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

(11) Publication number : **0 474 366 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number : **91307222.9**

(51) Int. Cl.^{5} : **G09G 1/16**

(22) Date of filing : **06.08.91**

(30) Priority : **06.08.90 US 563469**
06.08.90 US 563472

(43) Date of publication of application :
11.03.92 Bulletin 92/11

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

(71) Applicant : **TEXAS INSTRUMENTS INCORPORATED**
13500 North Central Expressway
Dallas Texas 75265 (US)

(72) Inventor : **Guillemaud, Andre J.**
3819 Stovepipe Lane
Sugar Land, TX 77479 (US)
Inventor : **Balistreri, Anthony M.**
4944 Woodway, Apt. No. 23
Houston, TX 77056 (US)
Inventor : **Guttag, Karl M.**
4015 South Sandy Court
Missouri City, TX 77459 (US)
Inventor : **Simpson, Richard D.**
16 Pavenham Road
Carlton, Bedford (GB)

(74) Representative : **Nettleton, John Victor et al**
Abel & Imray Northumberland House 303-306
High Holborn
London, WC1V 7LH (GB)

(54) Graphics display system including a video random access memory with a split serial register and a run counter.

(57) A graphics system includes a random access memory (105) with a split serial register (109) having low and high halves of a plurality of storage elements, an access start point address register (137), and an arrangement (140, 142, 145) for stopping an access run at the end of a predetermined run length. Thus, there are specified both start and stop point addresses for a read data access operation from the split serial register in a graphics processing system. By using these start and stop point addresses, the operating speed of the graphics processing system is increased. The random access memory also includes a multiplexer (160) for coupling a column of storage cells from the memory array to storage elements of the split serial register (109). Data stored in either a low half or a high half of the addresses of the memory array may be selectively coupled through the multiplexer to either a low half or a high half of the split serial register. For a tile oriented graphics display operation, this arrangement increases the number of choices of where within the random access memory array to store specific bits of the tile data to be displayed. Data representing a tile can be mapped into a single row of the random access memory array.

Jouve, 18, rue Saint-Denis, 75001 PARIS

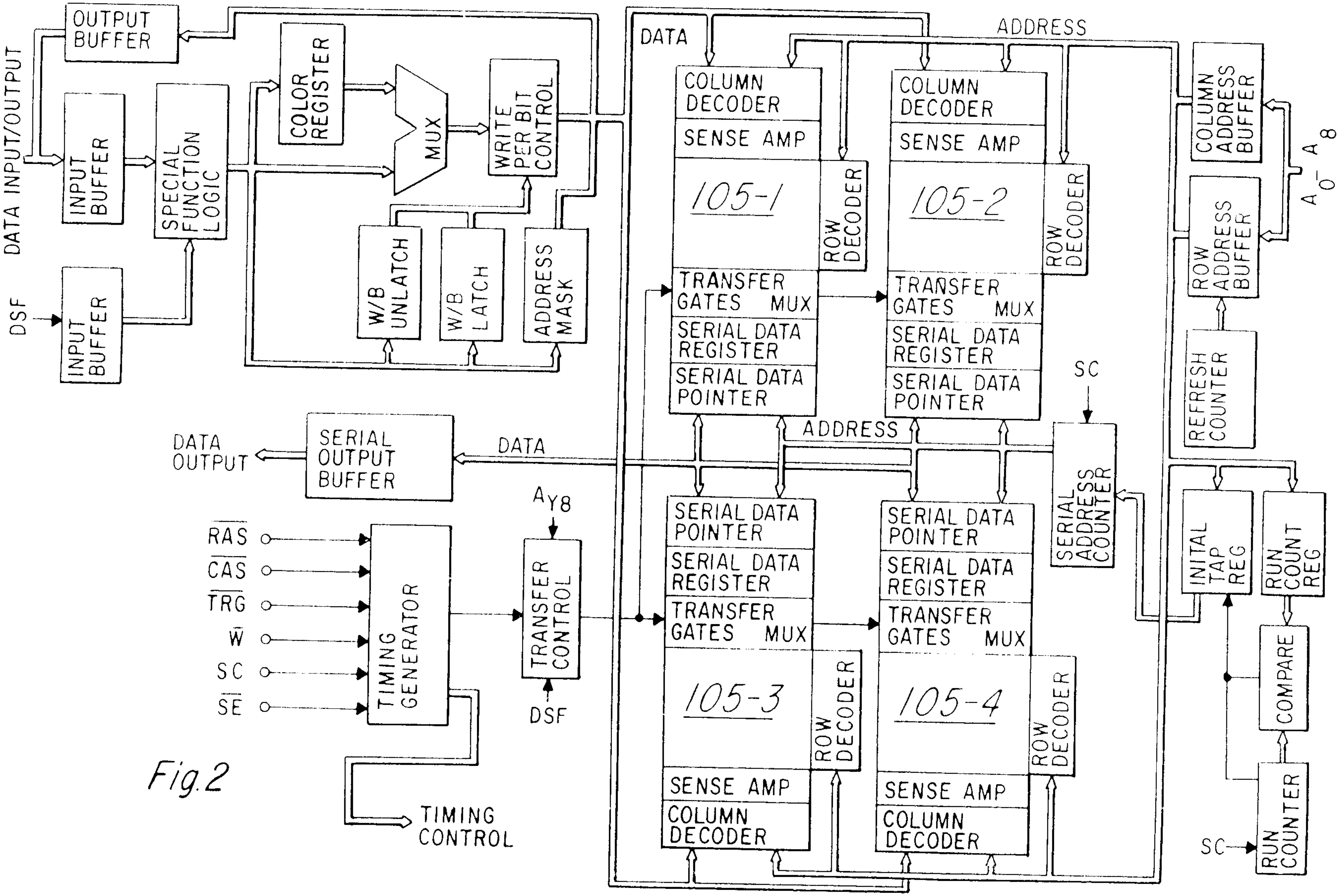

DATA INPUT/OUTPUT
OUTPUT BUFFER
INPUT BUFFER
SPECIAL FUNCTION LOGIC
DSF
INPUT BUFFER
COLOR REGISTER
MUX
WRITE PER BIT CONTROL
W/B UNLATCH
W/B LATCH
ADDRESS MASK
DATA
ADDRESS
COLUMN DECODER
SENSE AMP
105-1
ROW DECODER
TRANSFER GATES MUX
SERIAL DATA REGISTER
SERIAL DATA POINTER
COLUMN DECODER
SENSE AMP
105-2
ROW DECODER
TRANSFER GATES MUX
SERIAL DATA REGISTER
SERIAL DATA POINTER
COLUMN ADDRESS BUFFER
ROW ADDRESS BUFFER
$A_0 - A_8$
REFRESH COUNTER
SC
SERIAL ADDRESS COUNTER
ADDRESS
DATA
DATA OUTPUT
SERIAL OUTPUT BUFFER
A_{Y8}
TRANSFER CONTROL
DSF
$\overline{RAS}$
$\overline{CAS}$
$\overline{TRG}$
$\overline{W}$
SC
$\overline{SE}$
TIMING GENERATOR
TIMING CONTROL
SERIAL DATA POINTER
SERIAL DATA REGISTER
TRANSFER GATES MUX
105-3
ROW DECODER
SENSE AMP
COLUMN DECODER
SERIAL DATA POINTER
SERIAL DATA REGISTER
TRANSFER GATES MUX
105-4
ROW DECODER
SENSE AMP
COLUMN DECODER
INITAL TAP REG
RUN COUNT REG
COMPARE
SC
RUN COUNTER

Fig.2

RELATED CASES

This application contains subject matter disclosed in copending applications Serial No. 387,569 (TI-14315), filed 28 July 1989, Serial No. 07/563,469 (TI-14755), filed concurrently herewith, Serial No. 07/563,471 (TI-15211), filed concurrently herewith, and in the U.S. Patent Nos. 4,281,401, 4,639,890, and 4,747,081. The above mentioned applications and patents are incorporated herein by reference.

BACKGROUND OF INVENTION

Field of Invention

The present invention pertains to graphics display and processing systems for producing a graphics display generally and particularly using a semiconductor memory in the form of a random access memory with a split serial register arrangement.

Description of the Related Art

In a typical random access memory, the time required for accessing a row is approximately twice as long as the time required for accessing a column. For speedy overall operation, it is desirable to access a row and hold it while several of the columns are accessed along that row. This is referred to as "page mode" operation. If all of the columns across the array are accessed during a single row access, time saved may be as much as 50-70 percent. Such time saving may be achieved during memory write or during memory read operations. For any specific system design there may be trade-offs between memory write time and memory read time to achieve operating efficiency.

For computer graphics system applications, further trade-offs can be made between memory read and write times and the capability and speed of a microprocessor used for generating the data to be presented on a video screen. Other factors to be considered include the width and height of the display, the size of the video random access memory and whether the system is line oriented or tile oriented. Final choices among the several trade-offs is a matter left for the system designer or user.

Suppliers of devices which may be used in such applications prefer to manufacture devices that include several optional features so that designers and users will apply each type of device in many different applications. This enables one device design effort to result in large manufacturing runs and low costs per device.

In furtherance of the aforementioned design strategy, integrated circuit manufacturers have been improving the designs of video random access memory integrated circuit devices by adding features which increase the flexibility and speed of operation of those devices. Some of the features which have been added to random access memory designs include the addition of split serial registers to improve access to and from the memory array and selectable taps for readout from the split serial registers. These features enable the user to closely pack data into the video random access memory and to thereby avoid wasting memory space.

As previously mentioned, there are two modes of operation, i.e., line oriented and tile oriented. In line oriented operation, the graphics processor generates and stores data in sequential order as it will appear line-by-line on the display. Storage of data in the video random access memory and read out from storage to the display are accomplished in serial order bit-by-bit and line-by-line. Read out from the split serial register to the display is timed to correlate with the display sweep signal. Prior video random access memory device designs have features which are very desirable and are in wide spread use in line oriented systems.

