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

(11) EP 0 410 777 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
06.11.1996 Bulletin 1996/45

(51) Int Cl.[6]: **G09G 1/16**

(21) Application number: **90308245.1**

(22) Date of filing: **27.07.1990**

(54) **Video graphics display memory swizzle logic circuit and method**

Logische Schaltung und Verfahren zum Wiederordnen für einen graphischen Videoanzeigespeicher

Circuit logique et méthode de réarrangement pour une mémoire d'affichage de graphiques vidéos

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

(30) Priority: **28.07.1989 US 387567**
28.07.1989 US 387568

(43) Date of publication of application:
30.01.1991 Bulletin 1991/05

(73) Proprietor: **TEXAS INSTRUMENTS INCORPORATED**
Dallas Texas 75265 (US)

(72) Inventors:
- **Guttag, Karl M.**
Missouri City, Texas 77459 (US)
- **Sherlock, Ian J.**
Bedford, MK40 2SL (GB)
- **Simpson, Richard D.**
Carlton, Bedford,MK43 7 LS (GB)

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

(56) References cited:
EP-A- 0 071 744
EP-A- 0 279 230
WO-A-88/07235
US-A- 4 807 189

- **PATENT ABSTRACTS OF JAPAN vol. 12, no. 350 (P-760) 20 September 1988; & JP-A-63 104 186**

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

EP 0 410 777 B1

Printed by Jouve, 75001 PARIS (FR)

Description

TECHNICAL FIELD OF THE INVENTION

This invention relates to block-write graphic control data memory write systems and more particularly to an arrangement which allows for the economical reordering of data prior to controlling the block-write function.

European Patent Application 0 071 744 discloses a method, carried out by a microprocessor under the control of software, of writing a character to a colour graphics frame buffer. A bit map of each possible character is held in a memory with each bit representing the presence or absence of a pixel at a point in a grid. A row of the bit map is read and a word is formed in which each bit of the row occurs twice consecutively. Another word is formed in which the two bit code for the desired colour for the character is repeated. The two words are then ANDed together and the result is written into the frame buffer at the location for that row. The process is repeated for each row of the character.

The International patent application WO88/07235 discloses a bit map graphics system in which the pixels making up a frame are stored in a memory in an order which is suitable for accessing sets of pixels, which, in the displayed frame, make up a block several pixels high and several wide, which is useful for drawing vectors and characters on the display. The special order of the pixels in the memory is achieved by permuting the bits of the (X,Y) addresses of the pixels, where X and Y are the coordinates of the pixels in the displayed frame, and applying the permuted addresses to the memory. For some operations on the pixel data is read out of the memory and permuted into an order suitable for that operation before the operation is performed and is then permuted again and is stored back in the memory.

United States Patent No. 4 807 189 discloses a bit map graphics system having a multiplexer connected to select, for the value to be stored in the frame buffer for a selected pixel, between the value on the data bus and the value stored in a colour register. Storing a value in the colour register allows that value to be conveniently written to a number of locations in the frame buffer addressed in turn.

BACKGROUND OF THE INVENTION

Microprocessors intended for graphics applications must be able to move pixel information between memory bit maps as quickly as possible. In situations where many pixels must be transferred to a bit map, the transfer may be speeded up by using a block-write feature. Typically, a block-write is created by associating a color register with each VRAM, filling the color register with bits to determine the desired color value of selected portions of the VRAM, and then using both the address bits of the VRAM as well as the data bus input to the VRAM to determine the locations within the VRAM where the color represented by the value in the color register will appear. This technique does not burden the data bus with multiple copies of the same pixel value and thus increases the available memory bandwidth, again speeding up data transfers.

The simplest application where the block-write can be used to advantage is the fill, which transfers the same pixel value into a defined area of memory. Also, some forms of data expansion are well suited to the application of block-write techniques. Thus, when a bit map is stored in compressed form the 1's and 0's can represent the presence or absence of a pixel and block-writes can be used to decompress the bit map. Typically, this sort of expansion is applied to character fonts which are often stored in compressed form to save memory.

Problems arise because memory accesses must be made in regular mode and in block-write mode via the same bus and they must be consistent such that data written (or read) in one mode must be able to be read (or written) in the other mode. This is a problem, since before data can be written to VRAMs in block-write mode, the bit order of the compressed representation of the data must be manipulated or swizzled relative to the regular mode access. This bit order change is necessary because typically the compressed data is stored with one bit representing each multibit display pixel in a specific order. The storage of these bits is serial with each bit representing a corresponding display point. For example, the first bit (bit 0) would represent pixel position one. The second bit (bit 1) would represent pixel position two and the third bit (bit 2) would represent pixel position three. Thus, the bits on the bus, in this example would represent the pixel positions one for one, such that bus bit position zero would contain data for the first pixel, while bus position three would contain data for the fourth pixel. However, because of the physical arrangement of the VRAMs where successive pixels are stored in different VRAM chips (or Units), the data must be reordered before presentation to the VRAMS. Consider the case where the VRAMS are four bits wide (four planes) with a 32 bit wide data bus. The data bus would have bus positions 0-3 connected to the first VRAM which in turn can control bits
0-3 of the first pixel in a normal write situation. Without swizzling, the compressed data in bus bit position 1 (the second position) which should be destined to control the second pixel will end up being communicated to the second input of the first VRAM, which with a normal access be associated with the ninth pixel and not the required second pixel. Thus, a bit order rearrangement is necessary when functioning in the block-write mode.

A further problem is encountered since the nature of a data swizzle depends on the size of the pixel. Several different swizzles must be made to accommodate a broad range of pixel sizes and VRAM configurations. Thus, it is fair to say that the block-write mode of the video RAMs can only be reasonably used for filling areas in exact multiples of the block size. The nature of

the VRAM's block-write function results in a scrambled writing to the pixels within a block unless some data reordering is accomplished.

Accordingly, a need exists in the art for a swizzle arrangement which allows for the efficient manipulation of data so as to accomplish block-writes in an economical manner.

A further need exists in the art for swizzle logic which can be used for any size pixel or VRAM configuration.

A further need exists in the art to design a system using the block write mode that can correctly and efficiently control the writing down to each pixel within the block as well. Further, there is a need for such a system that can be applied for different numbers of color planes.

SUMMARY OF THE INVENTION

The present invention provides a video graphics circuit comprising:

a first plurality of data input terminals;
a memory for storing data representing pixels of an image, each pixel being represented by a plurality of bits and the memory having a second plurality of data input terminals;
a reordering circuit for connecting the first plurality of data input terminals to the second plurality of data input terminals in different orders depending on whether the video graphics circuit is operated either

in a first mode in which data presented at the first plurality of data input terminals are stored as pixel values in said memory or
in a second, block write, mode in which single bits of the data presented at the first plurality of data input terminals determine whether a value held in a register is applied to respective pixels in the memory; and

an expansion circuit connected between the first plurality of data input terminals and the reordering circuit for replicating bits presented at the first data input terminals and applying the replicated bits in parallel to the reordering circuit.

The reordering circuit may be a swizzle circuit having a plurality of inputs and a like plurality of outputs, said swizzle circuit comprising: a like plurality of latches, each controlling one input and one or more outputs; and circuitry for controlling which output to which any input is connected at any instant of time.

The reordering circuit may be a multiplex circuit.

The reordering circuit may include a memory having a look-up table.

The video graphics circuit may include circuitry for shifting bits presented at the first data input terminals before they are applied to the expansion circuit.

The memory may be organised in planes and bits of a pixel are stored on respective planes.

The memory may comprise one or more VRAMS.

The video graphics circuit may include circuitry for determining the number of the VRAMS containing the same pixel information and for controlling said expansion circuitry in accordance with said determination.

The expansion circuitry may be arranged for controlling the expansion when said graphics memory is configured with any number of VRAMS controlling said pixels.

A graphics processing system may comprise the video graphics circuit.

