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

(11) Publication number: **0 462 708 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **91304736.1**

(51) Int. Cl.5: **G09G 1/16**, G06F 12/02

(22) Date of filing: **24.05.91**

(30) Priority: **19.06.90 US 540546**

(43) Date of publication of application:
27.12.91 Bulletin 91/52

(84) Designated Contracting States:
DE FR GB

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

(72) Inventor: **West, Roderick Michael Peters**
Unit 10, Carriage Way
Colchester, Vermont 05446(US)
Inventor: **Williams, Todd**
Box RR1, Box 1015, Kingshill Road
Westford, VT 05494(US)

(74) Representative: **Killgren, Neil Arthur**
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

(54) Video random access memory.

(57) A Video Random Access Memory device wherein full and efficient use of a serial access memory portion provides a simple and efficient means of avoiding Mid-Line Reloads and Real-Time Data Transfers whereby selected parts of two different rows in a random access memory portion are transferred simultaneously to the serial access memory portion via addressable transfer gates under the control of address/control logic.

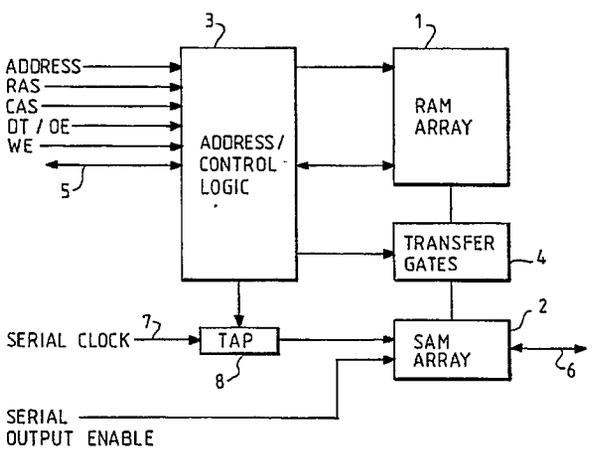

FIG. 1 PRIOR ART

EP 0 462 708 A2

The present invention relates to memory devices capable of high speed data transfer to a peripheral device, such as a raster display.

Video Random Access Memories (VRAMs) are a type of memory commonly used in video displays for computer systems. A VRAM is essentially a conventional dynamic random access memory (DRAM) with the addition of a second port where data may be accessed serially. A VRAM consists of a random access memory (RAM) portion and a serial access memory (SAM) portion with transfer gates which allow data to pass between the RAM and the SAM. The SAM array usually has the memory capacity of one row of the RAM array. A full row of memory data may be passed between RAM and SAM in a single data transfer access. The RAM port and the SAM port may be operated asynchronously and independently except when the data transfer between the RAM and the SAM is taking place.

This independent and asynchronous operation of the two ports finds application in the video displays of computer systems where the RAM port is used to update the contents of display memory and the serial port is used to provide data to be rastered onto the display. The RAM port may be operated at the frequency of the computer system and the SAM port at a frequency dictated by the requirements of the raster display. Since the SAM array usually has the capacity of a single row of display data, it must be continually reloaded with new rows of display data during the time of the display frame. In general, each new row of display data is obtained from a row whose index is one greater than that of the previous row. The reloading of the SAM array with new rows of display data from the RAM array is achieved by performing data transfer cycles at the RAM port. These data transfer cycles between the RAM array and the SAM array are the only interruption to the normal RAM access cycles at the RAM port. They may be separated into two distinct types. The first is data transfer when the SAM port is inactive, with no data passing to the raster display and with the serial clock stopped. This is usually associated with reloading of the SAM during blanking of the display frame which, typically, may account for 20% of the time. The second is data transfer when the SAM port is active, with data passing to the raster display. Since in this case the serial clock is running, the data transfer cycle at the RAM port requires accurate synchronisation with the serial clock in order to maintain the required seamless flow of data to the raster display from the SAM port. This second situation is often referred to as "Real-Time Data Transfer" in the nomenclature of Video Random Access Memories or as "Mid-Line Reload" in the nomenclature of VRAM-based display subsystems.

In the design of a display memory subsystem, the control and timing of such "Real Time Data Transfers" presents a major problem. A "Real Time Data Transfer" is a critically timed real-time access, requiring synchronisation between the RAM and the SAM ports, and can be very wasteful of RAM port bandwidth, a crucial aspect in many display memory subsystems. Additionally such critically timed real time accesses may require potentially complex and high-speed circuitry to synchronise and control them. The avoidance of such events is therefore highly desirable.

The conventional method of avoiding mid-line reloads, used in many VRAM-based display memory subsystems, involves a number of restrictions upon how the contents of display memory are mapped onto the display screen. These restrictions are usually first, to use a fixed start address for the display data on the first horizontal scan line of the display frame, secondly to use a fixed address increment to generate the start address of each subsequent horizontal scan-line and thirdly to use a horizontal scan-line length which requires an amount of display data not greater than the capacity of the SAM arrays of the VRAMs in the display memory subsystem. All these restrictions must be satisfied to avoid a mid-line reload. For a general purpose graphics adapter or display memory subsystem these restrictions cannot be applied.

Second generation VRAMs were enhanced with the ability to transfer half a row of random access memory into half of the SAM while the other half of the SAM is being scanned out to the display. This means of avoiding real-time data transfers is found on some modern 1Mb VRAMS, for example a 1Mbit multiport DRAM manufactured by the Toshiba Corporation and the memories described in the US patents 4,825,411 and 4,855,959 incorporate the so-called "Split Register" feature. These VRAMS have the SAM array divided into two halves, which can be loaded independently by so-called "Split Register Data Transfers" whereby one half of the SAM is loaded while the other half is active. Typically an output status pin is provided to indicate the half of the SAM being scanned out.

This feature goes some way to alleviating the problem, but, however, it does not make full and efficient use of the SAM array capacity and can potentially result in twice as many data transfer accesses than would be required by a VRAM without the feature.

Accordingly, it is an object of the present invention to make full and efficient use of the SAM and to thereby provide a simple and efficient means by which, under certain circumstances, these "mid-line reloads" can be avoided.

