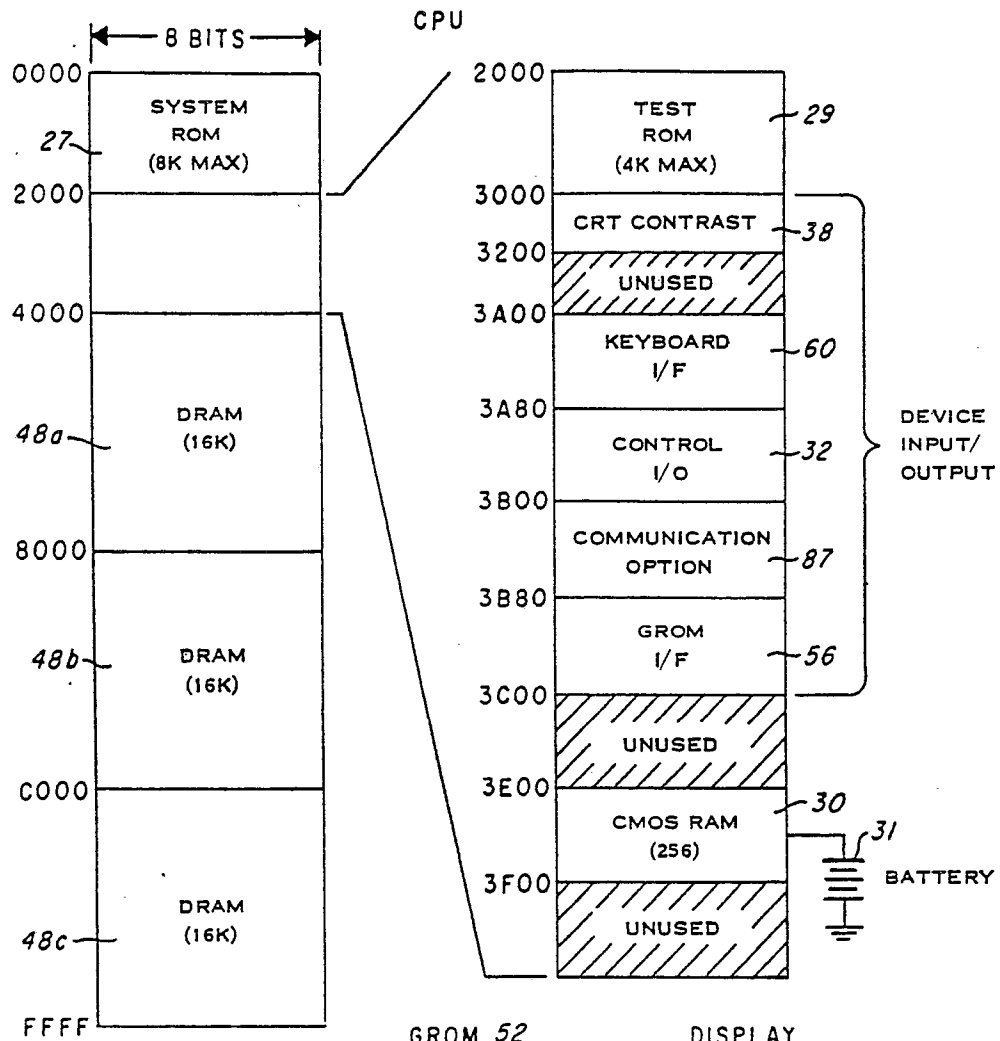


Fig. 1c





GROM 52  
(16K BYTE EACH)

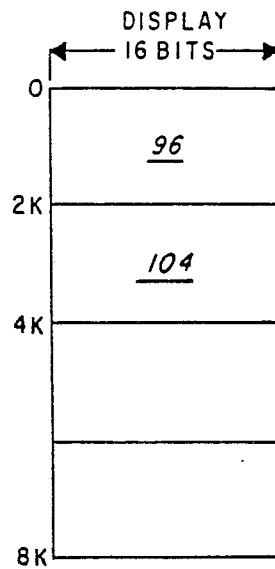
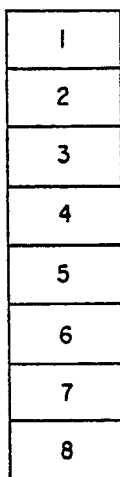