In one example using VRAMS and pixels of four bits, assuming that each VRAM has four data input paths (one for each bit of the pixel), there is a separation, or reordering, of four bit positions between the compressed data and the actual input to the VRAMs. This reordering is performed by a swizzle circuit.

Thus compressed bus bit 0 goes to post swizzle position 0, while bus bit 1 goes to post swizzle position 4. Likewise, compressed bus bit 2 goes to post swizzle position 8 and compressed bus bit 3 goes to post swizzle position 12. This continues for 7 compressed bit positions with compressed bit 7 going to post swizzle position 28. The next compressed bit, bit 8, goes to post swizzle position 1, while compressed bit 9 goes to post-swizzle position 5. This discontinuous sequence continues for the full bus width.

In an example where the pixel size is 8 bits, two four bit wide VRAMs would be required, each holding one half of the eight bit pixel. In this situation, then, the expansion requires a different algorithm, namely the reordering of the ordinate position of the compressed bits by 8 positions. It is recognized that all VRAMs comprising the same pixel are preferably provided the same identical control signal. Thus, for a 2 VRAM pixel (for example, 8 bits) two positions of the bus preferably reflect the same compressed bit value.

There are two options for performing the swizzle. One is to create a larger, i.e. 64 lead, bus. This requires additional or larger VRAMs and more circuitry for controlling the bus. The other option is to have a different swizzle pattern in the swizzle circuit. In both cases the compressed data must control more than one VRAM if the pixel is contained in more than one VRAM.

The memory addressing is adjusted to correspond to the larger amount of data being written to the VRAM when performing a series of block-write accesses (such as filling a large screen area). Effectively, the 4 data bits going to a VRAM are expanded internally by a factor of 4 in the block-write mode. Thus a 32-bit data bus is expanded to 128 bits inside the VRAMs in block-write mode. Therefore, to step efficiently from one addressable location to the next adjacent one requires that the address be incremented/decremented (depending on direction) by 128 (in terms of the bit address) rather than

32 as would be done in regular addressing.

The swizzle operation in one example is realized by the proper connection of a multiplexer function for each given bit position. The multiplexing would select between the normal (or straight pass) mode and one or more swizzle functions as needed.

It is a technical advantage of this invention to provide a mechanism for writing pixels to a memory array both in normal mode as well as block write mode in a consistent manner.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and further advantages thereof, reference is now made to the following Detailed Description, taken in conjunction with the accompanying Drawings, in which:

FIGURE 1 shows a stylized view of a VRAM memory;
FIGURE 2 shows a VRAM memory connection to a data bus;
FIGURE 3 shows a swizzle circuit connected to the data bus;
FIGURES 4 and 5 show partial connections for alternate swizzle circuits;
FIGURE 6 shows a four position expansion;
FIGURE 7 shows the swizzle circuit cross-connections for all situations;
FIGURE 8 shows one embodiment of a swizzle circuit; and
FIGURE 9 shows an embodiment of a swizzle circuit used for several different memory configurations.

DETAILED DESCRIPTION OF THE INVENTION

Turning now to FIGURE 1, a brief discussion of the memory structure of a typical graphics memory system is in order before progressing to the actual detailed description of the functioning of the embodiment of this invention. While there are many memory structures and systems which could be used, in the preferred embodiment it is typical to use a structure such as shown in FIGURE 1 which uses eight VRAM memories 200, 201, etc. in an array. Each VRAM memory or unit has a 4 bit data port which can be treated as having planes 11, 12, 13 and 14. The construction of each plane is such that a single data lead is used to write information to that plane. These leads are labeled 0, 1, 2, and 3 for each plane. In a system that uses a 32 bit data bus, such as data bus 20, there would be 8 VRAM memories (two of which are shown in FIGURE 1) each memory having four data leads connected to the data bus.

Thus, for a 32 bit data bus VRAM memory 200 would have its four data leads connected to data bus leads 0, 1, 2, 3, respectively. Likewise, VRAM memory 201 would have its four leads 0, 1, 2, 3 connected to data bus leads 4, 5, 6, 7, respectively. This continues for the remaining six VRAM's such that the last VRAM has its leads connected to leads 28, 29, 30, 31 of bus 20. The full set of connections is shown in FIGURE 2.

Continuing with FIGURE 1, the memories are arranged such that the pixel information for the graphics display is stored serially across the planes in the same row. Assuming a four bit per pixel system, then successive pixels are stored in successive VRAMs. In such a situation pixel 0 would be in VRAM 200, and pixel 1 would be in VRAM 201. The pixel storage for pixels 2 through 7 are not shown in FIGURE 1 but are shown in FIGURE 2. The pixel information for pixel 8 then would be stored in VRAM 200, still in row 1 but in column 2 thereof. The reason for this arrangement of pixel information will be more fully appreciated from an understanding of how information is retrieved from the memory.

Continuing with FIGURE 1, each VRAM plane has a serial register 16 for shifting out information from a row of memory. The outputs from these registers are connected to data out bus 15 in the same manner as the data input leads are connected to the data input bus. Thus, data from a row of memory, say row 1, would be moved into register 16. This would occur for each plane of the eight memory array.

Looking at data output bus 15 at an instant of time, the first bit in each shift register would be on the bus. Thus assuming row 1 was being outputted to the bus, the bus would have on its lead 0 the row 1 bit A1 of memory 200. output bus 15 lead 1 would have on it row 1, bit B1; lead 2 would have row 1, bit C1; and lead 3 would have on it row 1, bit D1. These bits would be followed by memory 201 row 1 bits, A1, B1, C1, D1 on leads 4, 5, 6, 7, respectively. Thus, at a first instant of time, data out bus 15 would have on it the four bits forming pixel 0 followed by the four bits forming pixel 1, followed by the four bits forming pixel 2. This would continue until the 32 bits forming the 8 pixels 0-7 were on the consecutive leads of data out bus 15. These bits would be supplied to the graphics display and the shift registers would all shift one position providing the bus with pixel information for the next 8 pixels, namely pixels 8 through 15. This shifting would then continue until the entire line was shifted out and then a new line would be selected for loading into the output register.

Up to this point we have assumed that the bit information per pixel is 4 bits. If the pixel information were to be, say 8 bits, then two 4 bit wide VRAMs would have to be used for each pixel. This would change the bit patterns somewhat. This aspect of the invention will be discussed in further detail hereinafter. Also, it should be noted that memory sizes and structures continue to vary and the size and structure shown are only for illustrative purposes and this invention can be used with many different memory configurations and with different pixel sizes.

It must be noted that the depiction of memory in FIGURES 2 through 5 is a one-dimensional representation of what is conceptually a three-dimensional array as shown in FIGURE 1. Therefore, from this point, on the term "row" refers to the set of pixels addressed at any one time from the bus.

Turning now to FIGURE 2, a full eight VRAM memory arrangement is shown with the information for controlling pixels 0-7 contained in the top row of VRAMs 200 through 207, while pixels 8 through 15 are in row 2, and pixels 16 through 23 are in row 3, and pixels 24 through 31 are in row 4. This arrangement continues for each additional row of memory.

For normal write operations to the VRAM memory, bits of data are received over data bus 20. The position of the information on the bus determines where the data is to be stored in the VRAMs. Thus, a bit on lead 0 of bus 20 goes onto lead 0 of VRAM 200. Assuming the address location of the first row of VRAM 200 has also been selected, that bit information would become associated with bit 0 of pixel 0. This is the well known traditional operation of graphics systems and details of this operation will not be undertaken here. It is sufficient for our understanding of this invention to note that a given data word, such as data word 21, has bits in ordinate position and these bits will be transferred directly to the proper bit positions within the VRAMs because of the physical connections and associations between the data bus and the VRAMs. Also note that information in ordinate positions 0-3 of data word 21 can go, via bus 20, to one of many pixels 0, 8, 16, 24, 32, etc. The actual storage location will depend upon other concurrent addressing to the VRAMs, all of which is not shown here but is well known in the art.