Where system constraints prevent the total avoidance of "Mid-Line Reloads", or where, for whatever

reason, it is advantageous to use "Mid-Line Reloads", the invention provides a means of eliminating their real-time nature and thus the need for such critical timing. By removing the need for real-time VRAM data transfers, the invention eliminates the need for the potentially complex and high-speed circuitry required to synchronise and control such data transfers, and eliminates the potentially wasteful use of RAM port bandwidth in the synchronisation of such data transfers.

Therefore, in accordance with the present invention, there is provided a memory device comprising: a random access memory portion arranged with the memory locations in rows and columns; a serial access memory portion; a serial access means allowing external access to the serial access memory portion; transfer gates connecting the random access memory portion and the serial access memory portion for carrying out data transfer therebetween; control logic for controlling the data transfer between the random access memory portion and the serial access memory portion; characterised in that the control logic is adapted to activate transfer gates corresponding to at a least a first selected set of columns for data transfer between memory locations in the selected set of columns of at least one row of the random access memory portion and the serial access memory portion.

The control logic can be adapted to activate transfer gates corresponding to a first selected set of columns for data transfer between memory locations in the first selected set of columns of a first row of the random access memory portion and the serial access memory portion, and transfer gates corresponding to a second selected set of columns for data transfer between memory locations in a second selected set of columns of a second row of the random access memory portion and the serial access memory portion, the data transfers being simultaneous.

According to a preferred feature of the invention the random access memory portion is divided into at least two segments in such a way that logically adjacent rows are located in different segments and further characterised in that said first row is located in a first segment and that said second row is logically adjacent the first row and located in a different segment. This provides an advantageous way of accessing two rows simultaneously. It should be noted that there may be other ways of accessing two rows in the RAM simultaneously which also would be in accordance with the invention.

Advantageously, the selected sets of columns can comprise integer multiples of n columns, where n is 2, 4, 8 or higher powers of 2. This reduces the demand on the decoding of tie column address in the transfer gate selection. The invention has beneficial application even when the granularity is extremely coarse. If only the most significant 3 bits of the column address are decoded the transfer gates are divided into 8 separate blocks along the row length. In the most extreme form, only the most significant bit of the column address is used to select transfer gates divided into 2 separate blocks.

According to another preferred feature of the invention said first selected set of columns are the columns located between one end of the row and a selected column, including the selected column, and said second selected set of columns are the columns located between said selected column and the other end of the another row, not including the selected column, the selected column being selectable by the control logic.

The memory device can include a pointer capable of being loaded with an initial address indicating the location in the serial access memory portion that is currently available at the serial access means the pointer being updated, simultaneously with the data transfer between the random access memory portion and the serial access memory portion, with the value of said selected column. This aspect of the invention is advantageous when the data transfer occurs with the serial clock inactive.

Alternatively the memory device can include a pointer capable of being loaded with an initial address indicating the location in the serial access memory portion that is currently available at the serial access means, the pointer not being updated simultaneously with the data transfer between the random access memory portion and the serial access memory portion. This aspect of the invention is advantageous when the data transfer occurs with the serial clock active.

In another form of the invention, the memory device can also include a pointer capable of being loaded with an initial address indicating the cell in the serial access memory portion that is currently available at the serial access means and the pointer being updated, simultaneously with the data transfer between the random access memory portion and the serial access memory portion, with a value different from that of the selected column.

A memory device according to the invention finds particular, though not exclusive, application as a display memory for storing data representative of an image to be displayed in a display system, which may be incorporated in a data processing system. Accordingly, therefore the invention also relates to a display system including such a memory device and a display device.

In order to illustrate more clearly the present invention we shall first describe the structure and operation of a conventional VRAM and in the light of this we shall describe an embodiment of the present

invention with reference to the Table to be found at the end of the description and the drawings, wherein:

Figure 1 is a block diagram of a conventional VRAM;

Figure 2 is a timing diagram for a conventional Read Data Transfer Cycle with the Serial Clock inactive;

Figure 3 is a timing diagram for a conventional Read Data Transfer Cycle with the Serial Clock active. This is the so-called "Real-Time Data Transfer";

Figure 4 is a map of the serial access memory after a column wrapped read data transfer;

Figure 5 is a block diagram of a video random access memory in which the RAM portion is segmented into two segments;

Figure 6 is a timing diagram for a first form of column wrapped read data transfer with the serial clock inactive;

Figure 7 is a timing diagram for a second form of column wrapped read data transfer with the serial clock active;

Figure 8 is a map of the serial access memory before and after a type 2 column wrapped data transfer with the serial clock active;

Figure 9 is a block diagram of a display system employing a memory according to the invention.

The structure of a conventional VRAM is shown by Figure 1. It comprises a RAM array 1, a SAM array 2, address/control logic 3, and transfer gates 4. The RAM array is connected to the primary (RAM) port 5 of the VRAM and behaves in a manner identical to that of a DRAM, under the control of the address/control logic. The SAM array is connected to the secondary (SAM or Serial) port 6 of the VRAM and may be accessed serially under the control of an external asynchronous clock 7, the Serial Clock. The serial access to the SAM is controlled by the Tap Pointer (TAP) 8, which generates an address into the SAM from a counter which increments on each cycle of the Serial Clock. The Tap Pointer (TAP) is capable of being loaded with an initial address, under the control of the address/control logic. The address/control logic 3 supervises the address multiplexing and the data flow on the RAM port 5 and provides all the control and global timing functions of the VRAM. The transfer gates 4 allow memory data to pass between the RAM array 1 and the SAM array 2, under the control of the address/control logic 3.