Prior video random access memory device designs are much less flexible and useful with respect to tile oriented systems. Typically, a tile oriented display is divided into a grid of equal size and shape areas, called tiles. The size and shape of the tiles are factors which are among the factors to be chosen by the system designer or user. For instance, the tiles may be squares or rectangles. If rectangles, they may be oriented with the long dimension either laying horizontally or standing vertically. For some tile sizes and orientations, currently available video random access memory devices lack sufficient flexibility for efficient use in some possible system applications. More specifically, the data representing a single tile must be stored in different rows of the storage cells of the random access memory array.

Prior to the advent of split serial registers, graphics processing systems transferred an entire row of data bits from a dynamic random access memory into a long data register for readout. From the data register, the entire row of data bits was read out in serial sequence. We now know that this operation was significantly limiting the speed of some graphics processing systems because output of the data register was a serial bit stream read out under control of a serial clock. During the long readout operation, no new data could be transferred into the data register from the dynamic random access memory because the new data would corrupt the data existing in the data register before that existing data could be read out.

With the advent of the use of a split serial register in graphics processing systems, it became possible to define a high-half and a low-half of the split serial register through reference to the most significant bit in each half of a counter used for addressing the split serial register. Then, while accessing data from one half of the split serial register, the other half can be reloaded from the random access memory. Also during the loading of half of the split serial register, the address, within that half of the split serial register from which serial accessing will begin, is loaded into a start point address register. While the start point address of a data access run can be defined by information loaded into the start point register, the access run of the data readout ended at the midpoint of the split serial register when reading from the low-half or at the very end of the split serial register when reading from the high-half. This aspect of operating with a split serial register is a significant problem or limitation for some graphics processing systems designs.

For example, for a split serial register storing 512 bits, the end point for an access run in the low-half of the split serial register would be at bit 255 and the end point for an access run in the high-half of the split serial register would be at bit 511. Since typical useful access runs for a graphics processing system often vary from eight bits to 128 bits, the forcing of access runs to end either at bit 255 or at bit 511, unnecessarily slows the operating speed of the graphics processing system.

SUMMARY OF INVENTION

These and other problems are solved by a graphics display and processing system including a video random access memory with a split serial register having low and high halves of a plurality of storage elements, an access start point address register, and an arrangement for stopping an access run at the end of a desired run length.

A method and apparatus define both start and stop points of a data access operation, i.e., the specific run length, for a split serial register of a graphics processing system. The address of the access start point in the split register is loaded into the access start point address register. A run length is loaded into a run length counter. Together that information determines the start and stop point addresses for each access operation of the split serial register.

There is an operational advantage for the graphics processing system being able to start and stop read access operations at selected storage elements in the split serial register. The graphics display and processing system operates at a faster speed.

The graphics display and processing system has a random access memory arranged with a novel split register and a multiplexer for transferring data from the memory array to the split register. Data stored in either a low half or a high half of the memory array column addresses may be selectively transferred through the multiplexer to either a low half or a high half of the split register addresses.

An advantage of this arrangement is that system designers have greater flexibility, or more options, particularly in a tile oriented operation, in deciding where within the memory array to store specific bits of information to be displayed, particularly in a tile oriented operation. A tile of data can be mapped into the storage elements along a single row of the random access memory array. Thereafter the data from the storage elements of the random access memory are transferred into the split register and transmitted in line-by-line sequence to a raster scan display.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the invention may be gained by reading the following detailed description with reference to the drawing wherein:

Figure 1 is a block diagram of part of a graphics processing system;

Figure 2 is a block diagram of circuits on a video random access memory integrated circuit chip;

Figure 3 is a block diagram of a random access memory system arranged with a multiplexer between a memory array and a split serial register;

Figure 4 is a schematic diagram of an illustrative multiplexer for coupling either a low-half or a high-half of the memory array addresses to either a low-half or a high-half of the split serial register addresses;

Figure 5 is a truth table for defining the operation of the multiplexer of Figure 3;

Figure 6 is a map showing where tile data may be stored in the memory array for defining a screen of information in an exemplary tile oriented system operation;

Figure 7 is a table showing a sequence for transferring data from the memory cells of the random access memory array and storing it in the storage elements of the split serial register;

Figures 8 through 11 are block diagrams of a selected row of memory, a multiplexer, a split serial register and control circuits in four different operating conditions;

Figure 12 is a time line showing a sequence for moving data from the split serial register to the display; and

Figure 13 is a map showing where tile information is presented on a display screen.

Figure 14 is a block diagram of a random access memory system arranged with a multiplexer between a memory array and a split shift register.

DETAILED DESCRIPTION

Referring now to Figure 1, there is shown a block diagram of a data processing system 100 including a graphics display arrangement for presenting information. A more complete description of the arrangement and operation of the system of Figure 1 can be found in a patent application serial No. 821,641, filed January 23, 1986, which is incorporated herein by reference.

The data processing system 100 includes a host processing system 102, a graphics processor 103, such as a Texas Instruments Incorporated TMS34010 or TMS34020 Graphics System Processor, a video random access memory 105, a data register 107, a video palette 108, a digital to video converter 110, and a video display 112.

Host processing system 102 provides the major computational capacity for the data processing system 100. Included in the host processing system 102 are a processor, an input device, a long term storage device, a read only memory, a random access memory and assorted peripheral devices that form a computer system. Arrangement and operation of the host processing system are considered to be conventional. As a result of its processing functions, the host processing system 102 determines the information content of the graphic display to be presented on a screen for the user.

Graphics processor 103 provides the major portion of data manipulation for producing the particular graphics display to be presented on the screen. The graphics processor 103 is bi-directionally coupled to the host processing system 102 by way of a host bus 101. In the arrangement of Figure 1, graphics processor 103 operates independently from the host processing system 102. The graphics processor 103, however, is responsive to requests from the host processing system 102. Graphics processor 103 also communicates with memory 105 and the video palette 108 by way of a memory bus 104. Data to be stored in the video random access memory 105 is controlled by the graphics processor 103. The graphics processor, in turn, may be controlled either in part or wholly by a program stored in the video random access memory 105 or in a read only memory 114. Read only memory 114 may store various types of graphic image data.

Additionally the graphics processor 103 controls data stored within the video palette 108 and by way of a video control bus 116 the operation of the digital to video converter 110. Through the digital to video converter, the graphics processor 103 can control the line length and the number of lines per frame of the video graphic image. Significantly, the graphics processor 103 determines and controls where graphic display information is stored in the video random access memory 105. Subsequently, during readout from the video random access memory 105, the graphics processor determines the readout sequence from the video random access memory and the split serial register 107, the addresses to be accessed, and control information required to produce the desired graphic image on the display 112.

Video random access memory 105 stores the bit mapped graphics data which define the graphics image to be presented to the user. Control of the transfer of the data from the video random access memory 105 through the data register 107, the video palette 108, and the digital to video converter 110 to the display 112 is provided by the graphics processor 103. Video data output from the video random access memory 105 is transferred by way of a video output bus 118 to the data register 107 where it is assembled into a display bit stream. The data register 107 may be a shift register.

Storage elements of the data register 107 may be fabricated of either dynamic or static electronic circuits. Alternative choices of storage elements include any bistable electronic, magnetic, optical, or optoelectronic device with sufficient operating speed.

In accordance with a typical arrangement of the video random access memory 105, there is a bank of several separate random access memory integrated circuits. Storage cells of the video random access memory 105 may be fabricated as either dynamic or static electronic circuits. For a single readout access operation, only one bit of data is readout from a selected storage element of each of the integrated circuits. Thus a group of bits, including one bit from each of the several separate integrated circuits, are read out at once. The data register 107 assembles the display bit stream for transmission by way of a lead 120 to the video palette 108. Although the foregoing describes the video random access memory 105 as an electronic circuit, the invention may also be carried out by a memory fabricated as any bistable electronic, magnetic, optical or optoelectronic device with sufficient speed.

Under control of information from the graphics processor 103, the video palette 108, such as a Texas Instruments TMS34070 Video Palette, converts the data received from the data register 107 into video level signals on a bus 125. This conversion is accomplished through a look-up table. The video level signal output from the video palette 108 may include color, saturation, and brightness information.

Digital to video converter 110 receives the digital video signals from the video palette 108 and, under control of signals received by way of the video control bus 116, converts the digital video signals into analog levels which are applied to the video display 112 via an output line 127. The number of pixels per horizontal line and the number of lines per display are determined by the graphics processor 103. Also, the synchronization, retrace, and blanking signals are controlled by the graphics processor 103. Altogether, this group of signals specify the desired video output to the video display 112.

Video display 112 produces the specified video image for viewing by the user. There are two techniques which are used widely. The first technique specifies video data in terms of color, hue, brightness, and saturation for each pixel. For the second technique, color levels of red, blue and green are specified for each pixel. The video palette 108, the digital to video converter 110, and the video display are designed and fabricated to be compatible with the selected technique.

Referring now to Figure 2, there is shown a block diagram layout of an integrated circuit video random access memory 105. Included within the memory 105 are four arrays of memory cells 105-1, 105-2, 105-3, 105-4. The cells are arranged in rows and columns. For random access, row and column addresses are applied to the integrated circuit by way of address leads and an address bus. For random access, row and column addresses are decoded respectively by row and column decoders. Data is written from a data bus through the column decoders and sense amplifiers to the selected cells of the memory arrays. For serial readout, the data is read from a selected row of the memory arrays, through transfer gates to the data registers. A serial address counter applies a series of serial data register addresses to the serial data pointers, or decoder. In response to those addresses, sequences of data are transmitted from the serial data registers through a data bus to a serial output buffer. From there the data proceeds to the register 107, the video palette, the digital-to-video signal converter, and to the video display, as shown in Figure 1.

The various random access input and output circuits are replicated four times to complete the random access arrangement.

For serial output operation, a group of common circuits controls the readout operation very effectively. The group of circuits includes an initial tap register, a run count register, a run counter, a comparator, and a serial address counter which applies a series of addresses through another address bus to serial data pointers associated with each serial data register. Operation of these common circuits for serial readout are described at length hereinafter.