The method of presentation of data as described above requires 22 bits of data, and a full memory write cycle for each row (8 pixels). In some situations, for example, when a background color is to be painted on a screen, many pixels will have the same information written to them. The block-write method of loading a VRAM has been devised to handle this situation. This operation, which is well known in the art, uses a special register on each VRAM, such as register 210 shown in conjunction with VRAM 200, which contains bits for transfer to selected pixel locations within memory. These bits are loaded prior to the start of any block-write operation.

During the block-write operation the memory is loaded in a manner different from normal loading. The four data input leads are used, but this time each bit controls the transfer of the special register bits to a particular memory row in that VRAM. For example, in VRAM 200 assume it is desired to load pixels 0, 8 and 24 with the bits from register 210 while leaving pixel 16 unchanged. In this situation, leads 0, 1, 3 would have logical 1's thereon while lead 2 would contain a logical 0. This same situation would prevail for the entire 32 bit bus in that the ordinate position of the bits would determine whether or not information is to be transferred into a corresponding pixel in a corresponding VRAM memory row. This, it will be appreciated, is different from the normal loading of data where the data itself comes from the data bus. For block-write operations, the data comes from the special registers associated with each VRAM and the bits on the data bus merely give on-off or load-not load control depending upon their position on the various leads of the bus.

The data word that controls this operation is then said to be in compressed format such that the ordinate position of each bit being either a 1 or 0 controls a function. Also it should be noted that 1 and 0 representing on and off, respectively, is merely illustrative and the reverse may be true also.

Turning now to FIGURE 3, it will be seen that compressed data word 39 has ordinate positions 0-31 which must be presented to the VRAMs to control various pixels in accordance with the ordinate position of the data in the word. Thus, pixel 0 is to be controlled by compressed data bit 0, while pixel 1 is to be controlled by compressed data bit 1. In this manner, compressed data bit 31 should then control pixel 31. This is easier said than done.

Pixel 0 is easy since it is controlled by lead 0 of VRAM 200 which is connected to compressed bit 0. However, the bit in position 1 of compressed data word 39 begins the problem. In FIGURE 2 this non-compressed bit is connected to pin 1 of VRAM 200. However, as discussed above, the bit in compressed data ordinate position 1 is used to control the writing of information from the special register into pixel 1. Pixel 1 is controlled, in turn, by a 1 or 0 on lead 1 of VRAM 201. This lead, in turn, is connected to lead 4 of bus 20. A comparison of FIGURES 2 and 3 will show that in one situation bit position 1 of the input data word goes to lead 1 of bus 20 while in the other situation it goes to lead 4. Thus, clearly a reordering of bits is necessary when compressed words are used to control data transfer in the block-write mode.

This reordering is accomplished by swizzle circuit 32 which is interposed between the compressed data input and the actual data bus. Swizzle circuit 32 is controlled by the processor to allow data to flow straight through, as would be the situation for FIGURE 2, or to reorder the leads in a certain pattern as is required for FIGURE 3. This arrangement does not require processor time to rearrange information, but rather establishes a pattern based on the physical structure of the memory bus arrangement and calls upon that structure whenever a block-write operation is invoked.

The swizzle circuit could be hard wired or could be software controlled within or outside of the processor.

Now let us assume that instead of four bits per pixel it is desired to use eight bits per pixel and retain a 32-bit data bus. Also let us assume that we continue using VRAMs having four planes per unit as discussed with respect to FIGURE 1. In such a situation the reordering of the bits from the compressed word would be different

than it was when only four bits per pixel were used. This can easily be seen in FIGURE 4 where VRAMSs 200 and 201 now both contain pixel 0 information, while VRAMs 202,203 contain pixel 1 information.

It follows then that while again compressed data bit 0 continues to be associated with lead 0 of VRAM 200, all the other ordinate positions of the compressed word are associated with different leads of the bus. Take for example compressed word ordinate position 2. In FIGURE 3, compressed data word ordinate position 2 is associated with pixel 2 and bus lead 8. However, in FIGURE 4 the association is with bus lead 16. This then argues for a separate swizzle for systems where there is different pixel configurations. Also, since half of each pixel is contained in a separate VRAM, both halves are controlled by the same compressed data control bit. Thus, each compressed data control bit must be duplicated once for each additional VRAM which contains part of a given pixel. This also argues different swizzles for each pixel configuration.

From FIGURE 4 it is clear that because each bit of the compressed word connects to two VRAM inputs that only 16 bits of the compressed word will control all of the VRAMs in a 32 bit bus configuration. The first system for solving this problem is to maintain the 32 bit bus and take two bus cycles to use both halves of the 32 bit compressed word. The other option is to use all 32 bits of the compressed word which expands the data bus to 64 bits.

FIGURE 9 shows a schematic diagram of how a simple multiplexer would achieve the required swizzle for output bits 0, 1, and 2 for supporting the 4 plane and 8 plane modes of the preferred embodiment. In normal mode the multiplexer function simply passes the corresponding bit position from input to output (i.e. 0 to 0, 1 to 1, and 2 to 2). For the 4 plane mode selection, the input to output connections are made as outlined in FIGURE 4 (0 to 0, 8 to 1, 16 to 2). For the 8 plane selection, the connections are made as outlined in FIGURE 5 (0 to 0, 0 to 4, 8 to 2). Of course, other multiplexer functions could be made to support other numbers of planes and different bus organizations.

While in the preferred embodiment, the swizzle function is performed by multiplexer hardware function, other means such as a software based table lookup method could be used to perform the swizzle.

Turning to FIGURE 5, it is seen that expanding the compressed word by duplicating each bit corresponding to the number of VRAMs used per pixel will result in the ability to use the same swizzle circuit for different memory/pixel configurations. This solution, as performed by duplicating/expansion circuit 52 has the effect of also activating both VRAMs of a given pixel, since the color information must be provided to all pixel bits even when these bits are positioned within two VRAMs.

The essence of the operation is the fact that the duplication and expansion occurs prior to the swizzle operation, thereby allowing the same swizzle configuration for both operations. In typical operations the same configuration would be used for any given system and thus only one determination of duplication/expansion need be made. However, situations may arise where more than one VRAM system configuration is controlled by the same processor, and thus dynamic control can be required. This can easily be achieved by arranging duplicate/expansion circuit 52 to function under control of the system processor on a case by case basis.

Duplicate/expansion circuit 52 can be any type of register circuit or processor that can reorder and pad numbers. This can be operated by microcode under control of the main processor or by a special processor or can be performed by a host processor if desired. The function performed by circuit 52 is mathmatical in nature and thus one skilled in the art can easily devise many arrangements to perform the desired function.

Circuit 52 can be system adaptable to change the duplicating and expansion function on a dynamic basis in response to received data or in response to a flag in a register to allow for changing pixel/memory configurations. Thus, for a pixel size of 16 bits and a VRAM of the same size as shown in FIGURE 1, namely four bits, four VRAMS would be used for each pixel and thus the expansion would be by four bits. In this situation, as shown in FIGURE 6, expanded word 61 would have the data from compressed bit ordinate position 0 expanded into ordinate positions 0, 1, 2, 3 of the expanded word. In this situation the data from compressed ordinate position 1 would be expanded into ordinate bit positions 4, 5, 6, 7, and so forth.

It can be seen from the chart in FIGURE 7 that the duplicated data at the inputs 0, 1, 2, 3 of the swizzle circuit go to outputs 0, 4, 8, 12. From FIGURE 4 it can be seen that these outputs go to VRAMS 200,201,202,203 which are the four VRAMS which would hold pixel 0 if that pixel were to be 16 bits long.

The compressed word is provided in a register such that it can be rotated through all 32 bits for any given memory clock cycle regardless of how many bits are expanded. This allows for continuous system operation without regard to pixel size. This also allows for total flexibility of memory storage to allow for starting and stopping at any given pixel boundary.