The read data transfer cycles used in conventional VRAMs are shown by Figures 2 and 3. A Read Data Transfer cycle is indicated by DT/OE set to a low level at the falling edge of the Row Address Strobe (RAS). At the falling edge of the RAS the row address (R) is obtained from the address input and row R is activated. At the falling edge of the Column Address Strobe (CAS), the column address (C) is obtained from the address input. Subsequently, the actual RAM to SAM data transfer occurs at the rising edge of DT/OE. At the data transfer, the SAM is loaded with the contents of RAM array row R and the Tap Pointer (TAP) is loaded with the column address C. On the first rising edge of the Serial Clock after the actual data transfer, the new contents of the SAM are available at the SAM port, starting at the SAM location given by the Tap Pointer value at the time of the first Serial Clock rising edge. The first item of serial data is {R;C}. {R;C} refers to the data item at row R and column C. {R;C:C+4} refers to 5 data items at row R and columns C through C+4. This notation will be used throughout the description. Each subsequent rising edge of the Serial Clock, causes the Tap Pointer to increment and present the contents of the SAM serially at the SAM port: {R;C} is followed by {R;C+1}, then by {R;C+2} and so on. If the read data transfer is performed with the serial clock inactive as is shown by Figure 2, then the timing of the transfer is uncritical since data is not being passed to the display. However, if the Read Data Transfer is performed with the Serial Clock running as is shown by Figure 3, then the data transfer, signalled by the rising edge of DT/OE, must be correctly timed to occur during the correct Serial Clock cycle so as to maintain the correct sequence of data at the SAM port.

If the Tap Pointer reaches the last location in the SAM, then on the next rising edge of the Serial Clock its value wraps back to zero to address the start of the SAM and will continue to increment from zero for each subsequent Serial Clock cycle. This is generally undesirable since the sequence of data presented at the SAM port is then discontinuous, jumping from the end of the row back to the start of the same row.

In a memory according to an embodiment of the present invention at the falling edge of RAS, the row address (R) is obtained from the address input and two rows (R and R+1) are activated. At the falling edge of CAS, the column address (C) is obtained from the address input. At the data transfer, data is transferred between the two RAM array rows (R and R+1) and the SAM. Data is transferred between RAM array row R, column locations C to the row end, and SAM locations C to the end of the SAM. Additionally, data is transferred between RAM array row R+1, column locations 0 to C-1, and SAM locations 0 to C-1.

This may be expressed as:

SAM(C:END) = {R;C:END}
SAM(0:C-1) = {R+1;0:C-1}

which may be combined in a single expression as:

SAM(0:END) = {R+1;0:C-1}||{R;C:END}.

In these expressions the parameter END is used for the last column address of a row and the last address of the SAM. The dyadic operator || indicates concatenation. This form of data transfer we shall designate Column Wrapped Data Transfer (CWDT). The column address (C) forms the boundary of the CWDT.

After a CWDT Read Data Transfer, the SAM contains a full row of continuous data from address {R;C} to {R+1;C-1}, starting at SAM(C). The data is continuous in RAM address space starting at the CWDT boundary with {R;C} at SAM(C) and wrapping around the end of the SAM through to {R+1;C-1} at SAM(C-1). This is shown diagrammatically as the map of the SAM and its contents shown by Figure 4.

The CWDT function may be used as an alternative to, or in addition to the conventional data transfer accesses available in current VRAMs. For a VRAM providing CWDT and conventional data transfers, these may be distinguished through the use of a function pin or by another suitable means. In the embodiment described here it is assumed that the CWDT function is used in place of conventional VRAM data transfers.

It is advantageous that the RAM array be segmented into at least two segments such that at least one row address bit (including the least significant bit) is used to select a segment and the remainder of the row address bits are used to select a row within each segment. Such memory segmentation is employed in large memories in order to reduce the loading on individual rows and columns. This decreases signal generation and propagation delays while reducing both variation in data rates and power consumption. For a memory in accordance with the invention, the segmentation of the memory also enables the simplification of the simultaneous activation of a plurality of rows by placing logically sequential rows in physically separate segments.

Figure 5 is a block diagram of a VRAM with the RAM array segmented into two physically separate segments. One segment contains all even rows and the other contains all odd rows. Each segment has a separate set of transfer gates 9,10 to allow memory data to pass between the RAM array segments 11,12 and the SAM array 13, under the control of the address/control logic 14. The RAM Port 15 operation of the VRAM is unchanged, and its SAM port operation, 16 is only changed by the use of the CWDT function.

With reference to Figure 5, the CWDT data transfer is achieved by the address/control logic 14 activating two rows (R and R+1 in separate segments) and selecting which transfer gates to open, allowing selective data transfer between the two rows and the SAM. For a CWDT data transfer with row address R and column address C, the address/control logic selects transfer gates (C:END) for the segment containing row R and transfer gates (0:C-1) for the segment containing row R+1. In this way the CWDT boundary is quantised at single column granularity and this requires that the column address (C) he fully decoded for the selection of transfer gates. In many cases however, it would be sufficient to quantise the CWDT boundary at higher granularity (e.g. at 2,4,8,16,32... column boundaries). This reduces the demand on the decoding of the column address in the formation of the CWDT boundary and transfer gate selection. The invention has beneficial application even when the CWDT boundary granularity is extremely coarse. If only the most significant 3 bits of C are decoded the transfer gates are divided into 8 separate blocks along the row length. In the most extreme form, only the most significant bit of C is used to select transfer gates divided into 2 separate blocks.

CWDT may be used as an alternative to, or in addition to the conventional data transfer cycles available in current VRAMs. Although CWDT will be discussed in relation to Read Data Transfers (RAM to SAM), as used in a display memory subsystem, it also finds application in relation to Write Data Transfers (SAM to RAM) found in some current VRAMs. The application of CWDT to Write Data Transfers will not be discussed but is within the scope of the present invention, as will be apparent to those skilled in the art.

The present invention provides for two forms of CWDT. The two forms differ only in whether or not the Tap Pointer 17 is updated. The first form of CWDT, designated CWDT#1, is similar to a conventional Read Data Transfer in that, at the data transfer, the Tap Pointer is loaded with the column address (C) obtained at the falling edge of CAS. The second form of CWDT, designated CWDT#2, differs from a conventional Read Data Transfer in that, at the data transfer, the Tap Pointer remains unchanged. Both forms of CWDT may be used with the Serial Clock either inactive or running. It is considered that CWDT#1 is more likely to be used with the Serial Clock inactive and CWDT#2 is more likely to be used with the Serial Clock running. CWDT#1 updates both the contents of the SAM and the Tap Pointer, therefore, if it is used with the Serial Clock running, the data transfer must be accurately timed with respect to the Serial Clock cycles. CWDT#2

updates only the contents of the SAM. When CWDT#2 is used with the Serial Clock running, provided the data transferred is the same as and overlaps the previous SAM data in the region of the Tap Pointer, the data transfer need not be accurately timed with respect to the Serial Clock cycles.