Fig. 2

Fig. 3

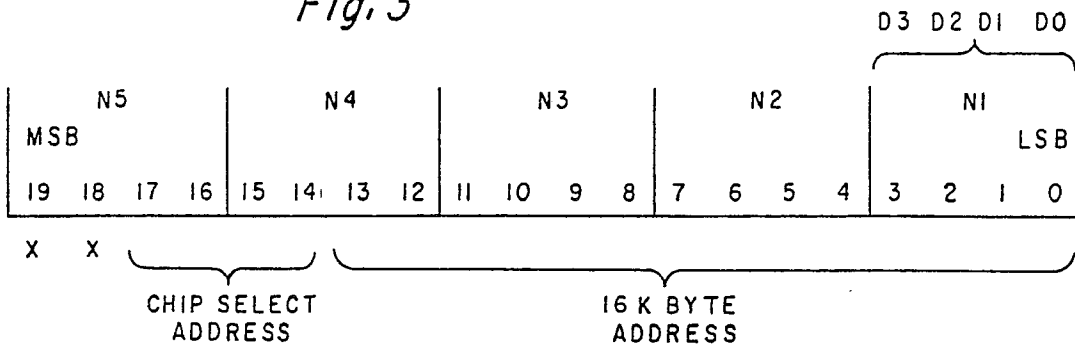
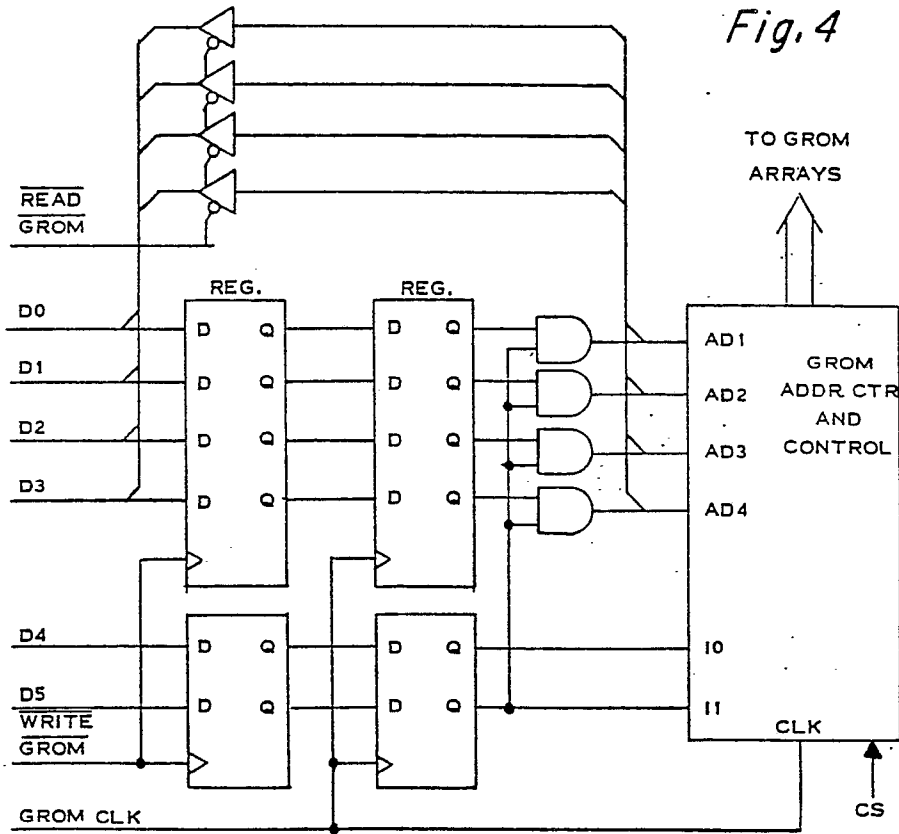