Referring now to Figure 3, there is shown a more detailed block diagram of the graphics processor 103, the video random access memory 105-1, a split register 109-1, some control circuitry, and interconnecting busses and leads. Video random access memory 105 is an exemplary memory including four memory arrays 105-1, 105-2, 105-3, and 105-4 of rows and columns of storage cells. Typically the four memory arrays are included on one semiconductor chip. Information representing a single pixel for the display includes several bits of data. There is one bit, e.g., B0, stored in each of the arrays for one pixel. All of those bits for the one pixel are stored at the same row address and the same column address so that they can be either written into or read out from the whole memory in one access operation. In any specific design, there usually are as many memory arrays as there are bits in a pixel. If more than four bits are needed per pixel, either more arrays per chip or more chips may be provided.

Although the four video random access memory arrays 105-1, 105-2, 105-3, and 105-4 are shown ganged together in Figure 3, only one of them, 105-1, will be discussed hereinafter for purposes of simplifying both the drawing and the description without compromising any generality. It is to be understood that what is shown and said about the one memory array 105-1 and associated multiplexer 130-1, split register 109-1 and control circuits also applies to the other memory arrays and associated circuits which are ganged together from one or more semiconductor chips.

The screen of the video display 112 of Figure 1 can be thought to be organized in either of two ways. Usually in the display, there are a large number of horizontal lines each including a large number of pixels. Another often used, but different, scheme divides the display into a number of tiles. Each tile includes some number of pixels in a contiguous area of the display. The tile areas for a display are uniform in size and therefore include some number of pixels across and some number of pixels in height. For purposes of subsequent discussion, the number of pixels across a tile equals the number of columns in a section of the video random access memory array 105-1 and the number of storage elements in a section of the split register 109-1. The split register 109-1 may be either a shift register or a serial register. Hereinafter the split register will be described as split serial register 109-1. Information in one line across a tile is considered to be a segment of the tile. Since there is only one bit per pixel in the memory array 105-1, the number of bits in a segment of the tile in array 105-1 equals the number of columns used in the sections of the memory array and the number of storage elements used in the sections of the split serial register 109-1. The number of pixels in height equals the number of lines in the tile. For a subsequent illustrative example, a tile is thirty-two pixels across and eight pixels high in the display depicted in Figure 12, to be discussed subsequently.

The video random access memory array 105-1 is arranged so that the data, representing a tile of the display, is stored as sequential segments of the tile in a single row of the memory array 105-1. An entire tile of data is stored in the single row of the memory array 105-1. Row and column address information, together with the desired display, or pixel information, is generated by the graphics processor 103.

For random access writing, the addresses are applied through the bus 104 to an address register 106 for accessing the identified row and column storage locations in the random access memory array 105-1. The display data, to be stored at each address, also is applied through the bus 104 and a lead 111 to the random access memory array 105-1. When the graphics processor develops the random access address and display data for any tile, that data, in a departure from the prior art, is transmitted through bus 104 and is written into the random access memory 105-1 by making a single row access and thereafter accessing sequentially selected columns where data is to be stored. This enables a random access write-in operation to write in during a single row access as much as a whole tile of data. Since the accessing of a row operation takes approximately twice as long as accessing a column, considerable operational time is saved by making the single row access and performing the several column accesses while the row access remains effective.

The memory array 105-1 and the split serial register 109-1 are each divided by address into a low part and a high part. Such divisions greatly facilitate readout from the memory array 105-1 to the video display 112 of Figure 1. Readout from the memory array 105-1 occurs sequentially by line of the display. As the raster scans a line of the display, the appropriate data for each pixel in the sequence is applied to the beam for projection on the display screen. The graphics processor 103 determines the order for addressing the storage cells of the memory array 105-1, the selection made by the multiplexer 130-1, and the order for reading data from the storage elements of the split serial register 109-1 to achieve the desired output sequence of information being forwarded to the video display 112.

Rows of the memory are addressed by part rows, e.g., low address half rows and high address half rows. Under control of the graphics processor 103, the multiplexer 130-1 determines whether the data read out of either the low-half or the high-half of the memory array is transmitted by bit line to either the low-half or the high-half addresses of the split serial register 109-1. Once the data is stored in the split serial register 109-1, one or more bits of data or segments of tiles can be transmitted from the split serial register 109-1 out to the video display 112.

Additional control circuitry, which is common to the several memory arrays, is provided for determining the specific portion of data to be transmitted from the split serial register 109-1. The data from the split serial register 109-1 can be read out from a separate tap at each storage element. Conceptually, the taps are represented by a gate circuit 132 which receives a separate output from each storage element of the split serial register 109-1. A counter decoder 135 determines which one of the split serial register storage element outputs is transmitted to the video display 112 during any time slot determined by a readout clock signal CLOCK .

Since the initial pixel data to be transmitted may be in any split serial register 109-1 location, the graphics processor 103 loads the address of that initial pixel data into an initial tap, or start point, register 137. Upon receipt of a reset signal from a comparator 145, the initial tap register 137 loads the initial pixel data address into the counter decoder 135 for enabling the gate circuit 132 to transmit the data from the correct split serial register storage element. Loading of the initial pixel data address may be in parallel to the counter decoder 135.

Data read out thereafter generally proceeds sequentially along the storage elements of the split serial register 109-1, however, unlike the prior art, it does not necessarily proceed continuously. We have found that in some circumstances, it is advantageous to read data from only a portion of one half of the split serial register 109-1.One such circumstance arises when the display system is tile oriented and the data representing a tile is stored along a single row of the memory array 105-1, as previously described herein. Then for efficiency, the sequential addressing for reading out the data from the split serial register 109-1 should be interrupted before the end of half the register. In a departure from the prior art, interruptions are made by the graphics processor 103 loading a number, or run count, into a run count register 140 for determining the number of sequential addresses of the split serial register 109-1 that are accessed before the interruption, i.e., before the counter decoder 135 and gate circuit 132 jump to a new initial, or start, tap address for subsequent readout. A convenient run length is equal to the number of pixels in a segment. The number in the count register 140 and a run count from a run counter 142 are compared by the comparator 145. When they do not match, the current sequence of addresses continues as the run counter is incremented by the signal CLOCK for each readout operation. When the two counts do agree, e.g., at the end of a segment, the comparator 145 produces a signal for resetting the run counter 142 and for loading the new initial tap address into the initial tap register 137. As a result, the counter decoder 135 enables the gate circuit 132 to jump to the new initial tap address of the split serial register 109-1 and thereby interrupt the sequential addressing.

It is advantageous for a tile oriented system to run through a sequence of addresses equal to the number of addresses needed for reading out all of the pixels in a segment of a tile from one half of the split serial register 109-1 and then transfer to the other half of the split serial register to read out the pixels in a segment from another tile. Since the tiles typically end before the end of a half of the register, it is advantageous to interrupt the readout operation at the end of the tile rather than at the end of half of the register. A more detailed description of this operation is to be described hereinafter in reference to Figures 5 through 12.

Referring now to Figure 4, there is shown the physical arrangements of the storage cells and bit lines of the memory array 105-1 and of the storage elements of the associated split serial register 109-1. The memory array 105-1 and the split serial register 109-1 both are divided into parts by addresses, a low-half L of the memory array addresses including half of the storage cells and a high-half H of the memory array addresses including the remaining storage cells. The low-half designators L0...L127 are shown below the high-half designators H0...H127 so that, at a glance, the two halves will stand out from each other in Figure 4.

In a departure from prior memory arrays, the storage cells L of the low-half addresses of the memory array 105-1 are interleaved with the storage cells H of the high-half addresses of the memory array. Similarly, in a departure from prior random access memory system arrangements, the storage elements L of the low-half addresses of the split serial register 109-1 are interleaved with the storage elements H of the high-half addresses of the split serial register 109-1.

Such interleaving of the storage cells of the two parts of the memory columns and of the storage elements of the two parts of the split serial register 109-1 provides a significant advantage when the memory array 105-1 and the split serial register 109-1 are fabricated as an integrated circuit. By interleaving, the total length and complexity of the bit lines, which are arranged for coupling the storage cells of the memory array 105-1 through the multiplexer 130-1 to the storage elements of the split serial register 109-1, are substantially reduced with respect to the bit line layout, which otherwise is required when the lower and upper halves of the memory array 105-1 and split serial register 109-1 are divided by whole physical sections rather than by the interleaving of the columns and register elements, as shown in Figure 4.

Figure 4 additionally illustrates a schematic for a section 130-1.1 of the multiplexer 130-1 which can be used for transferring data from one cell in either half of the addresses of the memory array 105-1 to a storage element in either half of the addresses of the split serial register 109-1. The illustrated section 130-1.1 of the multiplexer 130-1 couples one column address of the high-half or one column address of the low-half of the memory array 105-1 to associated high or low half storage elements of the split serial register 109-1. The complete multiplexer 130-1 uses one of such multiplexer sections 130-1.1 for every pair of bit lines.

Referring now to Figure 5, there is shown a truth table that describes the logical operation of the section 130-1.1 of the multiplexer 130-1 shown in Figure 4. A one-out-of-four code, received from the graphics processor by way of a path 160 of Figure 3, is applied to the control terminals H-H, L-L, L-H and H-L of the illustrative switching devices. In response to the one-out-of-four code, one gate of the multiplexer section 130-1.1 is enabled. The other three gates remain disabled. Input control signals are designated low-half-to-low-half, L-L; low-half-to-high-half, L-H; high-half-to-low-half, H-L; and high-half-to-high-half, H-H. Inputs are from the high-half H or low-half L of the memory array, and outputs go to the high-half H or low-half L of the split serial register.

It is noted that the multiplexer operations represented by the truth table of Figure 5 is not dependent upon the interleaving of the storage elements in the memory array or in the split serial register, as shown in Figure 4. The described multiplexer operations may be used for coupling the column addresses of other arrangements of split memory arrays to the addresses of the split serial register.

The graphics processor 103 of Figures 1 and 3 generates all of the information for the graphic display. Each bit of the data can be generated at any time and in any order. The processor 103 knows which bit is being generated at any time and where that bit is preassigned, or mapped, for storage in the random access memory 105. When randomly writing the bits into the random access memory 105, the order of writing the bits is unimportant except for efficiency considerations, but each bit must be stored in a storage element at its own preassigned, or bit mapped, location in the random access memory.