Figures 6 and 7 are timing diagrams illustrating the two forms of CWDT data transfer. Figure 6 illustrates the first type of CWDT, CWDT#1, by a Read Data Transfer with the Serial Clock inactive. Figure 7 illustrates the second type of CWDT, CWDT#2, by a Read Data Transfer with the Serial Clock active. In this embodiment of the invention the two forms of CWDT data transfer are distinguished by the level of CAS at the rising edge of DT/OE. If CAS is at a low active level at the rising edge of DT/OE then the Tap Pointer is updated. This is the CWDT#1 read data transfer as shown by Figure 6. If CAS is at a high inactive level at the rising edge of DT/OE then the Tap Pointer is not updated. This is the CWDT#2 read data transfer as shown by Figure 7.

As in conventional VRAMs, a Read Data Transfer cycle is indicated by DT/OE set to a low level at the falling edge of RAS. At the falling edge of RAS, the row address (R) is obtained from the address input and two rows (R and R+1 in separate segments) are activated. At the falling edge of CAS, the column address (C) is obtained from the address input. The column address (C) forms the boundary of the CWDT. Subsequently, the actual RAM to SAM data transfer occurs at the rising edge of DT/OE. The level of the CAS at the rising edge of DT/OE determines whether the Tap Pointer (TAP) is to be loaded with the column address C, hence whether the CWDT is a CWDT#1 or a CWDT#2. This is one particular means of control of the CWDT function. Other means of control can be devised, with the relative timings, polarities and operative functions of control inputs varied. The actual operation of CWDT accesses will depend on a number of factors, including whether the CWDT feature is offered as an alternative to or in addition to conventional data transfer accesses.

At the data transfer, the SAM is loaded with {R+1;0:C-1}¦¦{R;C:END}, the contents of RAM array rows R and R+1 divided at the CWDT boundary (C), and the Tap Pointer (TAP) is loaded with the column address C if the CWDT access is a CWDT#1. On the first rising edge of the Serial Clock after the actual data transfer, the new contents of the SAM are available at the SAM port, starting at the SAM location given by the Tap Pointer value at the time of the first Serial Clock rising edge.

For a CWDT#1 Read Data Transfer (as in Figure 6), where the Tap Pointer has been updated, on the first rising edge of the Serial Clock after the actual data transfer, the new contents of the SAM are available at the SAM port, starting with {R;C}. Each subsequent rising edge of the Serial Clock, causes the Tap Pointer to increment and present the contents of the SAM serially at the SAM port: {R;C} is followed by {R;C+1}, {R;C+2} and so on. When the Tap Pointer reaches the last location in the SAM, then on the next rising edge of the Serial Clock its value wraps back to zero to address the start of the SAM and will continue to increment from zero for each subsequent Serial Clock cycle. The serial data sequence around the time of the Tap Pointer wrapping back is {R;END-1},{R;END},{R+1;0},{R+1;1},{R+1;2} and so on. Thus the serial data sequence moves across the row boundary in a seamless and continuous manner in RAM address space.

For a CWDT#2 Read Data Transfer (as in Figure 7), where the Tap Pointer has not been updated and the Serial Clock is active, in order to keep the serial data sequence seamless and avoid any critical timing of the data transfer, the data transferred to the SAM must be the same as and overlap the previous SAM data in the region of the Tap pointer at the time of the actual data transfer. To illustrate this, in Figure 7, the data in the SAM prior to the data transfer is {R;0:C+8}¦¦{R-1;C+9:END}. This data was loaded into the SAM by a previous CWDT, with a row address of R-1 and a column address of C+9. At the data transfer, the SAM is loaded with {R+1;0:C-1}¦¦{R;C:END}. The data in SAM locations SAM(C:C+8) is unchanged by the data transfer and remains as {R;C:C+8}. This region of unchanged data is termed the "Overlap Region".

This is be shown diagrammaticaliy by the maps of the SAM and its contents shown by Figure 8 and by Table I.

In the timing diagram (Figure 7), the actual data transfer is shown to occur when the Tap Pointer has a value of C+4. On the first rising edge of the Serial Clock after the actual data transfer, the new contents of the SAM are available at the SAM port, starting with {R;C+4}. The CWDT#2 data transfer does not alter or affect the incrementing sequence of the Tap Pointer. The data transfer need not be critically timed within the Serial Clock stream, provided that at the moment of data transfer the Tap pointer is anywhere in the "Overlap Region". The choice of the size of the Overlap Region must be based on system constraints to ensure seamless serial data. Thus, in the example of Figure 7, the serial data sequence can proceed in a seamless manner and continuously from {R-1;C+9} through to {R+1;C-1}, nearly two full rows linked by a single CWDT#2 access; a sequence which can be extended by further CWDT#2 accesses. This is achieved without any real-time data transfers.

Whereas a Mid-Line Reload using a conventional real-time read data transfer has a "Transfer Window" confined to a single Serial Clock cycle, a CWDT#2 read data transfer has a Transfer Window as wide as the Overlap Region. The size of the Overlap Region may be chosen, based on system constraints, to avoid any critical timing of the data transfer and is definable by the system designer.

As an extension to the CWDT data transfer accesses described above, it would be possible to apply different values to the CWDT boundary and the update of the Tap Pointer. At the falling edge of CAS, the CWDT boundary is obtained from the address input. Provided that CAS is at an active low level (i.e. CWDT#1), the value used to update the Tap Pointer is obtained from the address input at the rising edge of DT/OE. In this manner, the CWDT boundary and the Tap Pointer can be set at different values. This has a certain synergy with conventional data transfers, in that they always have a CWDT boundary of 0 and the Tap Pointer can be updated with a value obtained from the address input.

Figure 9 is a block diagram of a display system employing a memory according to the invention. It shows a workstation consisting of a Central Processing Unit (CPU) 20, a Read Only Store (ROS) 22, a Random Access Memory 24, a disk drive for data storage 26, a user interface 28 which may be a keyboard and/or a mouse, a display device 30 connected via a display adapter 32. These units are connected together by a system bus 34. The display adapter 32 contains a display memory which employs a VRAM, according to the invention, wherein the RAM portion is updated via the RAM port, and the serial access port is used to provide data to be rastered onto the display, 30. It should be noted that this is only one possible embodiment of a display system according to the invention. Many other types are possible, including mainframe data processing systems with a number of users wherein there is a display device and display adapter for each user.