Fig. 4



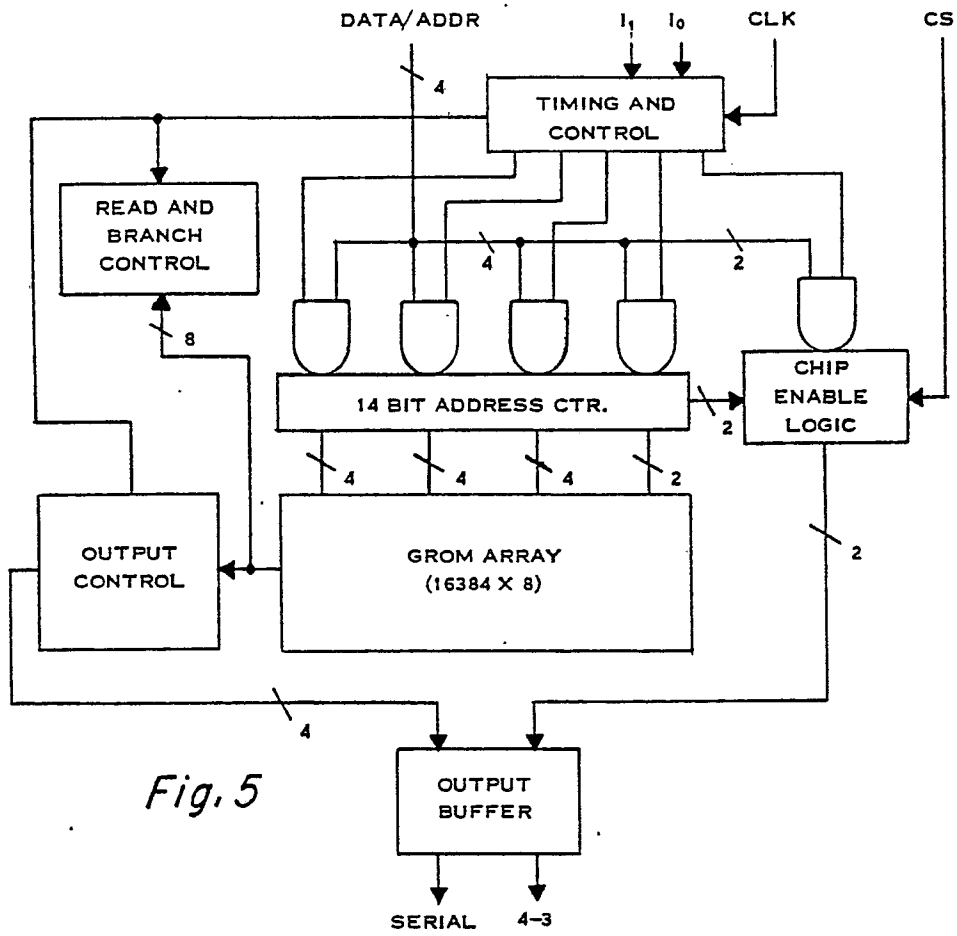