For efficient, speedy random writing of data into the memory array, it is effective to access for writing by tiles rather than by individual pixels. All changed information for one or more tiles is generated and written quickly. All stored information for other unchanged tiles need only be refreshed, not written again. By so writing only into tiles where information has changed, all of the writing can be accomplished very efficiently.

Referring now to Figure 6, there is shown a Cartesian coordinate map representing the storage locations in the random access memory array 105-1 presented in Figures 3 and 4. It is noted that the low-half and the high-half of the array are divided in the center rather than by interleaving the columns. This array is shown divided in the center for the purpose of illustrating the concept by a simplified drawing. Our preference in practice, however, is to interleave memory storage elements and the column leads by address, as shown in Figure 4. The interleaved arrangement can handle the high-half, low-half transfer operations in accordance with the subsequent discussion just as well as the illustrated random access memory array 105-1 of Figure 6.

The grid of Figure 6 is an 8 x 8 array of squares, each square of which represents the information to be presented as an eighth of a tile to be presented on the graphic display. In Figure 6 and subsequently in Figures 7-12, squares represent segments of data. Each segment is numbered so that the reader can follow numerically identified segments from storage in the memory array through the multiplexer and split serial register to the display, as subsequently discussed. The numbers in the squares identify the information as segments of the tiles. Each segment of data is stored in a section of a row of the memory. Data represented by each one of the numbered squares, or segments, includes thirty-two bits of data (i.e., 32 columns x 1 row) which are stored as shown in the memory array 105-1. A tile of the display image is represented by all of the bits stored in the memory cells along a single row of the random access memory array 105-1, e.g., row 1. It was noted previously that relating to the display of Figure 12, the tiles are thirty-two bits wide by eight bits high and are oriented in vertical columns in the display, as discussed subsequently with respect to Figure 12.

Recall that data can be written into the storage cells of the random access memory array 105-1 in any sequential order but each bit must be stored in its preassigned location. For the purpose of enhancing the overall speed of operation of the display system, the graphics processor creates the information for the display image by tiles and stores that information into the memory array. In Figure 6, all of the data included in each row of the memory array is equivalent to a full tile. As the graphics processor 103 creates all of the data making up a single tile, the graphics processor only accesses a single selected row in the memory array 105-1 one time. While that selected row is accessed, the columns are accessed one at a time until all of the new data, related to the one tile, is stored in the accessed row of the random access memory. Because the graphics processor 103 writes, or stores, all of the data contained in one tile while accessing only one row of the memory array, considerable system operating efficiency is achieved. Each row access generally takes as much as twice the time needed for each column access. For efficient operation, the worst case occurs when only a single column is accessed for each row access. Our arrangement can achieve as much as a seventy percent reduction in operating time with respect to the worst case for writing randomly into the random access memory.

After a complete display screen of data is stored in the random access memory 105 of Figure 1, the system can commence reading that data from the random access memory for transfer to the split serial register 109-1, the data register 107 and on to the video display 112. As shown in Figure 6, data is stored in the memory by segment. The video display uses a well-known raster scanning technique for presenting the graphic information on a display screen or cathode ray tube. Meanwhile, the graphics processor 103 scans the random access memory 105 for transferring data to the display. Data from the memory array is transferred through the multiplexer, the split serial register, the data register 107, and the color palette, and is coordinated with the raster beam as it sweeps across one horizontal line after another projecting the graphics information onto the screen at predetermined locations. The sequential order of readout of data from the memory array is fixed by the hardware and firmware of the graphics display system. The first data for display is segment 1, then segment 2, etc., until finally segment 64. While the data from one half row of the addresses of the data register 107 are being read out to the display 112, data from another half row of addresses can be transferred from the memory array 105 to the idle half of the split serial register 109-1. The information sent from the split serial register may be -and often is- less than an entire half register for every half register of data transferred thereto from the memory.

From the split serial register, the data is sent to the video palette by way of the data register 107 of Figure 1. Data register 107 is arranged to receive in parallel, the sequential data streams from each of the arrays 105-1, 105-2, 105-3, and 105-4 by way of the output leads 118-1, 118-2, 118-3, and 118-4 of Figure 3. In operation, the data register 107 takes in the plural parallel streams of input data, and shifts them out in one interleaved sequence. All of the bits for each pixel are grouped together in the output stream of data. Thus all of the data describing each pixel is applied to the video palette at once.

The video palette translates the pixel data into the desired form for application to the digital to video converter. From the digital to video converter, the video signals are applied to the display. At the finish of each line of the screen, the raster retraces, or returns, to the beginning side of the screen, typically, one or more lines lower. During retrace, graphic information is blanked from the beam. Once the retrace is completed, the raster commences sweeping across another line of the screen and projects the graphic display information. Because the raster beam sweeps across the entire screen and then retraces, the data read from the memory array must be presented to the beam modulator of the video display 112 in the proper sequential order for each complete sweep across the screen.

The operation of reading data from the memory and transferring it through the multiplexer, the serial shift register and the gate to the data register may be better understood by carefully reading the following description of Figures 7-13.

Referring now to Figure 7, there is shown a table representing the operation of sequentially reading the stored data, as presented in Figure 7, from the random access memory of Figures 3 and 4. In Figure 7, the leftmost column shows the time slots for reading segments of data from the memory array 105-1. The column labeled the LOW HALF of the register shows which groups of bits, or parts of tiles, are read out of the memory array and into the low-half of the split serial register 107-1 of Figure 4 in preparation for transfer of selected segments to the video display 112 during odd numbered time slots of the readout operation. Likewise, the column labeled HIGH HALF of the register shows the groups of bits that are read out of the memory array and into the high-half of the addresses of the split serial register 107-1 in preparation for transfer of other selected segments to the display during even numbered time slots. The column labeled MULTIPLEXER presents an indication of the control signal information for operating the multiplexer 130-1 for the transfer of data into the split serial register 107-1 before it is sent to the video display. In the columns labeled RAM and REGISTER, the letters designate, respectively, the low-half L or high-half H of the addresses of the memory array 105-1 and the split serial register 107-1 and thus designate the multiplexer control signalling used in Figure 4.

As shown in Figures 7 and 8, during read time slot 0, data from the low-half L of the first row of the memory array 105-1 is read out through the multiplexer to the low-half L of the split serial register 107-1. The multiplexer 130-1 is set for a low-half L to low-half L transfer (control signal L-L). There are thirty-two bits in each of the segments, or squares, 1, 9, 17 and 25. Those numbers are underscored to indicate that they are segments of tile 1. While that data is residing in the split serial register 107-1 during time slot 1, only the thirty-two bits representing segment 1 are forwarded to the video display 112. The numeral 1, representing the tile segment 1, is enclosed in a circle to indicate that segment is to be forwarded to the video display. The initial tap address, to be stored in the initial tap register 137, is the address of the first bit of segment 1. Thirty-two is stored into the run count register 140 so that the sequential read is interrupted when 32 bits are read out from the split serial register. Segments 9, 17 and 25 are not forwarded during time slot 1. Refer now to Figures 6 and 8 for the next step of the sequential readout. While the segments 1, 9, 17, and 25 are residing in the low-half of the register, data from the low-half L of the second row of the memory array 105-1 is transferred through the multiplexer to the high-half H of the split serial register 107-1. The multiplexer 130-1 is set for a low-half L to high-half H transfer (control signal L-H). Bits of segments 2, 10, 18 and 26 are transferred into the high-half of the split serial register 107-1. During time slot 2, the 32 bits representing the segment 2 are forwarded to the display device. This is accomplished by storing the appropriate start address and run count while the segments are being transferred into the split serial register. The number 2 is encircled to indicate that the segment 2 is so forwarded to the display in sequence following segment 1. In Figures 7-11 for each of the time slots only one segment number is encircled to indicate which tile segment is forwarded in the sequence going to the display. An appropriate start address and run count are stored and used for each time slot.

Figures 10 and 11 show additional examples of the data from rows 1 and 2 of the memory being transferred through the multiplexer to the split serial register for sending segments 57 and 58 to the display.

Referring now to Figure 8, there is shown a time line for the sequence of tile segments being forwarded out of the split serial register 107-1 of Figure 3 to the display. By forwarding the data stored in sequential rows of the low-half of the random access memory alternately to the low-half and the high-half of the split serial register, followed similarly by data stored in the high half of the random access memory, the tile segments can be transmitted to the video display 112 of Figure 1 in sequential order 1, 2, 3, . . . 64, as shown in Figure 12.

Referring now to Figure 13, there is shown a graphic representation of the stored data, from the memory array 105-1, as an image created by the tile segment information presented in raster scan sequence on the video display 112. The numbered segments of the data are presented to the video display in sequential order 1, 2, . . . 64. Each scan of the raster includes eight segments of the data, one segment for each of eight tiles per scan. There are eight tiles in a complete screen of the display. Each tile is in a separate column on the screen and may be identified by the number in the top row of its column. Thus the tiles are numbered 1, 2 . . . 8. It is noted that all of the data for each tile of the display screen is mapped from a different row of the memory array 105 of Figure 6.

Figure 14 shows an alternative arrangement of the serial readout circuits and of the serial readout control circuits for a system using a split shift register. It is noted that features which are the same as the features of Figure 3 are designated with the same numerical identifier. Other numerical identifiers are used with respect to different features.

When data is to be read out serially from the array of memory cells, control signals on the leads 160 cause the data from either the low-half or the high-half of the memory to be transferred to either the low-half or the high-half of the split shift register. The split shift register is arranged at every stage with a tap for outputting data. A tap address register 175 is loaded with the address of the tap from which the first bit of a data sequence and subsequent data is to be outputted. When a signal is applied from the comparator 145 to the tap address register 175, the tap address is loaded into the register decoder 192. The register decoder 192 stores the tap address for the duration of the current serial readout operation. A decoded signal is applied to the gate 132-1 to identify and enable the selected tap for reading out data from the associated shift register stage.