FIGURE 7 shows the input to output correspondence of swizzle circuit 32 when the swizzle circuit is in the swizzle modes It should be realized that each input has two possible outputs: the swizzle output, as shown, and the straight-through output, which is not shown. Of course, the straight-through output has input 0 connected to output 0, with input 1 connected to output 1, input 2 connected to output 2, and so forth. A switching circuit is used to switch between the straight-through arrangement of the swizzle circuit and the swizzle mode of the swizzle circuit. FIGURE 8 shows one embodiment of the swizzle circuit 32 where registers 0 and 1 are shown for positions 0 and 1.

As shown in FIGURE 8, the input bus has 32 leads,

and the output bus also has 32 leads. Between these leads are a number of latches, two of which, 800 and 801, are shown. Each latch has a single input connected to an individual input bus lead and two outputs connected to the straight-through correspondence and to the swizzle correspondence in accordance with FIGURE 7. The latches load in a straightforward manner from information on the input bus upon the signal provided on the load lead. For the straight-through operation, a signal is provided on the REGULAR lead, and the outputs from the latches are clocked straight through the swizzle circuit with straight-through correspondence, as noted above. However, when swizzle circuit 32 is being utilized in the swizzle mode, the SWIZZLE lead is pulsed, and this serves to switch the outputs. For example, with respect to latch 801, in the straight-through mode, latch 801 is connected to lead 1 of the output bus. However, in the swizzle mode, as can be seen, another output from latch 1 is connected to lead 4 of the output bus. All of the latches of swizzle circuit 32 are wired with this correspondence such that the swizzle output lead of each latch is connected as shown in FIGURE 7 to the output bus lead. This arrangement allows for the selective control of swizzle circuit 32 in the straight-through mode or the swizzle mode, under control of the system processor.

The circuit shown in FIGURE 8 can be expanded to cover the multiple swizzles required for swizzle circuit 42. In this situation, an extra controlled output lead would extend from each latch to a different output. In this mode a second swizzle control signal would extend to control multiple outputs from each latch, the number of multiples being a function of the number of VRAMs containing the same pixel information.

While the circuit and method shown here has been described in terms of the block-write operation of a graphics processing system it can be used in numerous other situations where ordinate coordination is required for controlling physical adaptations. It should be noted that the circuitry, including the swizzle circuit and processor, could be integrated into a single chip.

While the discussion has referred to the block-write mode as it relates to VRAMs, it should be understood that the same type of memory operations could be added to memories not specifically intended to support video.

Although the present invention has been described with respect to a specific preferred embodiment thereof, various changes and modifications may be suggested by one skilled in the art, and it is intended that the present invention encompass such changes and modifications as fall within the scope of the appended claims.

Claims

1. A video graphics circuit comprising:

   a first plurality of data input terminals;
   a memory (200,201,207) for storing data representing pixels of an image, each pixel being represented by a plurality of bits and the memory having a second plurality of data input terminals;
   a reordering circuit (32) for connecting the first plurality of data input terminals to the second plurality of data input terminals in different orders depending on whether the video graphics circuit is operated either

      in a first mode in which data presented at the first plurality of data input terminals are stored as pixel values in said memory or
      in a second, block write, mode in which single bits of the data presented at the first plurality of data input terminals determine whether a value held in a register is applied to respective pixels in the memory; and

   an expansion circuit (52) connected between the first plurality of data input terminals and the reordering circuit for replicating bits presented at the first data input terminals and applying the replicated bits in parallel to the reordering circuit.

2. A video graphics circuit as claimed in claim 1 wherein the reordering circuit is a swizzle circuit having a plurality of inputs and a like plurality of outputs, said swizzle circuit comprising:

   a like plurality of latches, each controlling one input and one or more outputs; and
   circuitry for controlling which output to which any input is connected at any instant of time.

3. A video graphics circuit as claimed in claim 1 wherein the reordering circuit is a multiplex circuit.

4. A video graphics circuit as claimed in claim 1 wherein the reordering circuit includes a memory having a look-up table.

5. A video graphics circuit as claimed in any one of claims 1 to 4 including circuitry for shifting bits presented at the first data input terminals before they are applied to the expansion circuit.

6. A video graphics circuit as claimed in any preceding claim wherein the memory is organised in planes and bits of a pixel are stored on respective planes.

7. A video graphics circuit as claimed in claim 6 wherein the memory comprises one or more VRAMS.

8. A video graphics circuit as claimed in claim 7, fur-

ther including circuitry for determining the number of the VRAMS containing the same pixel information and for controlling said expansion circuitry in accordance with said determination.

9. A video graphics circuit as claimed in claim 7, wherein said expansion circuitry is arranged for controlling the expansion when said graphics memory is configured with any number of VRAMS controlling said pixels.

10. A graphics processing system comprising a video graphics circuit according to any of claims 1 to 9.

Patentansprüche

1. Videographikschaltung mit:

mehreren ersten Dateneingangsanschlüssen;

einem Speicher (200, 201, 207) zum Speichern von Daten, die Pixel eines Bildes repräsentieren, wobei jedes Pixel durch mehrere Bits repräsentiert wird und der Speicher mehrere zweite Dateneingangsanschlüsse aufweist;

einer Umordnungsschaltung (32) zum Verbinden der mehreren ersten Dateneingangsanschlüsse mit den mehreren zweiten Dateneingangsanschlüssen in unterschiedlichen Anordnungen abhängig davon, ob die Videographikschaltung entweder

in einer ersten Betriebsart, in der Daten, die an den mehreren ersten Dateneingangsanschlüssen vorhanden sind, als Pixelwerte in dem Speicher gespeichert werden, oder

in einer zweiten, Blockschreibbetriebsart, in der einzelne Bits der an den mehreren ersten Dateneingangsanschlüssen vorhandenen Daten bestimmen, ob ein in einem Register festgehaltener Wert an entsprechende Pixel in dem Speicher angelegt wird,

betrieben werden; und

einer zwischen den mehrern ersten Dateneingangsanschlüssen und der Umordnungsschaltung angeschlossenen Erweiterungsschaltung (52) zum Reproduzieren von an den ersten Dateneingangsanschlüssen vorhandenen Bits und zum Anlegen der reproduzierten Bits in paralleler Form an die Umordnungsschaltung.

2. Videographikschaltung nach Anspruch 1, bei welcher die Umordnungsschaltung eine Mischschaltung mit mehreren Eingängen und einer gleichen Anzahl von Ausgängen ist, wobei die Mischschaltung folgendes enthält:

eine ebensolche Anzahl von Zwischenspeichergliedern, von denen jedes einen Eingang und einen oder mehrere Ausgänge steuert; und

eine Schaltung zum Steuern, welcher Ausgang mit welchem Eingang an jedem Zeitpunkt verbunden ist.

3. Videographikschaltung nach Anspruch 1, bei welcher die Umordnungsschaltung eine Multiplexschaltung ist.

4. Videographikschaltung nach Anspruch 1, bei welcher die Umordnungsschaltung einen Speicher mit einer Nachschlagetabelle enthält.

5. Videographikschaltung nach einem der Ansprüche 1 bis 4, mit einer Schaltungsanordnung zum Verschieben von Bits, die an den ersten Dateneingangsanschlüssen anliegen, vor ihrem Anlegen an die Erweiterungsschaltung.

6. Videographikschaltung nach einem der vorhergehenden Ansprüche, bei welcher der Speicher in Ebenen organisiert ist und Bits eines Pixels in jeweiligen Ebenen gespeichert sind.

7. Videographikschaltung nach Anspruch 6, bei welcher der Speicher eine oder mehrere VRAM-Komponenten enthält.

8. Videographikschaltung nach Anspruch 7, ferner mit einer Schaltungsanordnung zum Bestimmen der Anzahl der VRAM-Komponenten die die gleiche Pixelinformation enthalten und zum Steuern der Erweiterungsschaltung entsprechend dieser Bestimmung.

9. Videographikschaltung nach Anspruch 7, bei welcher die Erweiterungsschaltung so angeordnet ist, daß sie die Erweiterung steuert, wenn der Graphikspeicher mit einer Anzahl von VRAM-Komponenten konfiguriert ist, die die Pixel steuern.