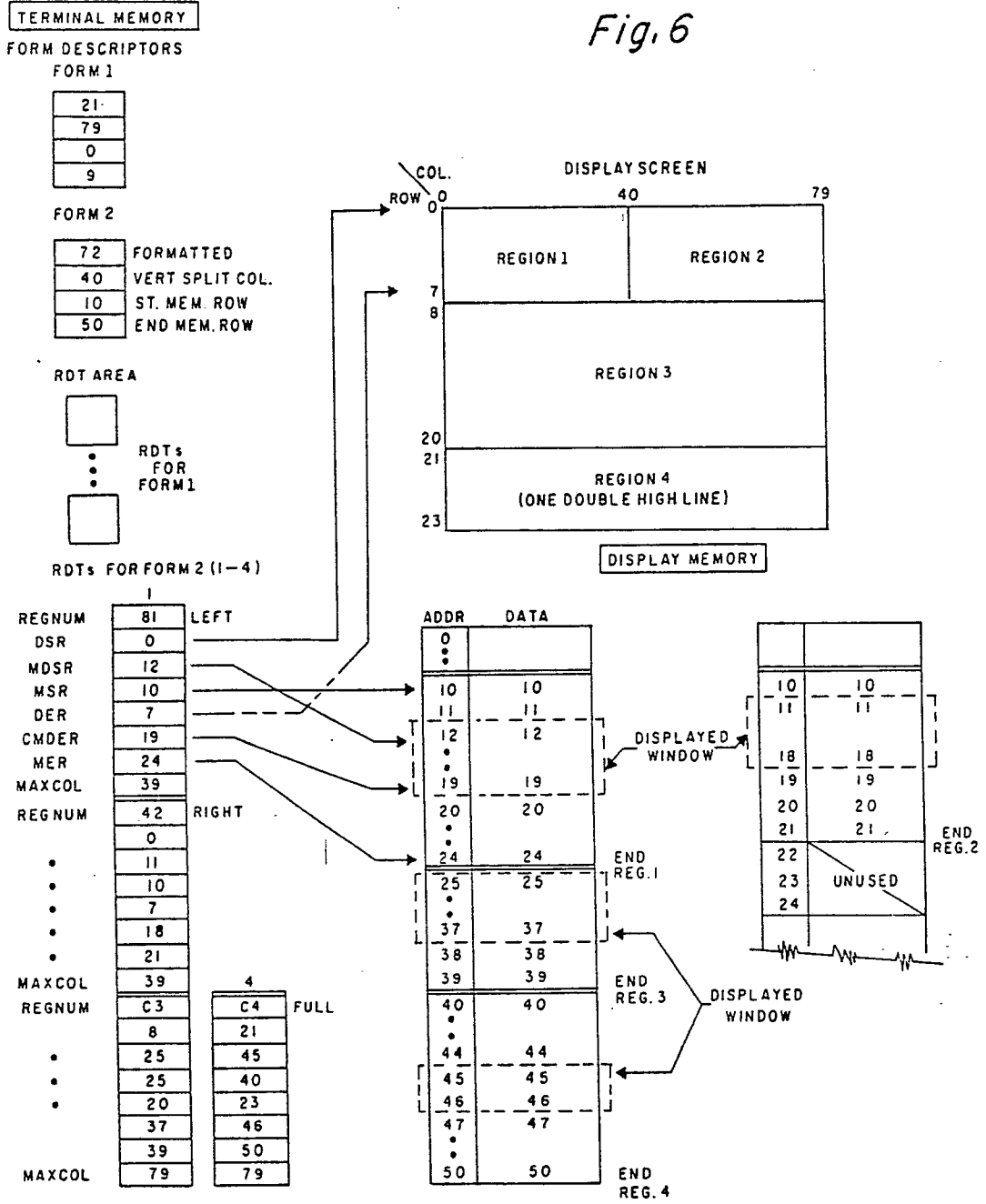