A shift clock signal, applied to the split shift register 191-1 and the run counter 142, shifts data along the stages of the shift register 191-1 and increments the count in the run counter 142. During each clock cycle, a new bit of data is stored in the shift register stage associated with the selected tap. That new bit is read out through the gate 132-1 and the lead 118-1 for transmission to the video display, not shown.

The comparator 145 continuously compares the count in the run counter 142 with the run count value stored in the run count register 140. Whenever they do not match, the comparator produces a low output signal. When they match, however, the comparator produces a high output signal. This high output signal is the signal which resets the run counter 142 to zero and enables the tap address register 175 to send a new tap address to the register decoder 192.

The foregoing description presents the arrangement and operation of an illustrative example of the invention. This exemplary arrangement and other arrangements, made obvious in view thereof, are considered to be within the scope of the appended claims.

Claims

1. A random access memory system comprising:

a data register including a plurality of data storage elements having sequential addresses, the storage elements being divided by address into a first part and a second part;

an address generator for sequentially addressing the storage elements;

a start address register for presetting the address generator to a first start address in the first part of the storage element addresses; and

a control circuit for disabling the sequential addressing of the storage elements and for presetting the address generator to a second start address in the second part of the storage element addresses.

2. A random access memory system, in accordance with claim 1, wherein the control circuit comprises:

the start address register;

a run counter for counting the number of sequential addresses in an access operation; and

a comparator for determining when to stop the sequential addressing of the storage elements.

3. A random access memory system comprising:

a data register including a plurality of storage elements for storing data, the storage elements having sequential addresses and being divided by address into a high-half and a low-half of the data register;

a decoder for generating a sequence of storage element addresses from which to read the stored data;

an address register for applying a first start address to preset the decoder for initiating a sequence of low-half storage element addresses;

a counter for counting the number of sequential addresses in an access operation;

a comparator for determining when to stop the generation of the sequence of low-half storage element addresses; and a circuit, including the address register, for causing the decoder to jump to a second start address in the high-half storage element addresses.

4. A random access memory system comprising:

a memory array of storage cells arranged for storing data to be accessed by distinct addresses of the storage cells, the storage cells being divided by address into a high-half and low-half of the storage elements;

a data register having storage elements arranged for storing data to be accessed serially by address, the storage elements being divided by address into a high-half and a low-half of the storage elements;

a gate for transferring stored data from the storage cells to the storage elements;

an access circuit for generating a sequence of addresses of the storage elements;

a register for presetting in the access circuit a first address to initiate a sequence of addresses of the storage elements in the low-half of the data register; and

a control circuit for stopping the generation of the sequence of addresses and for presetting in the access circuit a second address in the high-half of the data register.

5. A random access memory system, in accordance with claim 4, wherein:

the storage cells and elements are dynamic memory devices.

6. A random access memory system, in accordance with claim 4, wherein:

the storage cells and elements are static memory devices.

7. A random access memory system, in accordance with claim 4, wherein:

the storage cells and elements are bistable electronic devices.

8. A random access memory system, in accordance with claim 4, wherein:

the storage cells and elements are bistable magnetic devices.

9. A random access memory system, in accordance with claim 4, wherein:

the storage cells and elements are bistable optical devices.

10. A random access memory system, in accordance with claim 4, wherein:

the storage cells and elements are bistable optoelectronic devices.

11. A graphics processing system comprising:

a video display including a screen;

a data processor for generating data signals representing information to be presented on the video display screen;

a memory array of storage cells for storing data signals received from the data processor;

a data register having high and low address parts for receiving and storing data signals transferred from the storage cells of the memory array;

an arrangement for specifying both a specific start address and a specific stop address in the data register to determine, respectively, where to start and stop a data access operation that sends data signals, from storage in the data register, to the video display screen.

12. A graphics display system comprising:

a data processor for producing data to be displayed in tiles;

a random access memory system arranged for storing a whole tile of data in a selected row of storage cells; and

access circuits for transferring a tile of data from the data processor into the selected row of the random access memory system.

13. A graphics display system, in accordance with claim 12, further comprising:

a plurality of column lines, coupled to the storage cells, including low half and high half column lines associated with respective storage cells;

the random access memory system including a split register, each half of such split register being arranged for receiving data simultaneously from half of the plurality of column lines of the random access memory system; and

a multiplexer arranged for transferring data from either a low half or a high half of the storage cells through the column lines to either a low half or a high half of the split register.

14. A graphics display system, in accordance with claim 12, further comprising:

a raster scan display device; and

a serial readout circuit for transferring data from the random access memory to the raster scan display device in a line-by-line sequence.

15. A graphics display system comprising:

a data processor for generating data for a display screen tile including a predetermined number of line segments;

a random access memory with information storage locations arranged in rows and columns; and

access circuits for transmitting the tile data to the random access memory and storing it therein in a single row of data storage locations for the predetermined number of line segments of the display screen tile.

16. A graphics display system, in accordance with claim 15, further comprising:

a plurality of column lines, coupled to the storage cells, including low half and high half storage cells;

the random access memory including a split register, each half of such split register being arranged for receiving at once data from half of the plurality of column lines of the random access memory; and

a multiplexer arranged for transferring data from either a low half or a high half of the storage cells through the column lines to either a low half or a high half of the split register.

17. A graphics display system comprising:

a processor for producing data to be displayed in tiles;

a random access memory with information storage locations arranged in addressable rows and columns for storing the tile data;

an address register for selecting addressable row and column information storage locations to store the tile data;

a bus for transmitting the storage location addresses and the tile data respectively, to the address register and the random access memory; and

the random access memory being further arranged for storing a whole tile of the tile data in a single addressable row of the information storage locations.

18. A graphics display system, in accordance with claim 17, further comprising:

the random access memory including a split register including storage elements and being arranged for receiving data concurrently from a half of the storage cells; and

a multiplexer arranged for transferring data from either a low half or a high half of the storage cells to either a low half or a high half of the storage elements of the split register.

19. A graphics display system, in accordance with claims 13, 16 or 18 further comprising:

a display device; and

a circuit for transferring data from the split register to the display device in a line-by-line sequence.

20. A graphics display system, in accordance with claims 1, 3, 4, 11, 13, 16 or 18, wherein:
the split register is a split serial register.

21. A graphics display system, in accordance with claims 1, 3, 4, 11, 13, 16 or 18 wherein:
the split register is a split serial register.

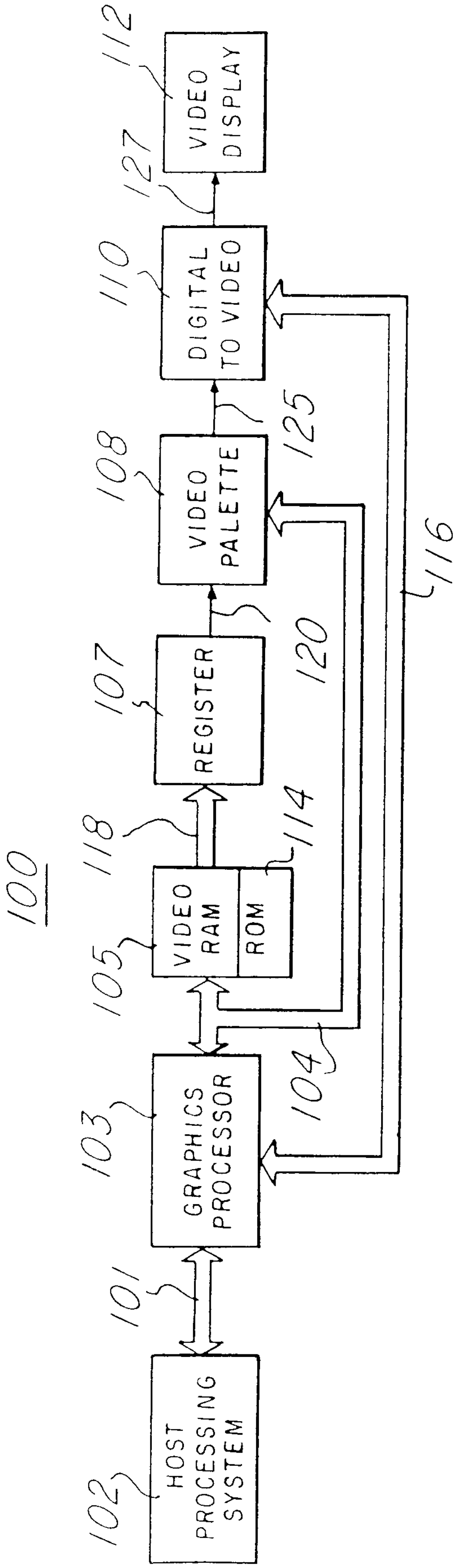

100
102
HOST PROCESSING SYSTEM
101
103
GRAPHICS PROCESSOR
104
105
VIDEO RAM
ROM
114
118
107
REGISTER
120
108
VIDEO PALETTE
125
110
DIGITAL TO VIDEO
116
127
112
VIDEO DISPLAY