10. Graphikverarbeitungssystem mit einer Videographikschaltung nach einem der Ansprüche 1 bis 9.

Revendications

1. Circuit graphique vidéo comprenant :

une première pluralité de bornes d'entrée de données ;
une mémoire (200,201,207) pour stocker des données représentatives d'éléments d'image d'une image, chaque élément d'image étant représenté par une pluralité de bits et la mémoire ayant une seconde pluralité de bornes d'entrée de données ;
un circuit de reordonnancement (32) pour relier la première pluralité de bornes d'entrée de données à la seconde pluralité de bornes d'entrée de données dans différents ordres suivant que le circuit graphique vidéo est commandé soit

dans un premier mode dans lequel des données présentes à la première pluralité de bornes d'entrée de données sont stockées en tant que valeurs d'éléments d'image dans ladite mémoire, ou
dans un second mode d'écriture par bloc dans lequel des bits uniques des données présentés à la première pluralité de bornes d'entrée de données déterminent si une valeur tenue dans un registre est appliquée auxdits éléments d'image respectifs dans la mémoire ; et

un circuit d'expansion (52) relié entre la première pluralité de bornes d'entrée de données et le circuit de réordonnancement pour répliquer des bits présents aux premières bornes d'entrée de données et appliquer les bits répliqués en parallèle au circuit de réordonnancement.

2. Circuit graphique vidéo selon la revendication 1, dans lequel le circuit de réordonnancement est un circuit d'aiguillage ayant une pluralité d'entrées et une pluralité analogue de sorties, ledit circuit d'aiguillage comprenant :

une pluralité analogue de circuits à verrouillage, chacun commandant une entrée et une ou plusieurs sorties ; et
un circuit pour commander quelle sortie est connectée à quelle entrée à tout instant.

3. Circuit graphique vidéo selon la revendication 1, dans lequel le circuit de réordonnancement est un circuit multiplex.

4. Circuit graphique vidéo selon la revendication 1, dans lequel le circuit de réordonnancement inclut une mémoire ayant une table à consulter.

5. Circuit graphique vidéo selon l'une quelconque des revendications 1 à 4, incluant un circuit pour décaler des bits présents aux premières bornes d'entrée de données avant qu'ils soient appliqués au circuit d'expansion.

6. Circuit graphique vidéo selon l'une quelconque des revendications précédentes, dans lequel la mémoire est organisée en plans et des bits d'une donnée d'image sont stockés sur des plans respectifs.

7. Circuit graphique vidéo selon la revendication 6, dans lequel la mémoire comprend une ou plusieurs VRAM.

8. Circuit graphique vidéo selon la revendication 7, comprenant en outre un circuit pour déterminer le nombre de VRAM contenant la même information d'élément d'image et pour commander ledit circuit d'expansion en fonction de ladite détermination.

9. Circuit graphique vidéo selon la revendication 7, dans lequel ledit circuit d'expansion est agencé pour commander l'expansion lorsque ladite mémoire graphique est configurée avec tout nombre de VRAM commandant lesdits éléments d'image.

10. Système de traitement graphique comprenant un circuit graphique vidéo selon l'une quelconque des revendications 1 à 9.

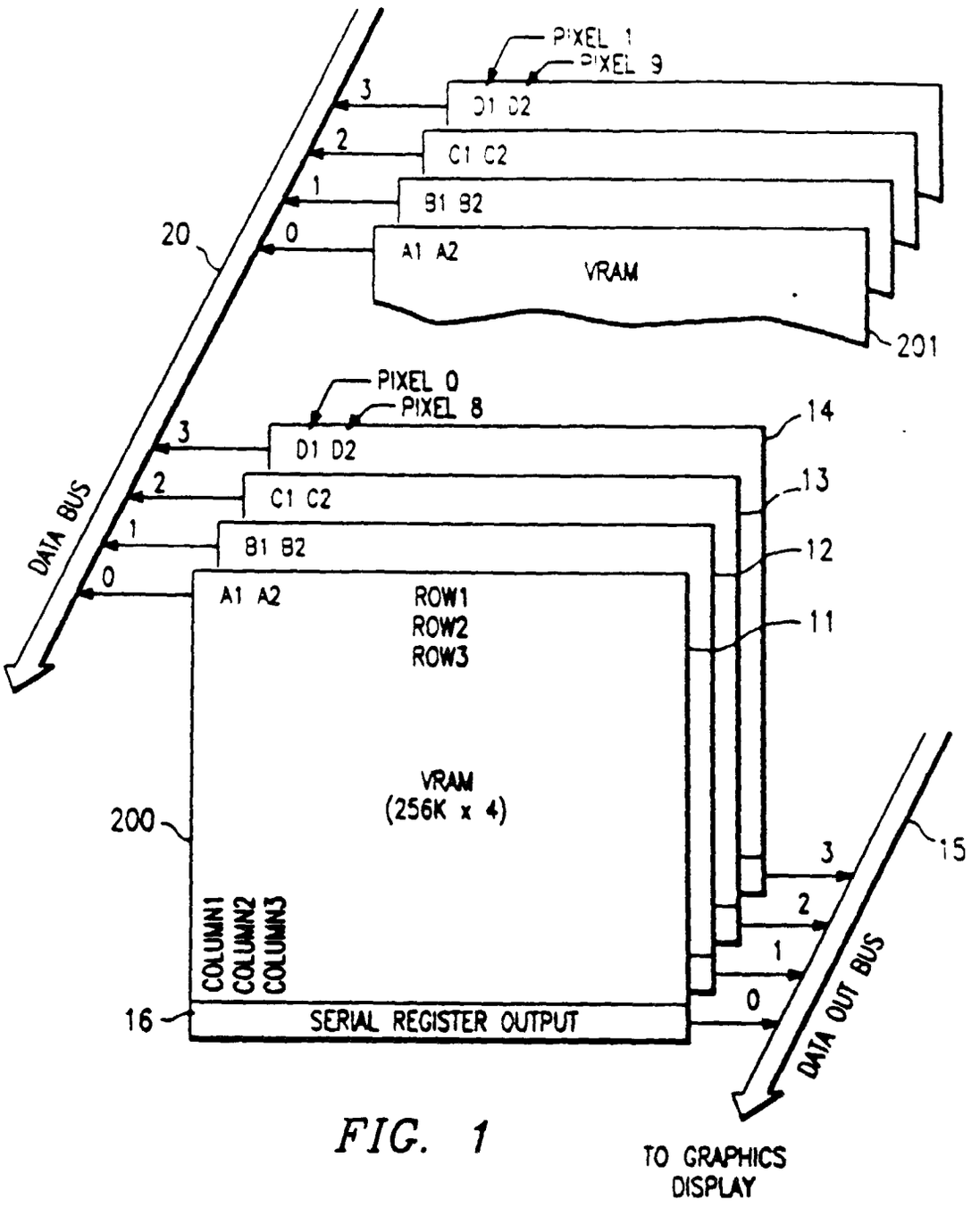

PIXEL 1
PIXEL 9
D1 D2
C1 C2
B1 B2
A1 A2
VRAM
201
20
PIXEL 0
PIXEL 8
D1 D2
C1 C2
B1 B2
A1 A2
ROW1
ROW2
ROW3
VRAM
(256K x 4)
COLUMN1
COLUMN2
COLUMN3
SERIAL REGISTER OUTPUT
200
16
DATA BUS
3
2
1
0
14
13
12
11
15
DATA OUT BUS
TO GRAPHICS
DISPLAY