Fig. 5

Fig. 6



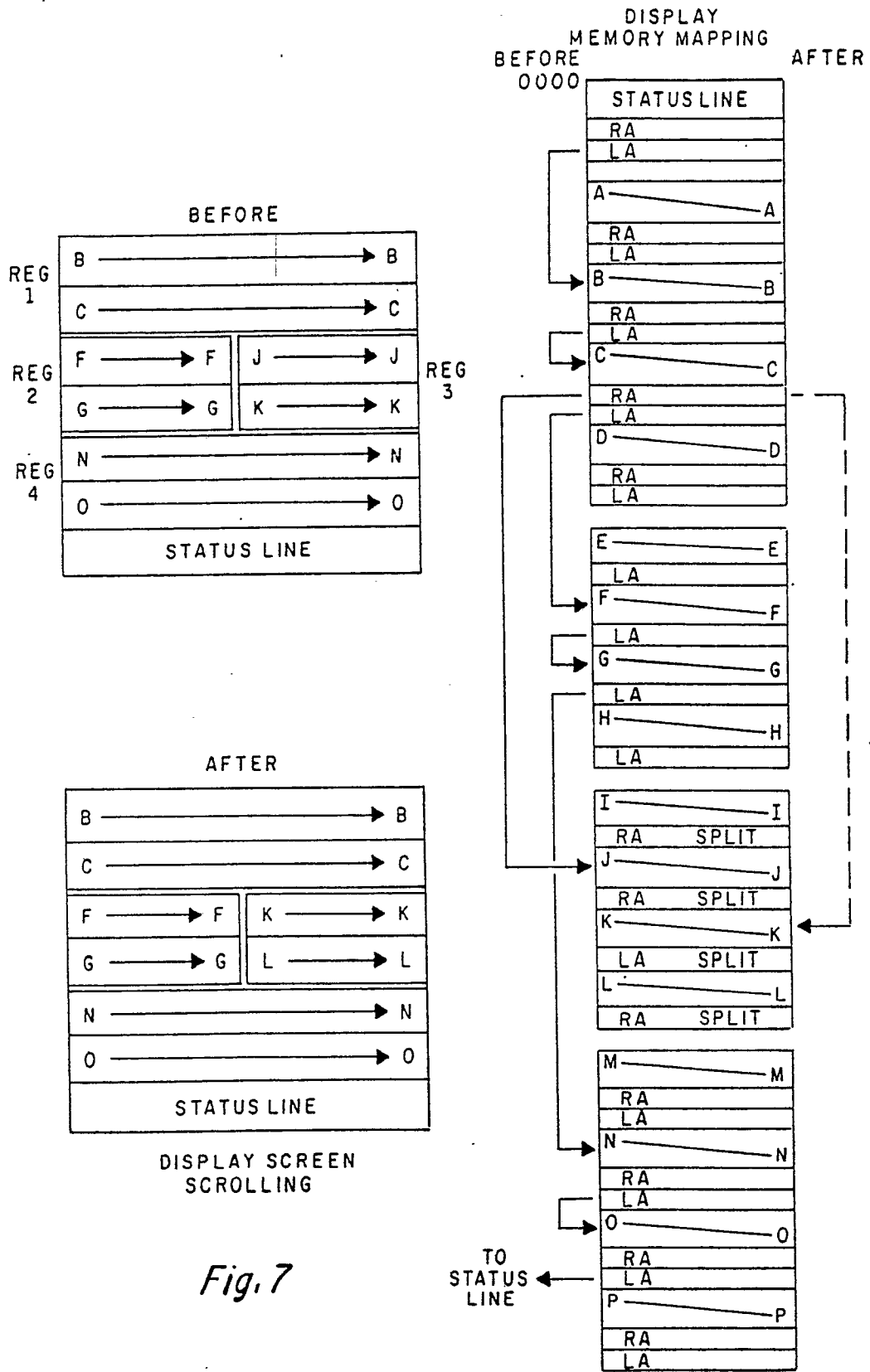


Fig. 7