Fig. 1

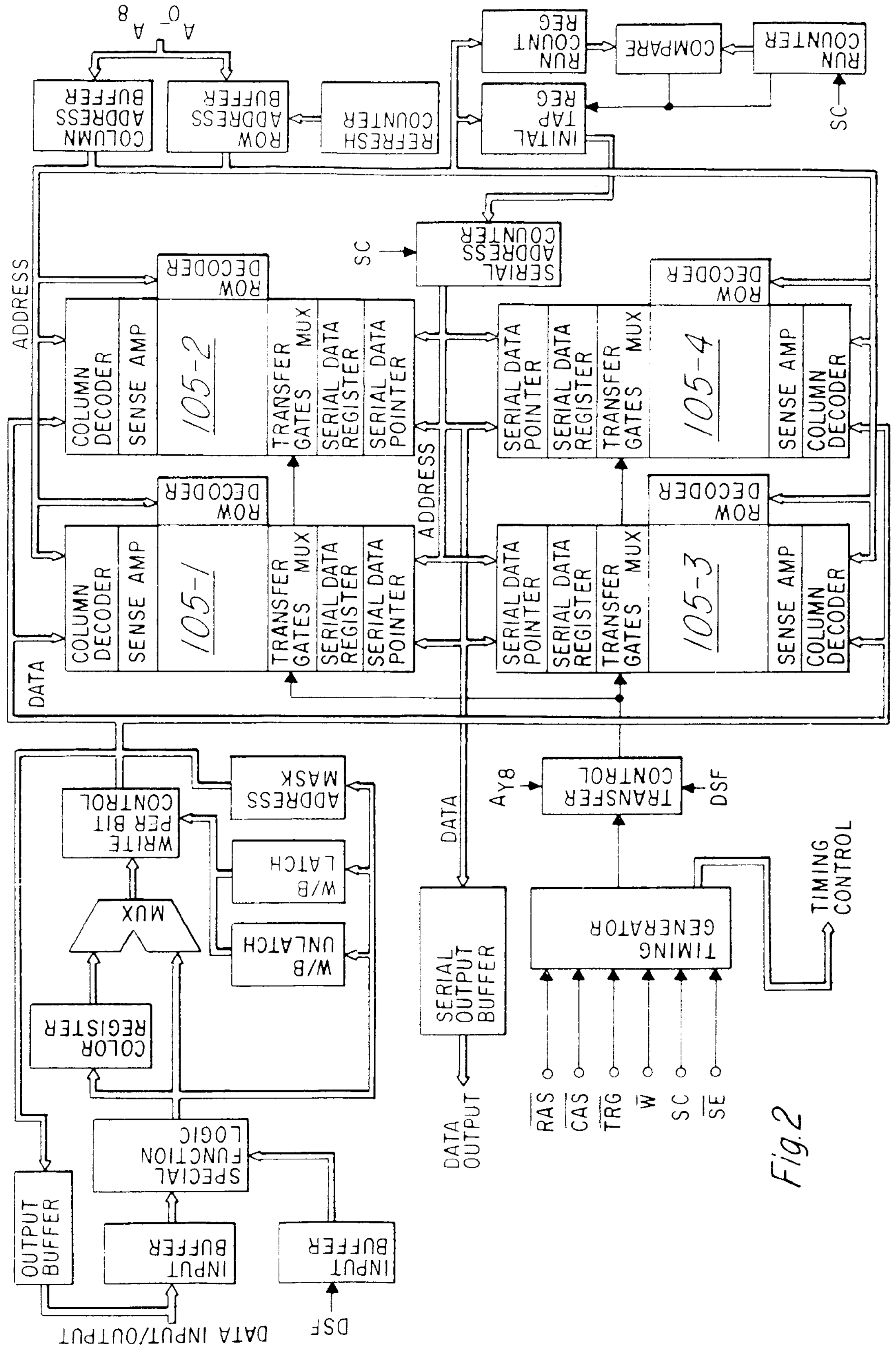

A0_A8
COLUMN ADDRESS BUFFER
ROW ADDRESS BUFFER
REFRESH COUNTER
RUN COUNT REG
COMPARE
RUN COUNTER
INITIAL TAP REG
SC
SERIAL ADDRESS COUNTER
SC
ADDRESS
ROW DECODER
COLUMN DECODER
SENSE AMP
105-2
TRANSFER GATES MUX
SERIAL DATA REGISTER
SERIAL DATA POINTER
ROW DECODER
COLUMN DECODER
SENSE AMP
105-1
TRANSFER GATES MUX
SERIAL DATA REGISTER
SERIAL DATA POINTER
ADDRESS
SERIAL DATA POINTER
SERIAL DATA REGISTER
TRANSFER GATES MUX
105-4
SENSE AMP
COLUMN DECODER
ROW DECODER
SERIAL DATA POINTER
SERIAL DATA REGISTER
TRANSFER GATES MUX
105-3
SENSE AMP
COLUMN DECODER
ROW DECODER
DATA
WRITE PER BIT CONTROL
ADDRESS MASK
W/B LATCH
W/B UNLATCH
MUX
COLOR REGISTER
SPECIAL FUNCTION LOGIC
OUTPUT BUFFER
INPUT BUFFER
INPUT BUFFER
DATA INPUT/OUTPUT
DSF
DATA
SERIAL OUTPUT BUFFER
DATA OUTPUT
AY8
TRANSFER CONTROL
DSF
TIMING GENERATOR
TIMING CONTROL
RAS
CAS
TRG
W
SC
SE

Fig.2

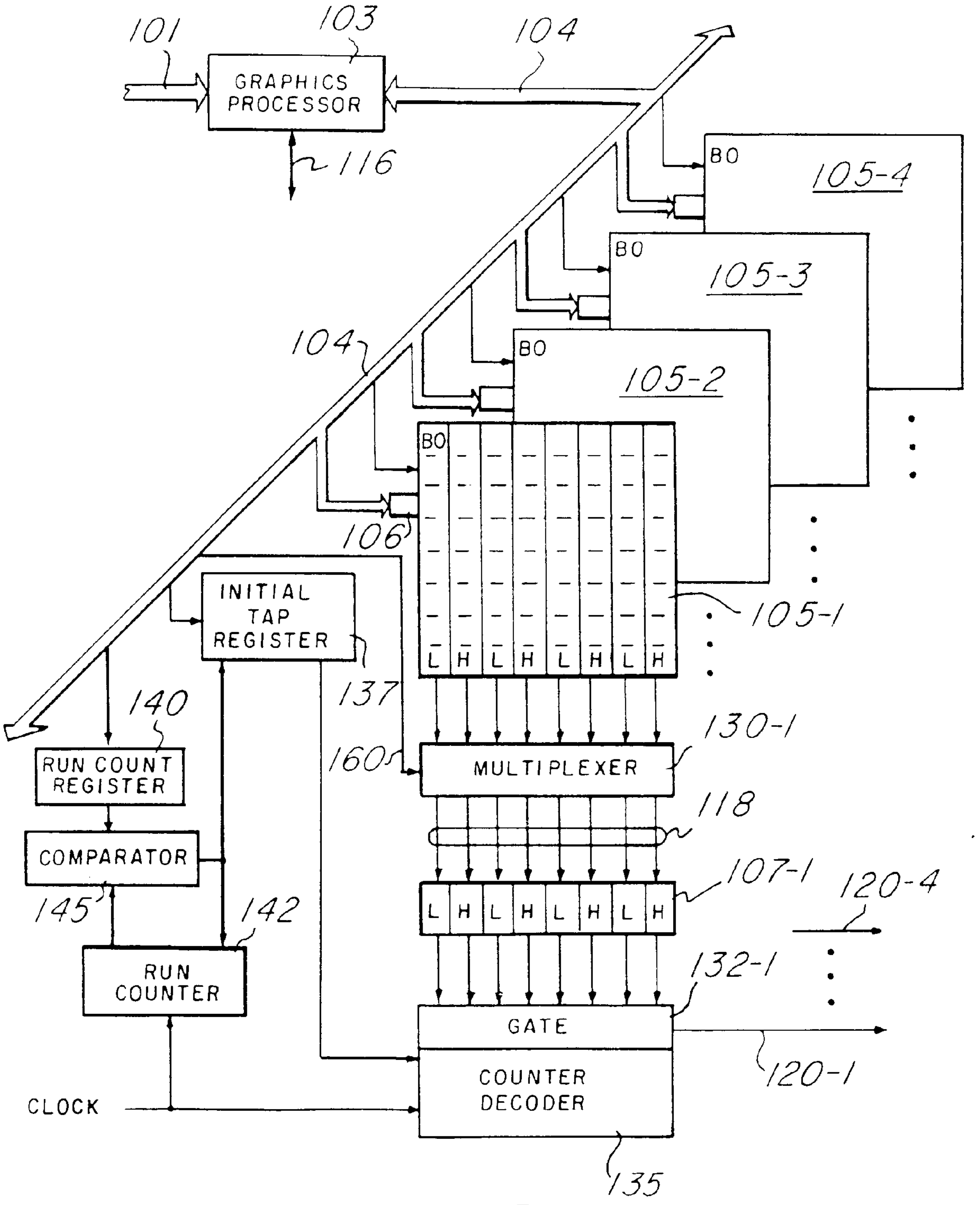

101
103
104
GRAPHICS PROCESSOR
116
BO
105-4
BO
105-3
BO
105-2
104
BO
106
105-1
L H L H L H L H
INITIAL TAP REGISTER
137
140
RUN COUNT REGISTER
COMPARATOR
145
142
RUN COUNTER
160
MULTIPLEXER
130-1
118
107-1
L H L H L H L H
120-4
132-1
GATE
COUNTER DECODER
120-1
CLOCK
135

