(12) locations

(19) European Patent Office
Office européen des brevets

(45) Date of publication and mention of the grant of the patent:

(21) Application number: 93105175.9

(22) Date of filing: 29.03.1993

(51) Int Cl.®: G09G 1/16, G09G 5/36

(54) Picture storage apparatus and graphic engine apparatus
Bildspeichersystem und grafisches Anzeigesystem
Dispositif de mémorisation d'image et dispositif d'affichage graphique

(84) Designated Contracting States:
DE FR GB

(30) Priority: 30.03.1992 JP 102475/92

(43) Date of publication of application:
06.10.1993 Bulletin 1993/40

(73) Proprietor: SONY CORPORATION
Tokyo 141 (JP)

(72) Inventors:
• Ozawa, Hiroyuki
  Shinagawa-ku, Tokyo (JP)
• Yoshimori, Masaharu
  Shinagawa-ku, Tokyo (JP)
• Hayashi, Hiroshi
  Shinagawa-ku, Tokyo (JP)

(74) Representative: Melzer, Wolfgang, Dipl.-Ing. et al
Patentanwälte Mitscherlich & Partner,
Postfach 33 06 09
80066 München (DE)

(56) References cited:
US-A- 4 958 303

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

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention relates to a picture storage apparatus and a graphic engine apparatus, such as a display device in a computer graphics system.

2. Description of the Related Art

In a display apparatus employed in a computer graphic system or an engineering workstation, the drawing speed significantly influences the processing capability of the entire system and represents an important factor governing the processing capability. A variety of systems have been developed for elevating the drawing speed. Examples of these systems include a memory interleaving system and a block light system, such as a so-called picture cache system.

The pixel cache system and the memory interleave system are hereinafter explained.

A display apparatus constructed in accordance with the cache system includes a picture data generator 1 for decoding the commands and generating pixel data, a picture memory 2 having a storage capacity corresponding to the resolution and adapted for storage of pixel data, and a pixel cache memory 3 with a storage capacity of n x n pixels placed between the pixel data generator 1 and the picture memory 2 as shown in Fig. 1.

The commands supplied from a computer, such as commands for drawing a line or a surface, data transfer command in the picture memory, such as so-called BITBLT (Bit Block Transfer) command, or a fill command of filling the inside of a figure, are decoded by picture data generator 1 to generate pixel data. These pixel data are stored in a picture memory 2 via high-speed pixel cache memory 3 and pixel data stored in the picture memory 2 are read out in the form of the scanning by a Braun tube, not shown, by a raster scanning, for displaying a picture. That is, high-speed drawing is enabled by arranging the pixel cache memory 3, which permits high-speed accessing, between the picture data generator 1 and the picture memory 2. For example, if moving, copying or filing of a small figure can be made within the pixel cache memory 3 by e.g., a transfer command or a fill command, reading of pixel data from pixel memory 2 may be eliminated to raise the drawing speed.

However, with the pixel cache system, if the pixel cache memory 73 is of a small storage capacity and addresses indicating the positions of the pixel data from pixel data generator 1 on the display screen exceed boundary of the address region of the pixel data stored by the pixel cache memory 3, it becomes necessary to read and write pixel data between the pixel cache memory 3 and the picture memory 2 whenever the addresses traverse the boundary of the address region. Above all, the processing efficiency is significantly lowered when the picture memory 2 is random updated to access pixel data.

The display apparatus constructed in accordance with the memory interleave system includes a picture data generator for decoding commands and generating pixel data, an n number of memories m, where j = 0 through n - 1, having a storage capacity equal to 1/n corresponding to the resolution of the display screen and adapted for storing the pixel data, and an n number of memory controllers MP, n where j = 0 through n - 1, for controlling the n number of memories m.

The g number of memories mj make up a picture memory 6 corresponding to the display screen, with the memories mj dividing the picture memory 6 into e.g., 16 parts (N = 16), as shown in Fig. 3. If the pixel at the upper left corner on the display screen is an origin, the horizontal and vertical directions are x and y axes, and the pixels on the display screen are indicated by Pxy, where x, y are coordinates or pixel addresses on the display screen, the memories m, m, m, m, ..., m store pixel data of the pixels P1, P2, P3, ..., P16 (q, r = 0, 1, 2, ...).