Fig. 8a

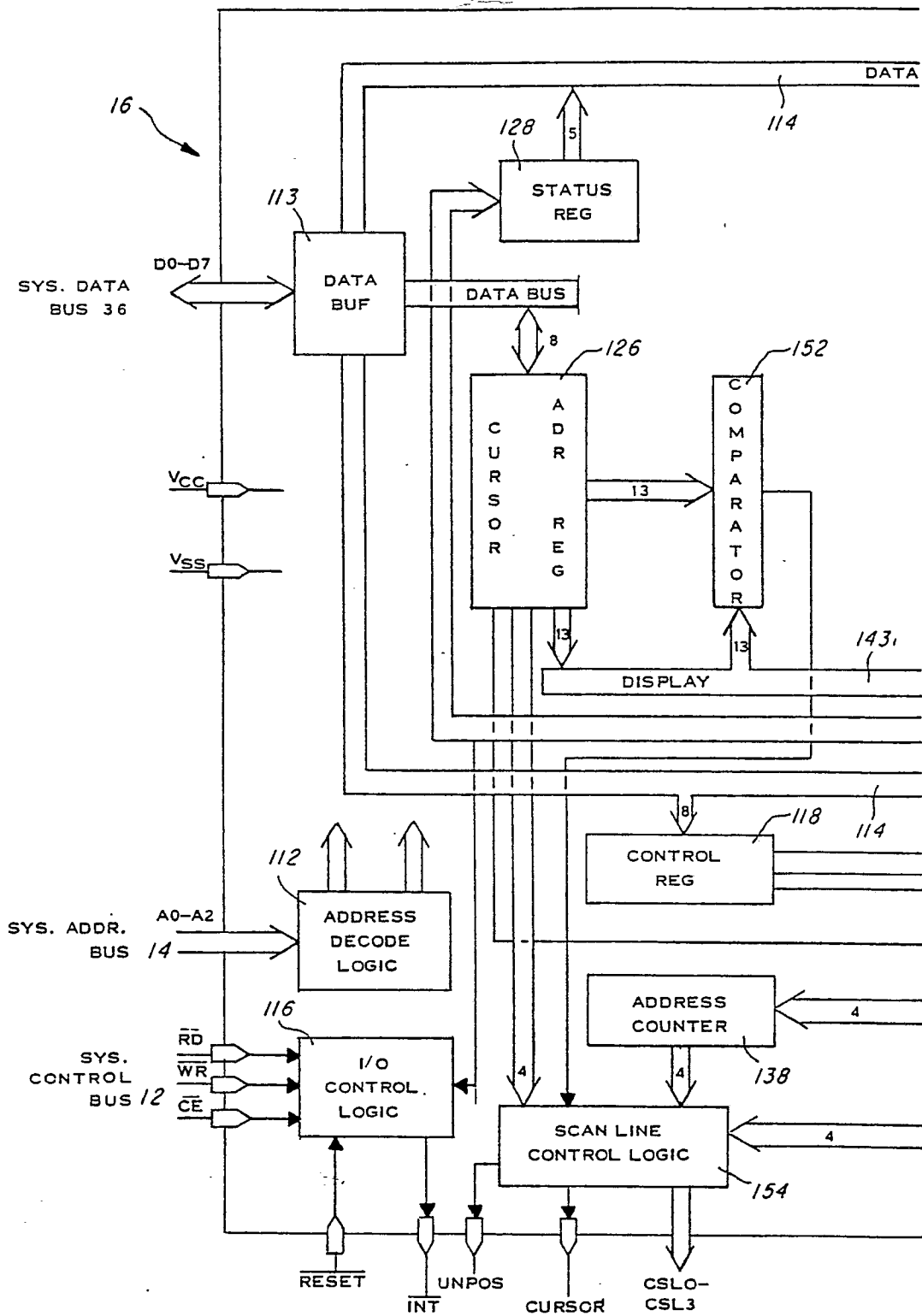


Fig. 8b