Fig.3

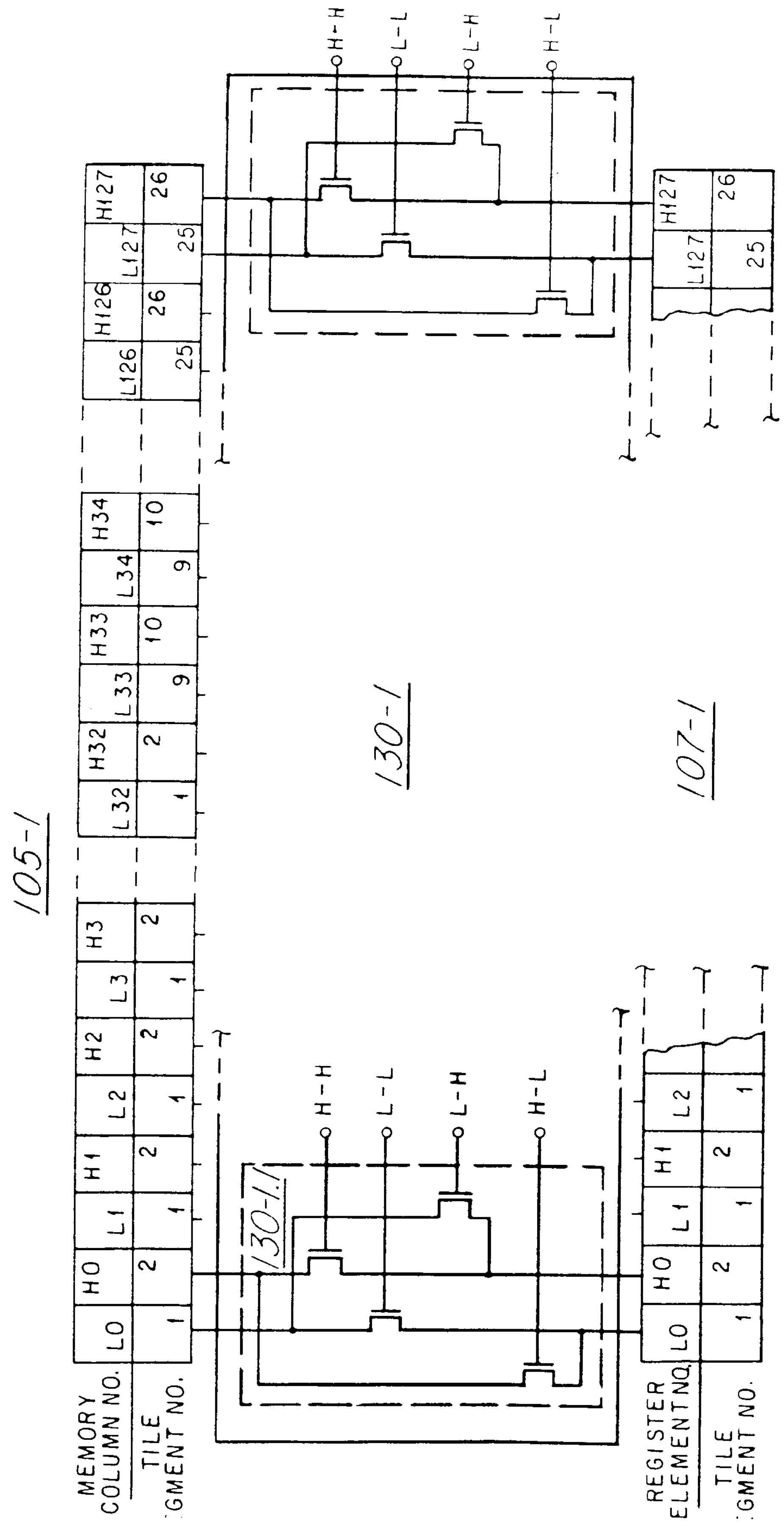
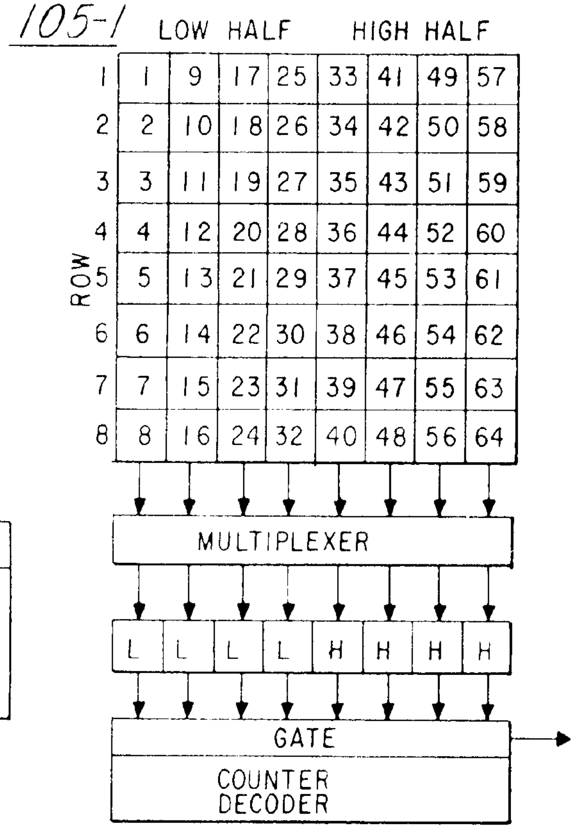

105-1
MEMORY COLUMN NO.
TILE
:GMENT NO.
L0
H0
L1
H1
L2
H2
L3
H3
L32
H32
L33
H33
L34
H34
L126
H126
L127
H127
1
2
1
2
1
2
1
2
1
2
9
10
9
10
25
26
25
26
130-1.1
H-H
L-L
L-H
H-L
H-H
L-L
L-H
H-L
130-1
107-1
REGISTER ELEMENT NO.
TILE
:GMENT NO.
L0
H0
L1
H1
L2
L127
H127
1
2
1
2
1
25
26

Fig.4

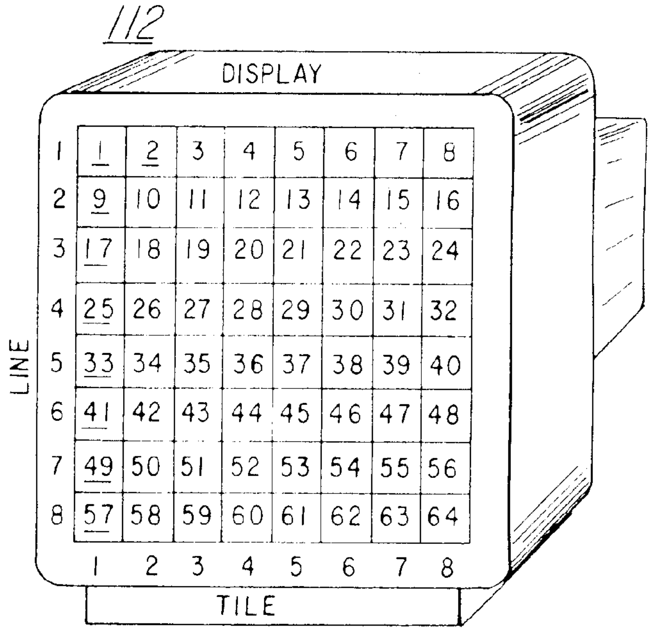
IN	H H	L L	L H	H L	OUT
H	1	0	0	0	H
L	0	1	0	0	L
L	0	0	1	0	H
H	0	0	0	1	L

Fig.5



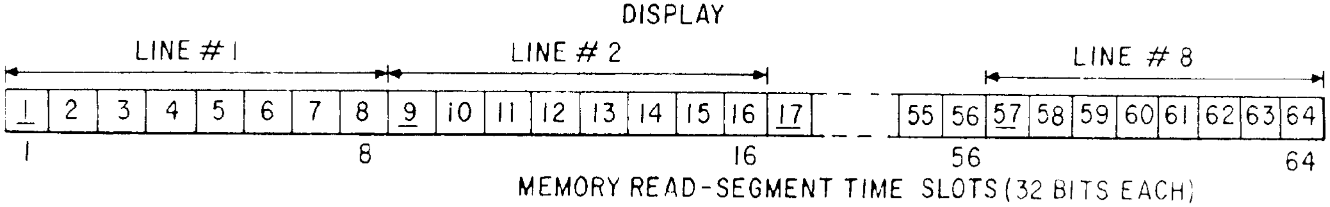
ROW	LOW HALF				HIGH HALF			
1	1	9	17	25	33	41	49	57
2	2	10	18	26	34	42	50	58
3	3	11	19	27	35	43	51	59
4	4	12	20	28	36	44	52	60
5	5	13	21	29	37	45	53	61
6	6	14	22	30	38	46	54	62
7	7	15	23	31	39	47	55	63
8	8	16	24	32	40	48	56	64

Fig.6



LINE	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	9	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23	24
4	25	26	27	28	29	30	31	32
5	33	34	35	36	37	38	39	40
6	41	42	43	44	45	46	47	48
7	49	50	51	52	53	54	55	56
8	57	58	59	60	61	62	63	64

Fig.13



MEMORY READ-SEGMENT TIME SLOTS (32 BITS EACH)

Fig.12

MEMORY READ TIME SLOTS	SPLIT SERIAL REGISTER								MULTIPLEXER RAM REGISTER		
	LOW HALF				HIGH HALF						
1	1	9	17	25					L	→	L
2					2	10	18	26	L	→	H
3	3	11	19	27					L	→	L
4					4	12	20	28	L	→	H
5	5	13	21	29					L	→	L
6					6	14	22	30	L	→	H
7	7	15	23	31					L	→	L
8					8	16	24	32	L	→	H
9	1	9	17	25					L	→	L
10					2	10	18	26	L	→	H
11	3	11	19	27					L	→	L
12					4	12	20	28	L	→	H
13	5	13	21	29					L	→	L
14					6	14	22	30	L	→	H
15	7	15	23	31					L	→	L
16					8	16	24	32	L	→	H
17	1	9	17	25	•	•	•	•	L	→	L
•	•	•		•	•	•	•	•	•		•
•	•	•	•	•	•	•	•	•	•		•
•	•	•	•	•	•	•	•	•	•		•
•	•	•	•	•	•	•	•	•	•		•
55	39	47	55	63					H	→	L
56					40	48	56	64	H	→	H
57	33	41	49	57					H	→	L
58					34	42	50	58	H	→	H
59	35	43	51	59					H	→	L
60					36	44	52	60	H	→	H
61	37	45	53	61					H	→	L
62					38	46	54	62	H	→	H
63	39	47	55	63					H	→	L
64					40	48	56	64	H	→	H