The invention simply and efficiently achieves full utilisation of the SAM portion in a VRAM. Every CWDT read data transfer loads the SAM with data that is continuous in RAM address space starting at the CWDT boundary and of a length equal to the full capacity of the SAM. By starting at the CWDT boundary, the serial data sequence can move in a seamless manner across a row address boundary, providing sequential data up to the full capacity of the SAM before a further data transfer is required. A conventional read data transfer does not permit the serial data sequence to move across row address boundaries without a real-time data transfer. A conventional read data transfer can only utilise the full capacity of the SAM, in a manner appropriate to a display memory subsystem, when the column address is 0.

The invention may, in certain circumstances, eliminate the need for "Mid-Line Reloads" in a display memory subsystem, by utilising the full capacity of the SAM. Additionally, in certain circumstances, the invention may reduce the number of VRAM data transfers required for each display frame. Where system constraints prevent the total avoidance of "Mid-Line Reloads", or where it is advantageous to use "Mid-Line Reloads", the CWDT#2 data transfer provides a means of eliminating the real-time nature of the "Mid-Line Reload". By removing the need for real-time VRAM data transfers, CWDT eliminates the need for the potentially complex and high speed circuitry required to synchronise and control such data transfers, and eliminates the potentially wasteful use of RAM port bandwidth in the synchronisation of such data transfers. It is an advantage of the invention that CWDT may be used as an alternative to, or in addition to the conventional data transfer accesses available in current VRAMs. Although CWDT has been discussed in relation to Read Data Transfers (RAM to SAM), as used in a display memory subsystem, it also finds application in relation to the Write Data Transfers (SAM to RAM) found in some current VRAMs. The application of CWDT to Write Data Transfers is within the scope of the invention, as will be apparent to those skilled in the art.

TABLE I

SAM locations	Data contents before transfer	Data contents after transfer
(0:C-1)	{R;0:C-1}	{R+1;0:C-1}
(C:C+8)	{R;C:C+8}	{R;C:C+8}
(C+9:END)	{R-1;C+9:END}	{R;C+9:END}.

Claims

1. A memory device comprising:

a random access memory portion arranged with the memory locations in rows and columns;

a serial access memory portion;

a serial access means allowing external access to the serial access memory portion;

transfer gates connecting the random access memory portion and the serial access memory portion for carrying out data transfer therebetween;

control logic for controlling the data transfer between the random access memory portion and the serial access memory portion;

characterised in that the control logic is adapted to activate transfer gates corresponding to at least a first selected set of columns for data transfer between memory locations in the selected set of columns of at least a first row of the random access memory portion and the serial access memory portion.

2. A memory device as claimed in claim 1 wherein the control logic is adapted to activate transfer gates corresponding to the first selected set of columns for data transfer between memory locations in the first selected set of columns of the first row of the random access memory portion and the serial access memory portion, and transfer gates corresponding to a second selected set of columns for data transfer between memory locations in a second selected set of columns of a second row of the random access memory portion and the serial access memory portion, the data transfers being simultaneous.

3. A memory device as claimed in claim 2, wherein the random access memory portion is divided into at least two segments in such a way that logically adjacent rows are located in different segments and further characterised in that said first row is located in a first segment and that said second row is logically adjacent the first row and located in a different segment.

4. A memory device as claimed in claim 2 or claim 3 wherein said first selected set of columns are the columns located between one end of the row and a selected column, including the selected column, and said second selected set of columns are the columns located between said selected column and the other end of the another row, not including the selected column, the selected column being selectable by the control logic.

5. A memory device as claimed in claim 4 which further comprises a pointer capable of being loaded with all initial address indicating the location in the serial access memory portion that is currently available at the serial access means and wherein the pointer is updated, simultaneously with the data transfer between the random access memory portion and the serial access memory portion, with the value of said selected column.

6. A memory device as claimed in any of claims 1 to 4 which further comprises a pointer capable of being loaded with an initial address indicating the location in the serial access memory portion that is currently available at the serial access means and wherein the pointer is not updated simultaneously with the data transfer between the random access memory portion and the serial access memory portion.

7. A memory device as claimed in claim 4 which further comprises a pointer capable of being loaded with an initial address indicating the location in the serial access memory portion that is currently available at the serial access means and wherein the pointer is updated, simultaneously with the data transfer between the random access memory portion and the serial access memory portion, with a value different from that of the selected column.

8. A memory device as claimed in any preceding claim wherein the selected sets of columns comprise integer multiples of n columns, where n is 2, 4, 8 or higher powers of 2.

9. A display memory for storing data representative of an image to be displayed on a display device, the display memory including a memory device as claimed in any preceding claim.

10. A display system comprising a display device and a display memory as claimed in claim 9 for providing data to the display device.