For example, commands for drawing line segments or pictures from CPU, data transfer commands or fill commands are decoded by picture data generator 5 to generate pixel data which are stored in the memories mj based on addresses supplied commonly from pixel data generator 61 under control by memory controller MP. The stored pixel data in the memories mj are read out by raster scanning for displaying the picture on the Braun tube, not shown. That is, 16 memory controllers MP0 through MP15 access one of blocks Bxy. constituted by 4x4 pixels P1 through P16. based on block addresses (X, Y) (X, Y = 0, 1, 2, ....) supplied from picture data generator 5, for simultaneously accessing the 16 pixels in the block Bxy to realize a high processing rate, as shown in Fig. 17.

However, this memory interleave system has a drawback that, if the pixel addresses (x, y) indicating the positions on the display screen exceed the boundary of the block Bxy, the processing efficiency is lowered significantly. Besides, when accessing is made continuously to the pixels Pxy supervised by the same memory controller MP, the processing efficiency is similarly lowered significantly.

Taking an example of a data transfer command, such as BITBLT command, if pixel data of an arbitrary block Bxy of picture memory 6, for example block B0, 2, is to be transferred to block B2, 1, as shown by arrow t, the pixel data read
out based on the block address (0, 2) from pixel data generator 5 are written by the block address (2, 1) under control by each memory controller MP, as shown in Fig. 17. In this manner, data transfer may be achieved within the memory m controlled by the memory controller MP itself to realize high-speed data transfer. However, if the area of the originate or the area of the receiver are not coincident with the boundary of the block Bx,y, that is, if pixel data of block Bx,y are to be transferred, as shown by arrow 2, to an area 7 composed of 4×4 pixels Pxy as shown in Fig. 3, each memory controller MP is unable to transfer data within the memory m controlled by memory controller MP, so that data transfer cannot be achieved without communication between the memory controllers MP.

Consequently, if data transfer is to be effected in the above-described manner, it is necessary to effect communication, such as data transfer, between memory controllers MP, and subsequently to write pixel data four times into blocks B1,2, B2,3, B4,2 and B2,2 present in the area 7, thus lowering the processing efficiency.

With the above-described memory interleave system, which consists in dividing the picture memory 82 having the storage capacity conforming to the resolution of the display screen into an n number of sections and controlling the memories m by dedicated memory controllers MP for simultaneously accessing 16 pixels Pxy in the same block Bx,y to achieve high processing speed, the memory controller MP has no address generating function, as a result of which the processing efficiency tends to be lowered significantly.


Objects and Summary of the Invention

In view of the above-described status of the art, it is an object of the present invention to provide a picture memory apparatus operated in accordance with the memory interleave system, in which pixel data of an area of an arbitrary size on the display screen can be transferred at an elevated speed to an optional position. It is another object of the present invention to provide a graphic engine apparatus having a high drawing speed.

In accordance with the present invention, there is provided a picture memory apparatus comprising an n number of memory means (M) each having a storage capacity equal to 1/n of a storage capacity corresponding to the resolution of a display screen and each being adapted for storing a pixel data, an n number of control means (XP) each having first and second input/output ports for inputting and outputting the pixel data and each adapted for controlling readout and writing of pixel data and from the memory means via first input/output port, and bus connection means for commonly interconnecting the second input/output ports of the n number of said controlling means. The n number of controlling means perform a control operation of finding the number of controlling means of a receiver of the pixel data in synchronism, while performing a control operation of supplying pixel data read out from the memory means via first input/output port to controlling means of a receiver via input/output port and bus connection means and of causing control means of the receiver to write pixel data supplied thereto via bus connection means and the second input/output port in said memory means via first input/output port.