FIG. 1

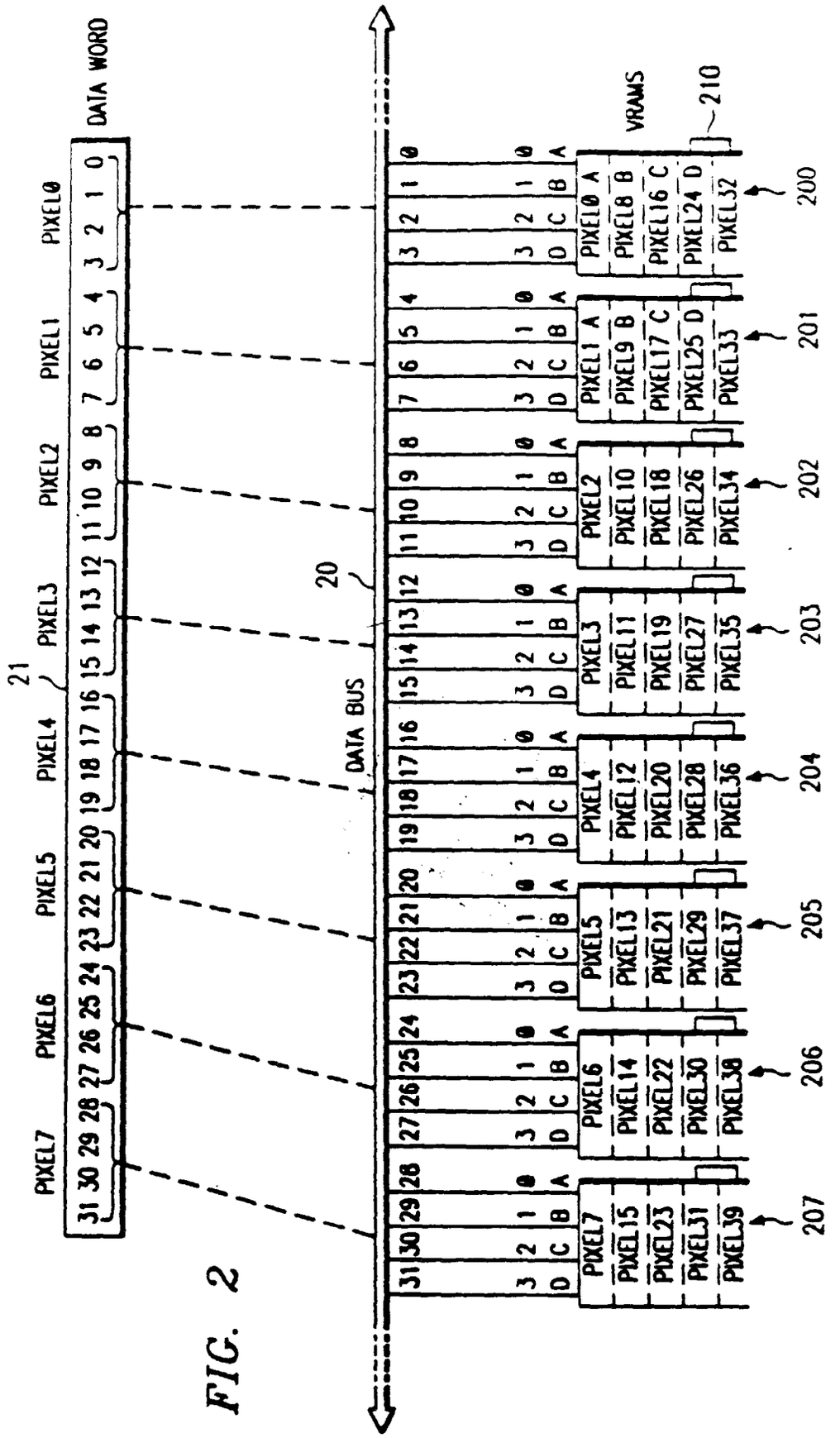

DATA WORD
PIXEL0
PIXEL1
PIXEL2
PIXEL3
PIXEL4
PIXEL5
PIXEL6
PIXEL7
21
20
DATA BUS
VRAMS
210
200
201
202
203
204
205
206
207

FIG. 2

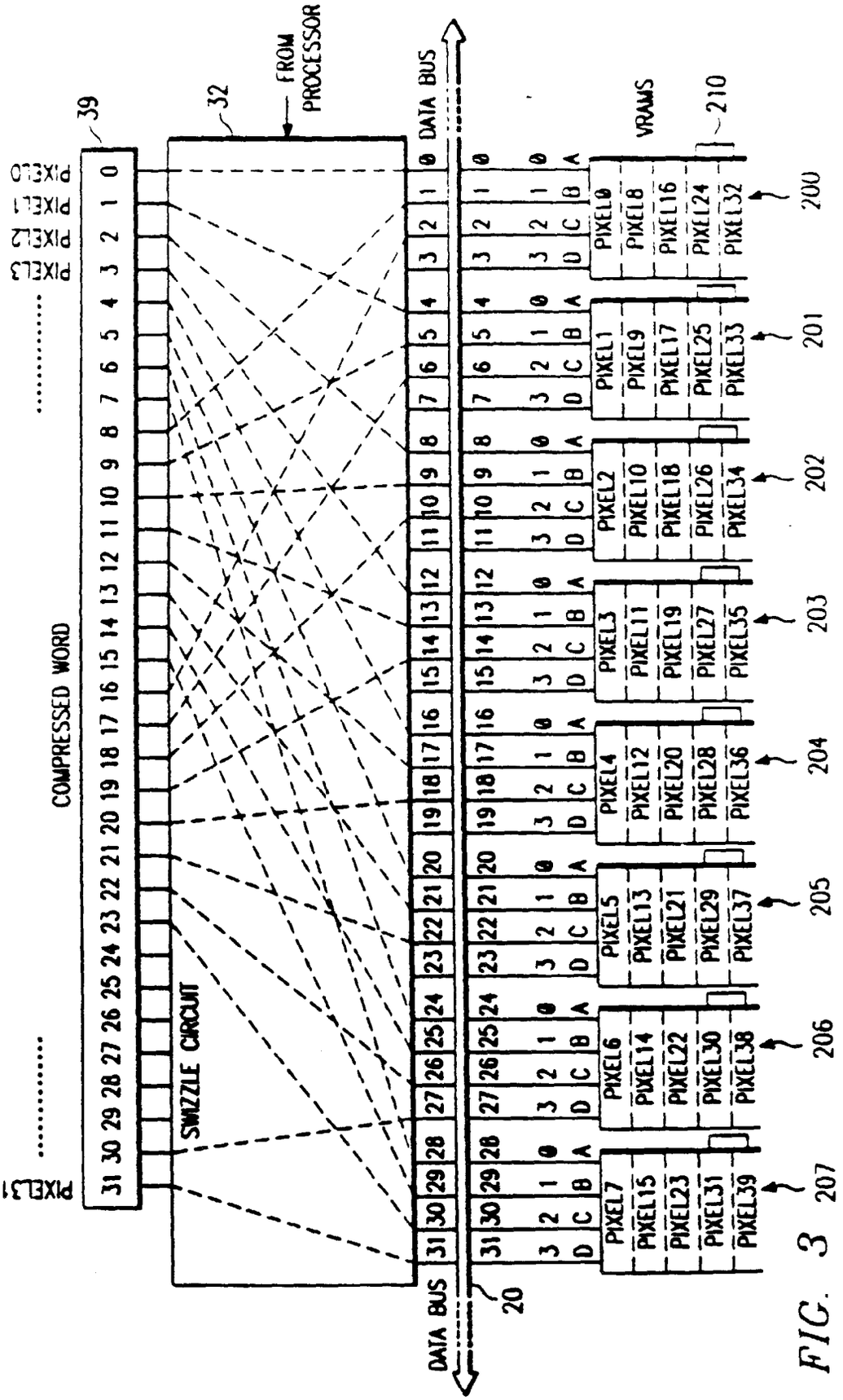

39
32
FROM PROCESSOR
DATA BUS
COMPRESSED WORD
SWIZZLE CIRCUIT
20
DATA BUS
VRAMS
210
200
201
202
203
204
205
206
207
PIXEL0
PIXEL1
PIXEL2
PIXEL3
PIXEL4
PIXEL5
PIXEL6
PIXEL7
PIXEL8
PIXEL9
PIXEL10
PIXEL11
PIXEL12
PIXEL13
PIXEL14
PIXEL15
PIXEL16
PIXEL17
PIXEL18
PIXEL19
PIXEL20
PIXEL21
PIXEL22
PIXEL23
PIXEL24
PIXEL25
PIXEL26
PIXEL27
PIXEL28
PIXEL29
PIXEL30
PIXEL31
PIXEL32
PIXEL33
PIXEL34
PIXEL35
PIXEL36
PIXEL37
PIXEL38
PIXEL39