Fig. 7

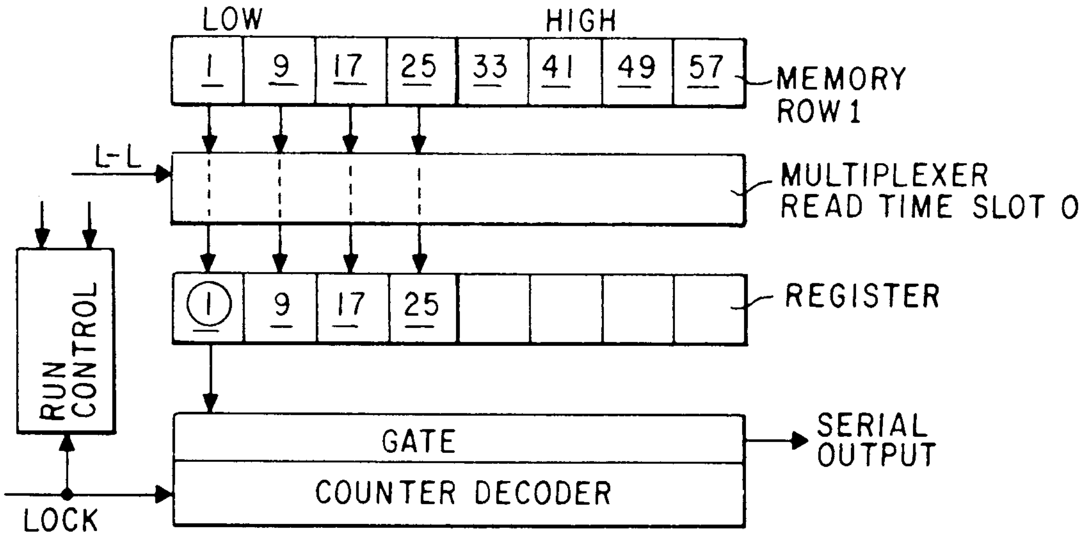

LOW
HIGH
1
9
17
25
33
41
49
57
MEMORY ROW 1
L-L
MULTIPLEXER READ TIME SLOT 0
RUN CONTROL
1
9
17
25
REGISTER
GATE
COUNTER DECODER
SERIAL OUTPUT
LOCK

Fig.8

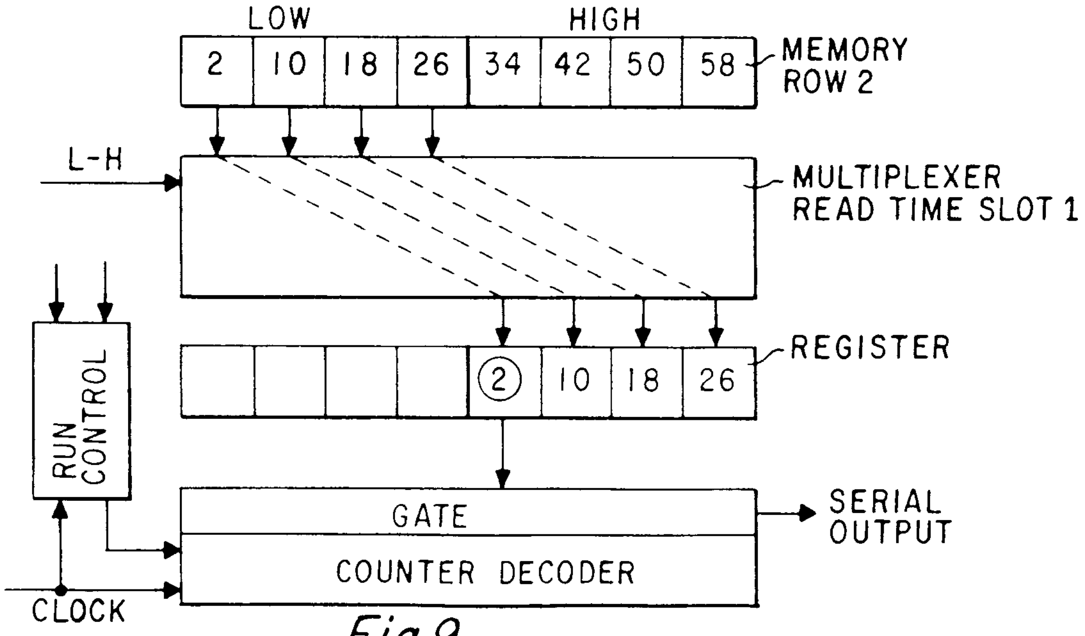

LOW
HIGH
2
10
18
26
34
42
50
58
MEMORY ROW 2
L-H
MULTIPLEXER READ TIME SLOT 1
RUN CONTROL
2
10
18
26
REGISTER
GATE
COUNTER DECODER
SERIAL OUTPUT
CLOCK

Fig.9

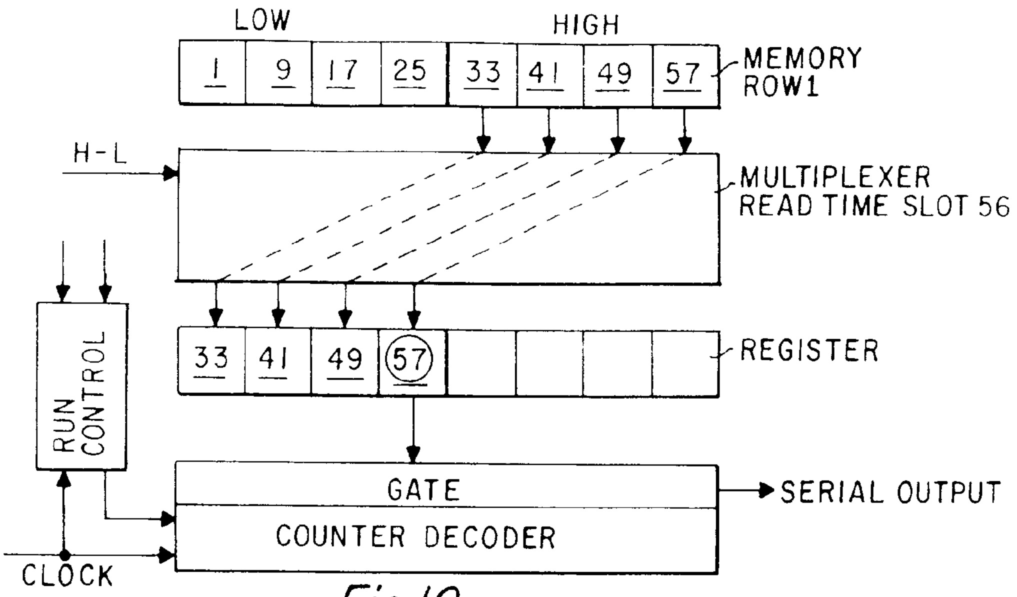


Fig.10

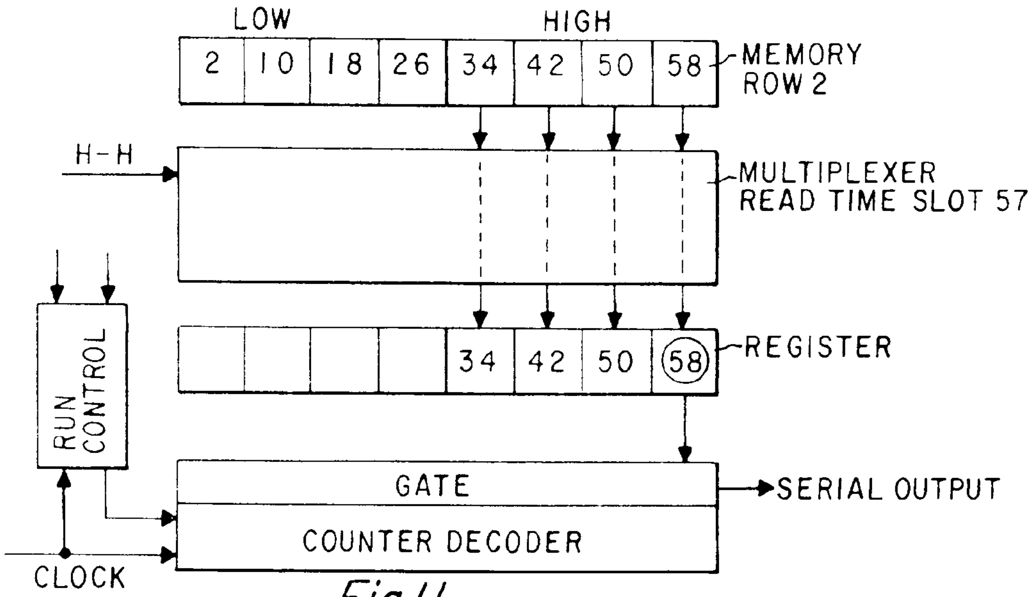


Fig.11

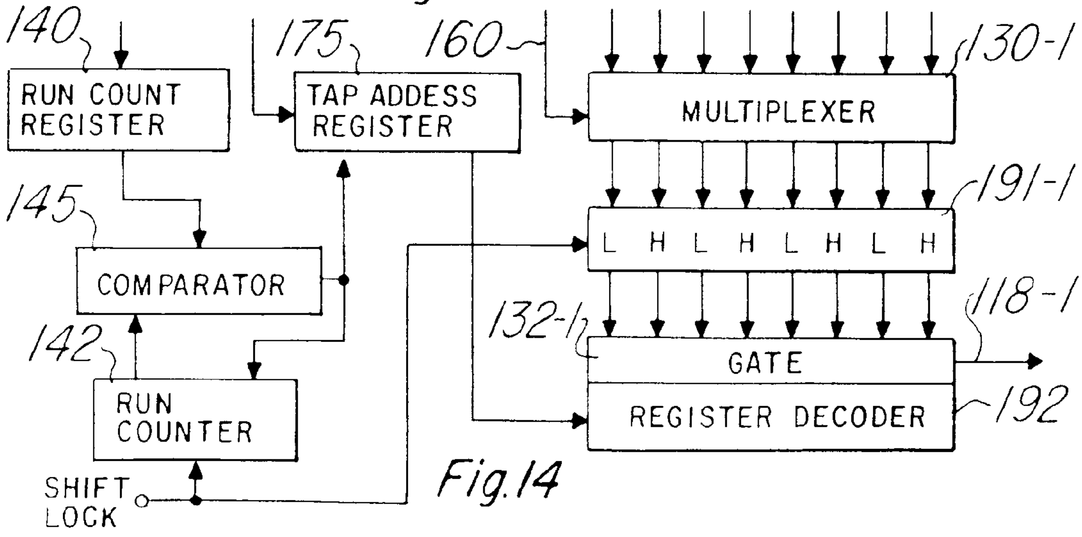


Fig.14