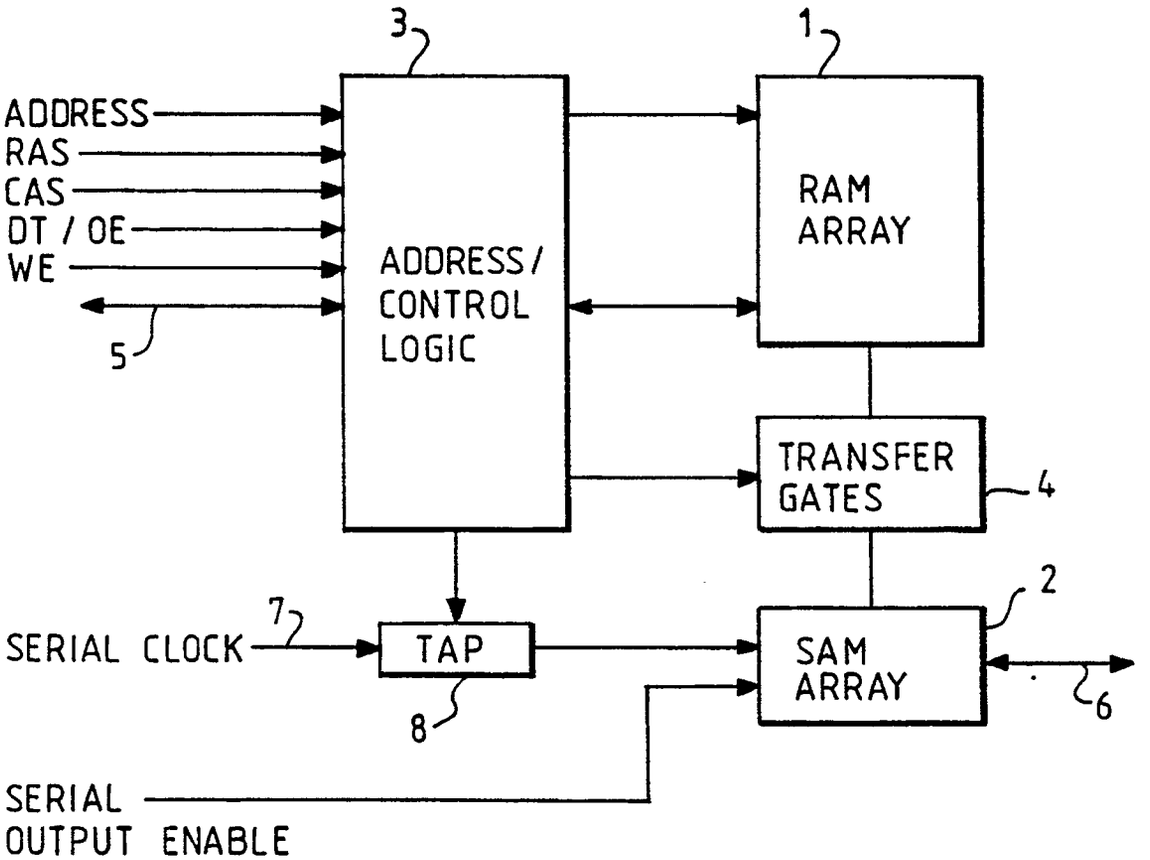

FIG. 1 PRIOR ART

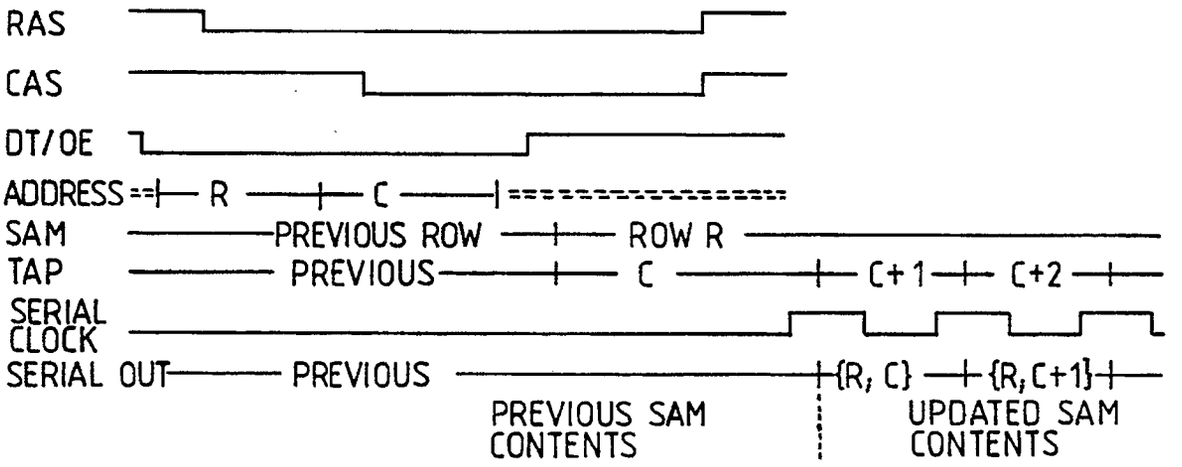

FIG. 2 PRIOR ART

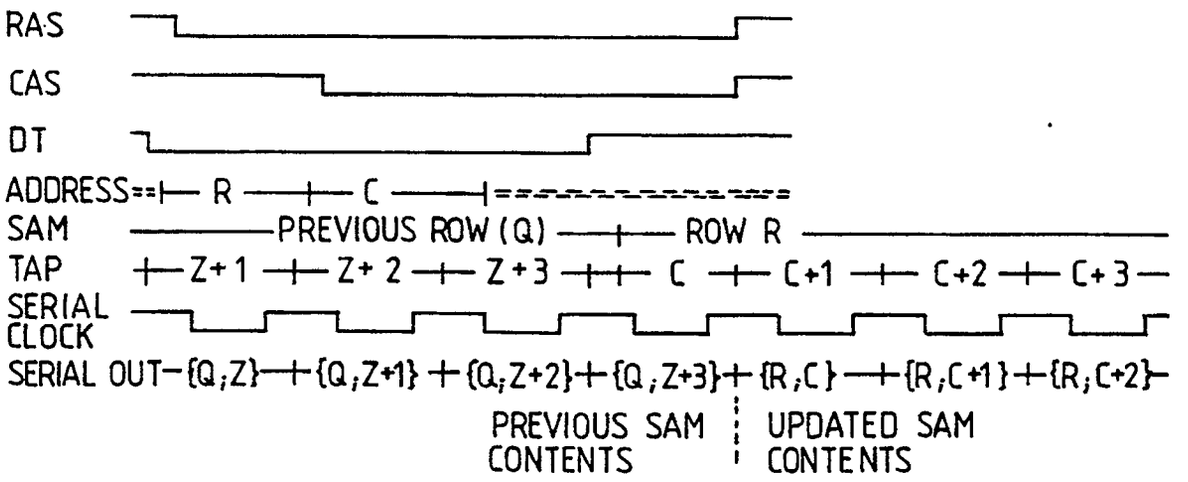

FIG. 3 PRIOR ART

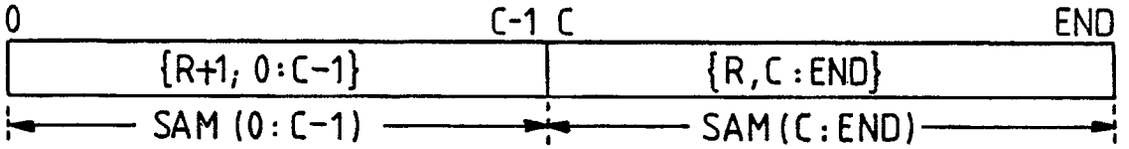
0
C-1 C
END
{R+1; 0:C-1}
{R,C:END}
SAM (0:C-1)
SAM (C:END)