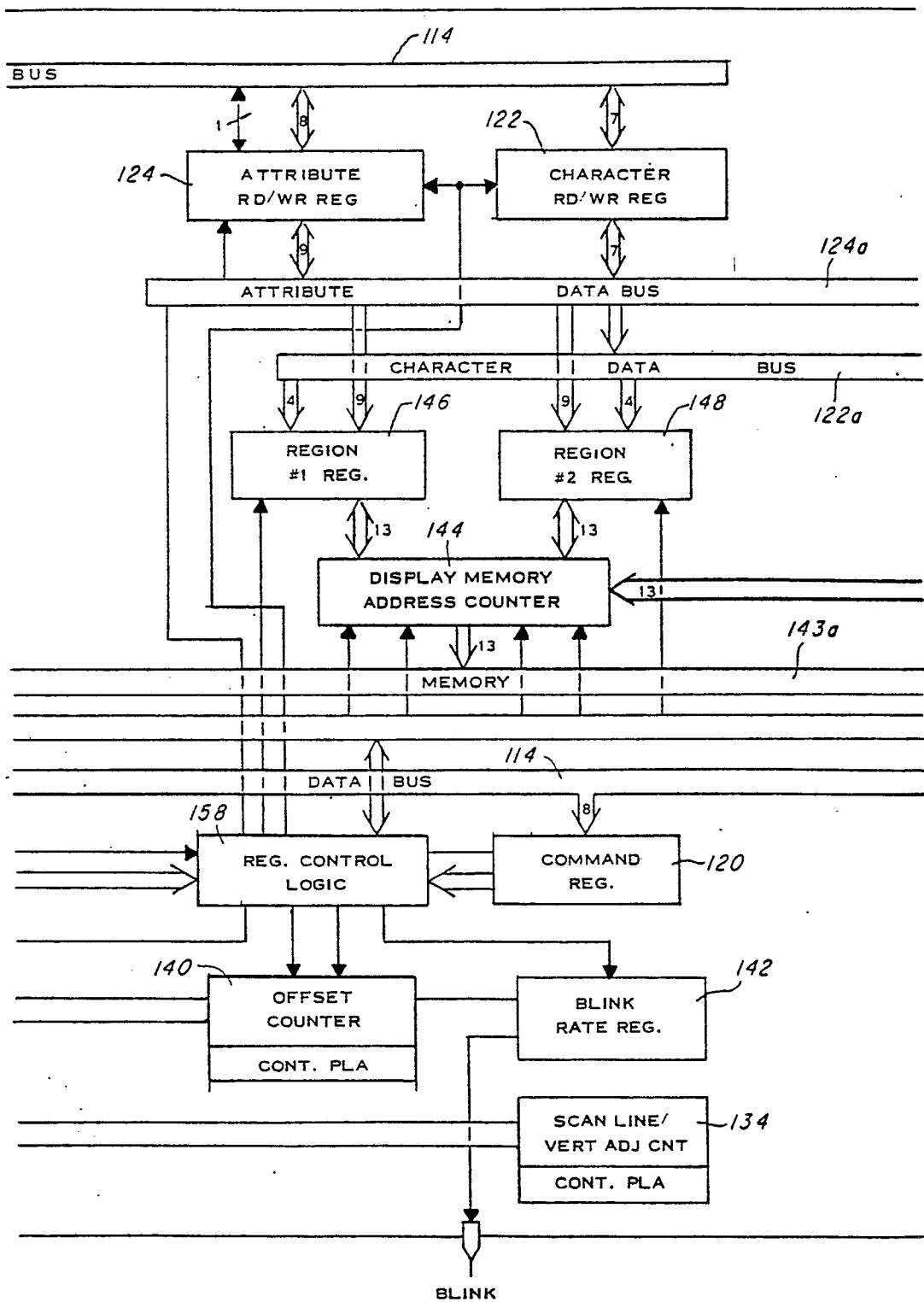


Fig. 8c

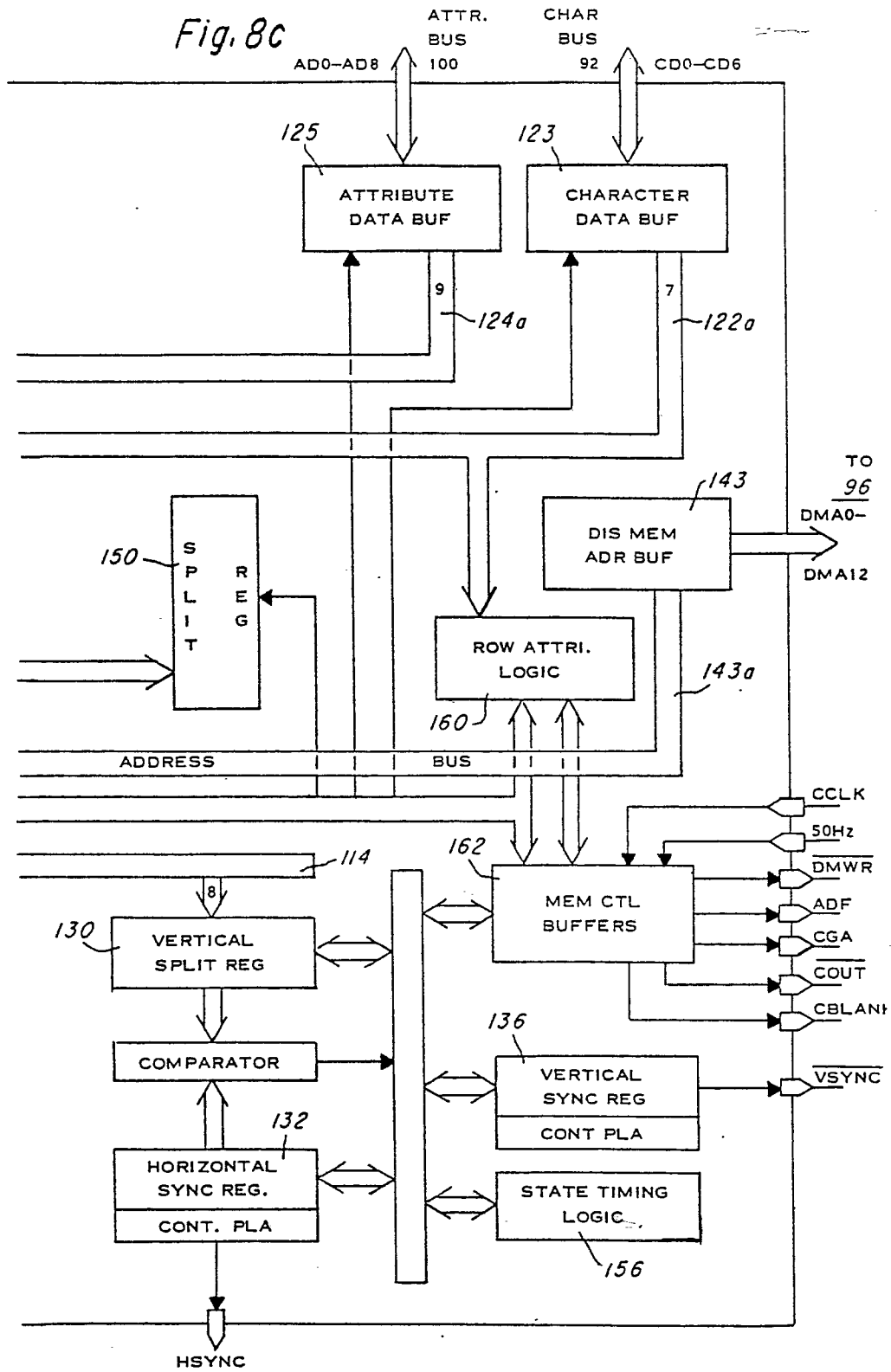