FIG. 3

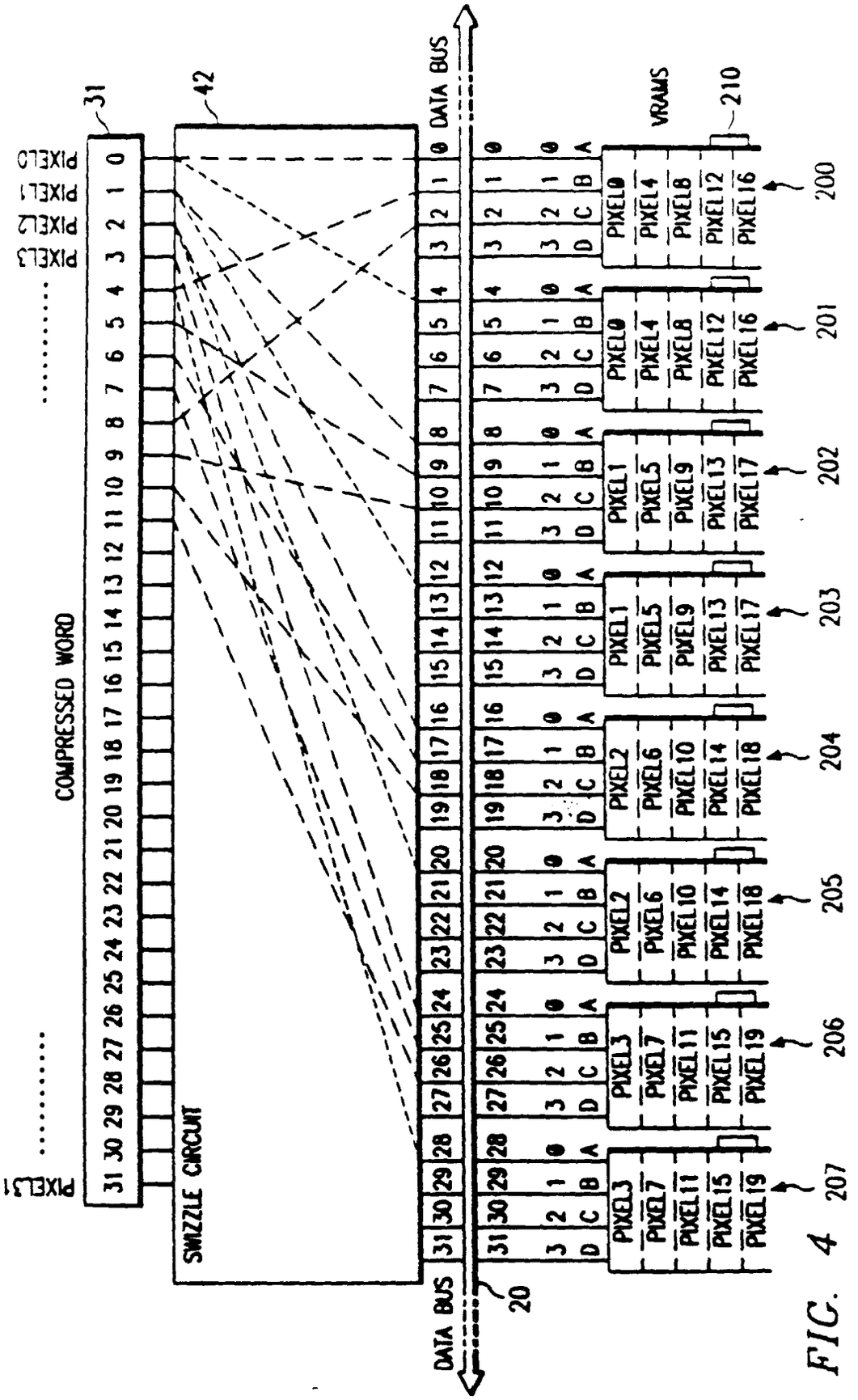

COMPRESSED WORD
31
42
SWIZZLE CIRCUIT
DATA BUS
20
VRAMS
210
200
201
202
203
204
205
206
207

FIG. 4

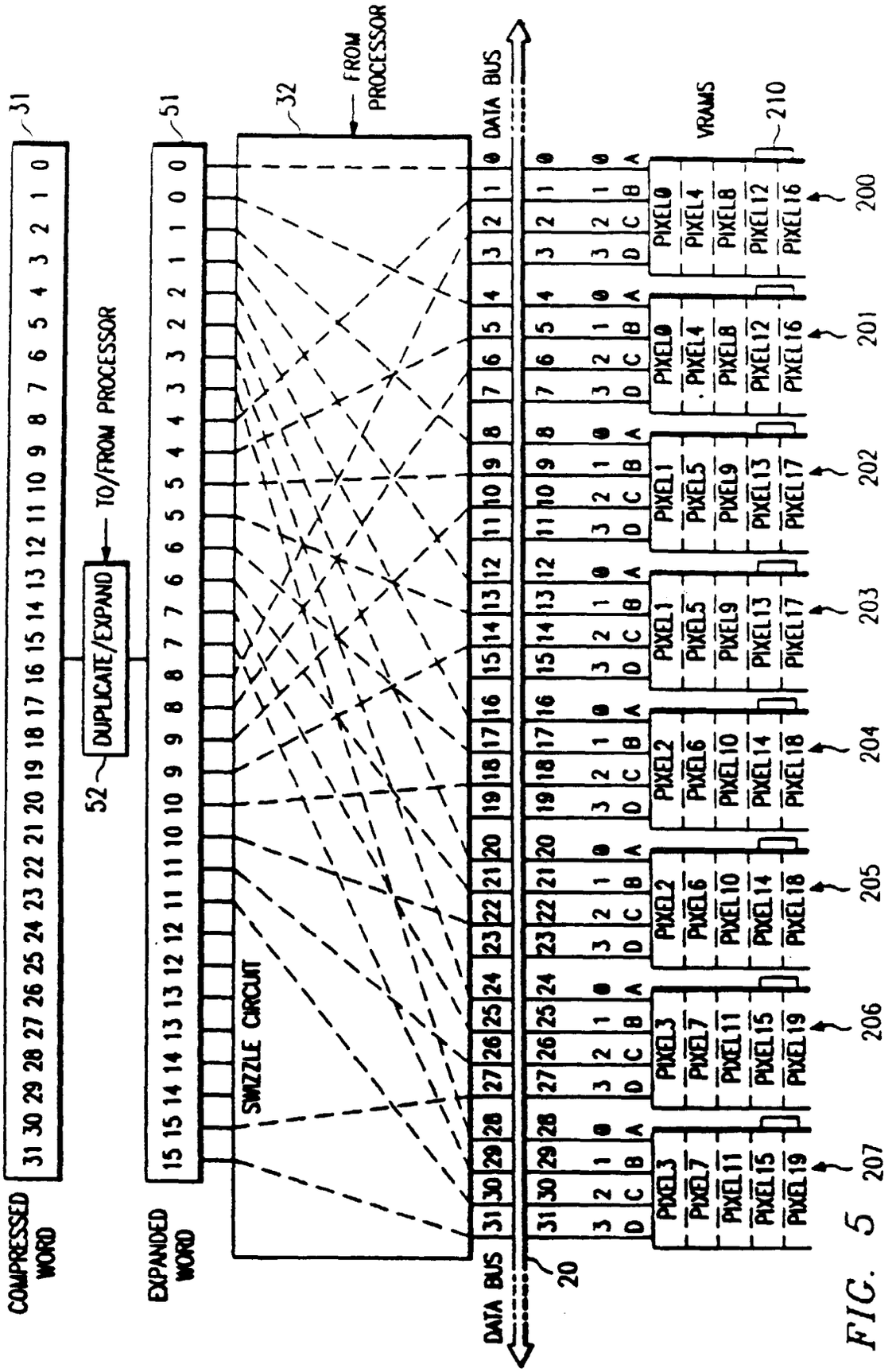
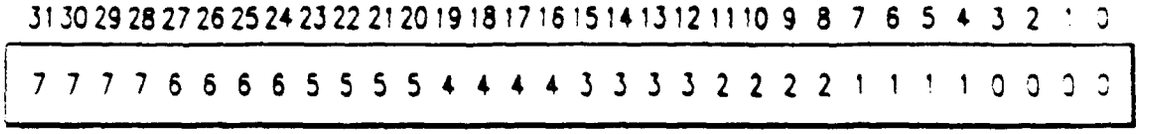