FIG. 4

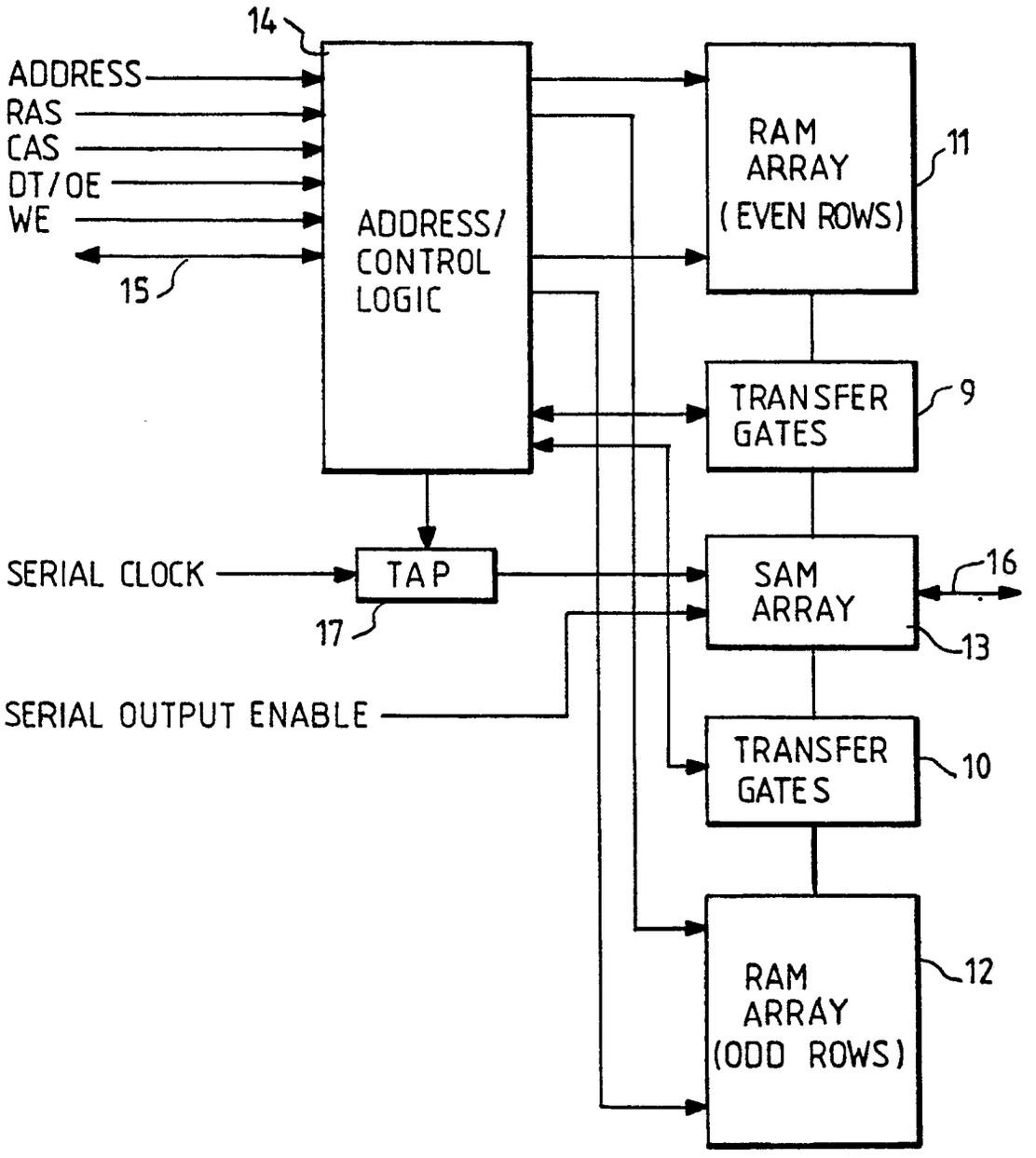
14
ADDRESS
RAS
CAS
DT/OE
WE
15
ADDRESS/
CONTROL
LOGIC
RAM
ARRAY
(EVEN ROWS)
11
TRANSFER
GATES
9
SERIAL CLOCK
TAP
17
SAM
ARRAY
16
13
SERIAL OUTPUT ENABLE
TRANSFER
GATES
10
RAM
ARRAY
(ODD ROWS)
12