Fig. 9

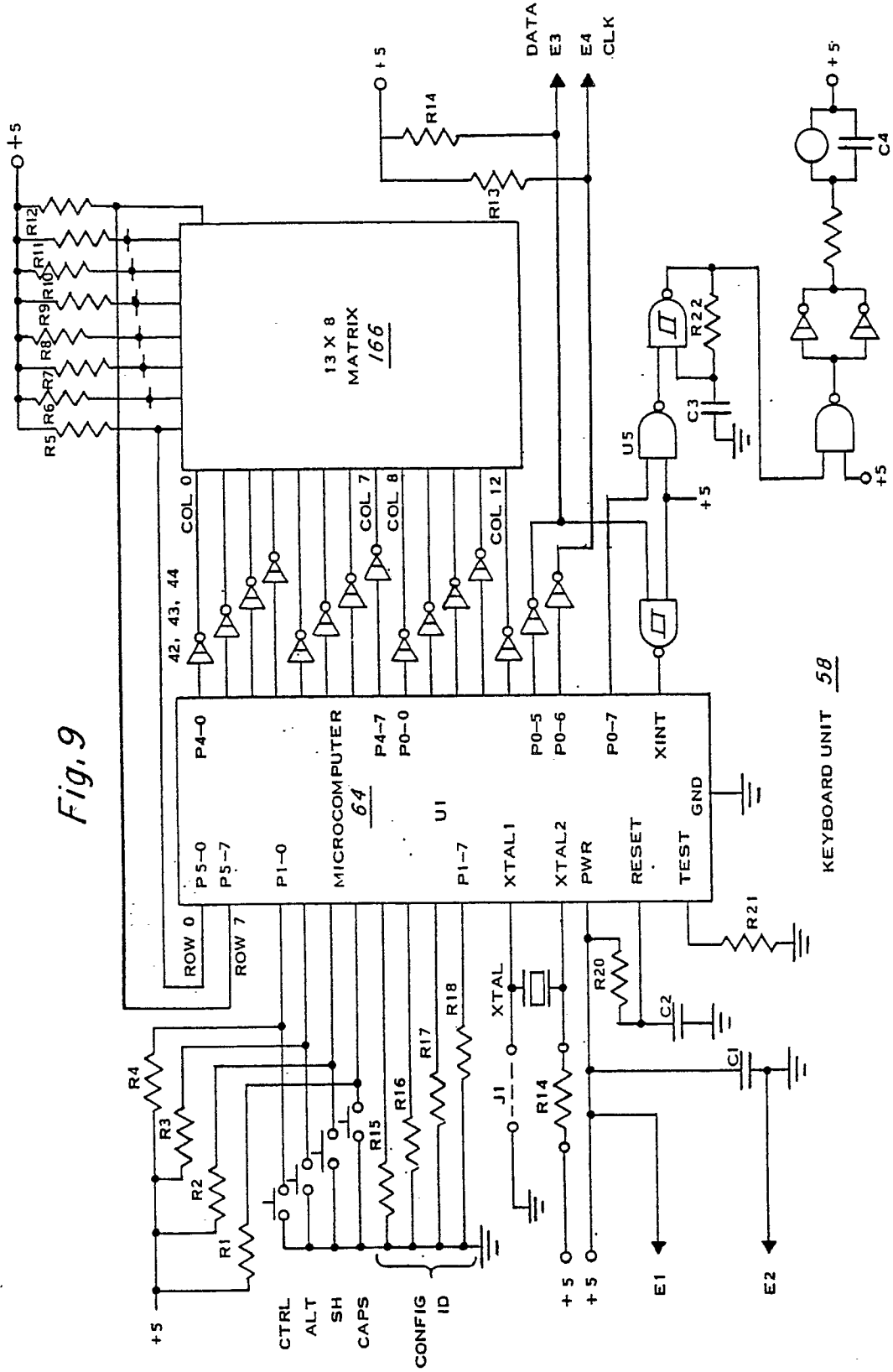




Fig. 10a