COMPRESSED WORD
31
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
52
DUPLICATE/EXPAND
TO/FROM PROCESSOR
EXPANDED WORD
51
15 15 14 13 13 12 12 11 11 10 10 9 9 8 8 7 7 6 6 5 5 4 4 3 3 2 2 1 1 0 0
SWIZZLE CIRCUIT
32
FROM PROCESSOR
DATA BUS
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
20
VRAMS
210
PIXEL0
PIXEL1
PIXEL2
PIXEL3
PIXEL4
PIXEL5
PIXEL6
PIXEL7
PIXEL8
PIXEL9
PIXEL10
PIXEL11
PIXEL12
PIXEL13
PIXEL14
PIXEL15
PIXEL16
PIXEL17
PIXEL18
PIXEL19
200
201
202
203
204
205
206
207

FIG. 5

FIG. 6



SWIZZLE CORESPONDENCE

INPUT	OUTPUT
0	0
1	4
2	8
3	12
4	16
5	20
6	24
7	28
8	1
9	5
10	9
11	13
12	17
13	21
14	25
15	29
16	2
17	6
18	10
19	14
20	18
21	22
22	26
23	30
24	3
25	7
26	11
27	15
28	19
29	23
30	27
31	31

FIG. 7

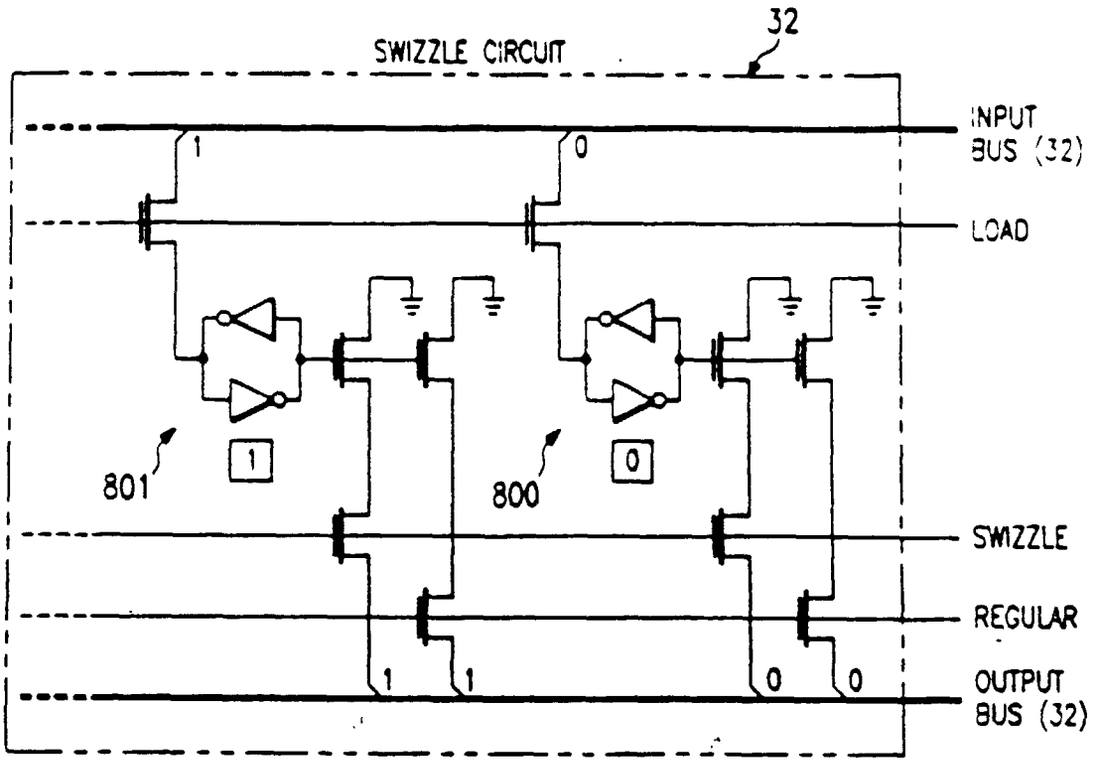

32
SWIZZLE CIRCUIT
INPUT BUS (32)
LOAD
1
0
801
800
SWIZZLE
REGULAR
OUTPUT BUS (32)

FIG. 8

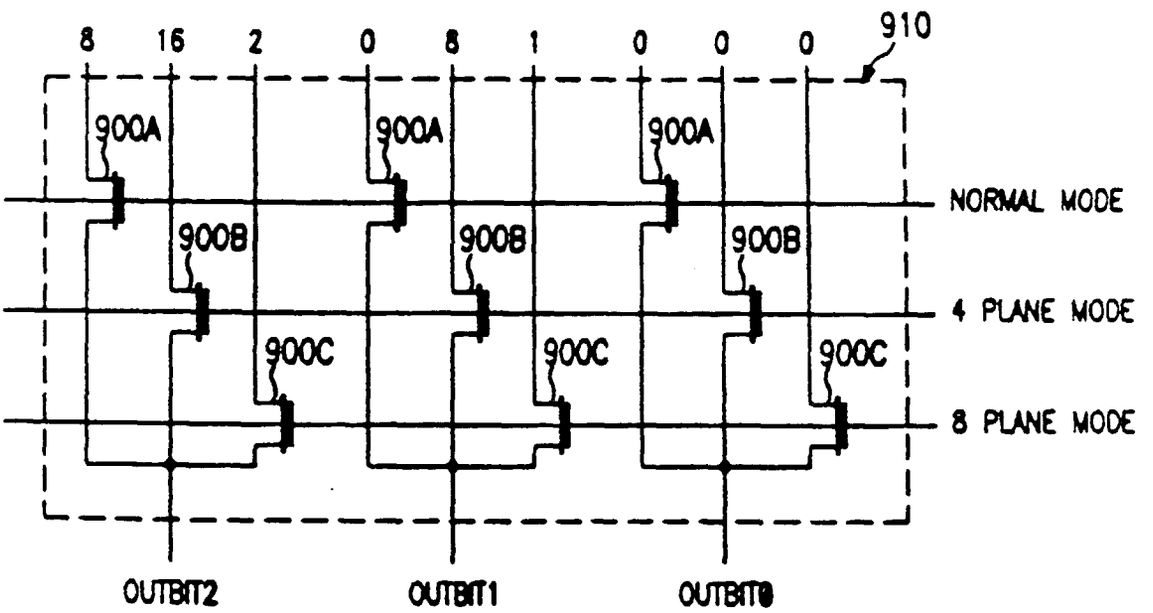

910
8 16 2 0 8 1 0 0 0
900A
900B
900C
NORMAL MODE
4 PLANE MODE
8 PLANE MODE
OUTBIT2
OUTBIT1
OUTBIT0

FIG. 9