FIG. 5

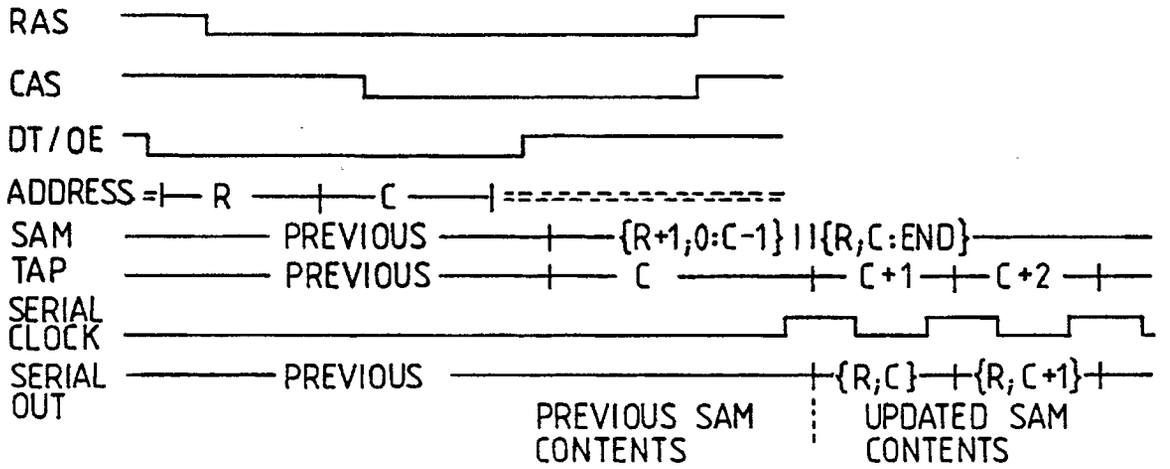

FIG. 6

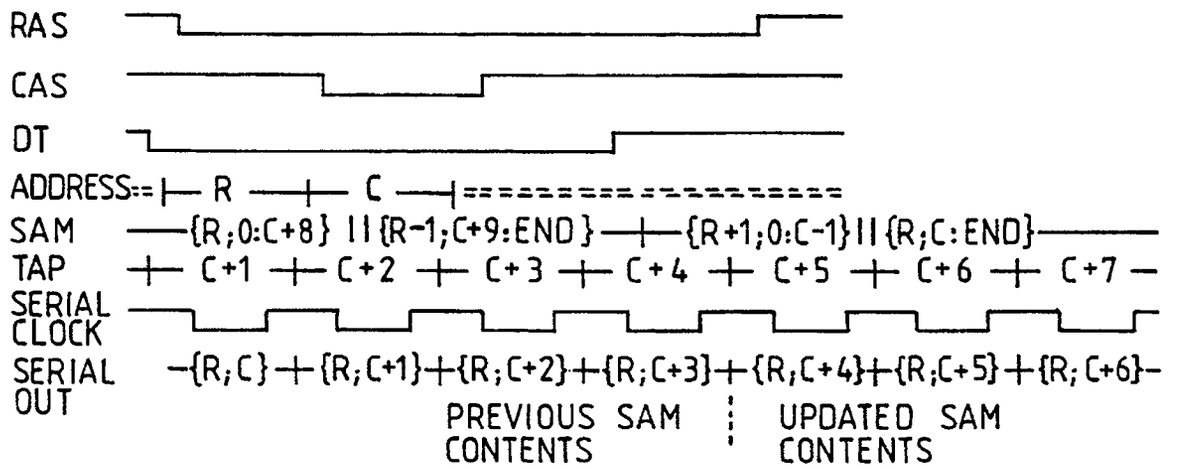

FIG. 7

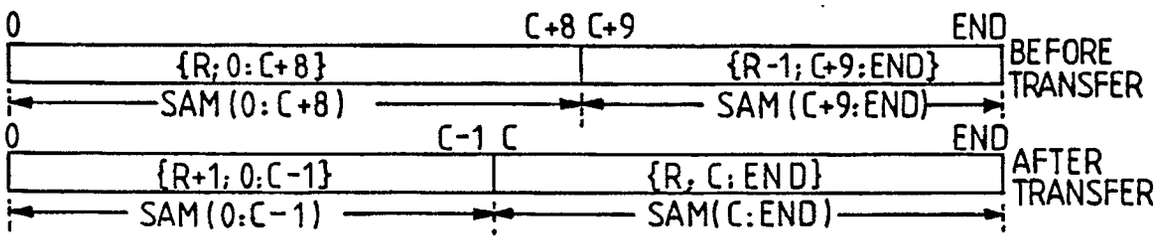

FIG. 8

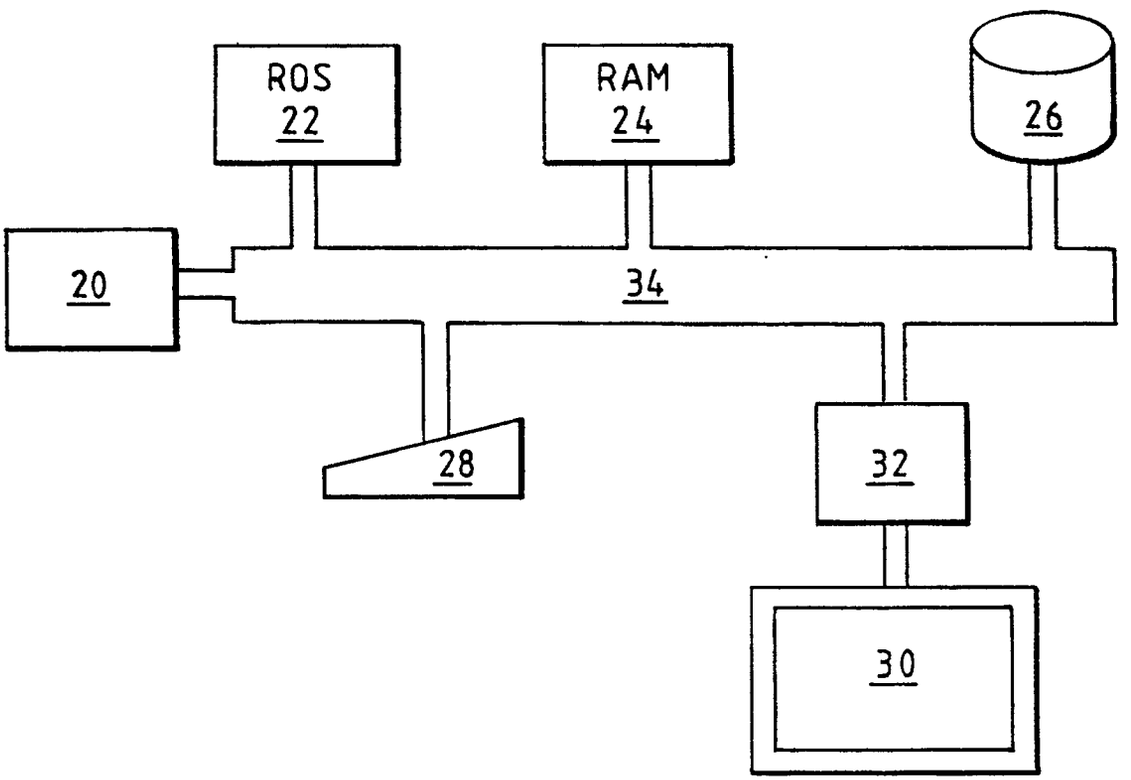

FIG. 9