In accordance with the present invention, there is also provided a picture memory apparatus comprising an n number of memory means (M) each having a storage capacity equal to 1/n of a storage capacity corresponding to the resolution of a display screen and each being adapted for storing a pixel data, an n number of control means (XP) each having first and second input/output ports for inputting and outputting the pixel data and each adapted for controlling readout and writing of pixel data and from the memory means via the first input/output port, and bus connection means for commonly interconnecting the second input/output ports of the n number of the controlling means. The n number of controlling means perform a control operation of supplying pixel data read out from the memory means via first input/output port and bus connection means to controlling means of a receiver via the second input/output port and the bus connection means, while also performing a controlling operation of finding the number of the controlling means of an originate in synchronism and of causing control means of a receiver to write pixel data supplied thereto from controlling means of an originate via the bus connection means and the second input/output port in the memory means via the first input/output port.

In accordance with the present invention, there is also provided a graphic engine apparatus comprising a memory for storage of commands for picture processing, a setup processor (SP) for sequentially reading out the commands stored in the memory for calculating parameters necessary for generating pixel data, a rendering processor (RP) for supervising data flow for graphic processing, a pixel data generating circuit (LP) for generating pixel data responsive to parameters and commands for generating pixel data from the setup processor, a pixel memory unit for storing pixel data from the pixel data generating circuit, and a video processing unit for converting pixel data read out from the pixel data generating circuit. The picture memory unit includes an n number of memory means (M) each having a storage capacity equal to 1/n of a storage capacity corresponding to the resolution of a display screen and each being adapted for storing a pixel data, an n number of control means (XP) each having first and second input/output ports for inputting and outputting the pixel data and each adapted for controlling readout and writing of pixel data and from the memory means via first input/output port, and bus connection means for commonly interconnecting the second input/output ports of the n number of said controlling means. The n number of controlling means perform a control operation of
finding the number of controlling means of a receiver of pixel data in synchronism, while also performing a control operation of supplying pixel data read out from the memory means via the first input/output port to controlling means of a receiver via second input/output port and bus connection means and of causing control means of the receiver to write pixel data supplied thereto via bus connection means and second input/output port in said memory means via first input/output port.

In accordance with the present invention, there is additionally provided a graphic engine apparatus comprising a memory for storage of commands for picture processing, a setup processor (SP) for sequentially reading out the commands stored in the memory for calculating parameters necessary for generating pixel data, a rendering processor (FP) for supervising data flow for graphic processing, a pixel data generating circuit (LP) for generating pixel data responsive to parameters and commands for generating pixel data from the setup processor, a pixel memory unit for storing pixel data from the pixel data generating circuit, and a video processing unit for converting pixel data read out from the pixel data generating circuit. The pixel memory unit comprises an n number of memory means (M) each having a storage capacity equal to 1/n of a storage capacity corresponding to the resolution of a display screen and each being adapted for storing a pixel data, an n number of control means (XP) each having first and second input/output ports for inputting and outputting the pixel data and each adapted for controlling readout and writing of pixel data in and from the memory means via said first input/output port, and bus connection means for commonly interconnecting the second input/output ports of the n number of the controlling means. The n number of controlling means perform a controlling operation of supplying pixel data read out from the memory means via first input/output port and bus connection means to controlling means of a receiver via second input/output port and the bus connection means, while also performing a controlling operation of finding the number of the controlling means of an originate in synchronism and of causing control means of the receiver to write pixel data supplied thereto from controlling means of an originate via the bus connection means and the second input/output port in the memory means via the first input/output port.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a circuit of a conventional display apparatus operated in accordance with a pixel cache system.
Fig. 2 shows a circuit of a conventional display apparatus operated in accordance with a memory interleave system.
Fig. 3 shows the relative positions of the pixels on a display screen constituting the conventional display apparatus operated in accordance with the memory interleave system.
Fig. 4 shows a circuit of a graphic engine apparatus to which a picture memory apparatus according to the present invention is applied.
Fig. 5 shows a modified construction of the picture memory apparatus and the picture data generating apparatus making up the graphic engine apparatus shown in Fig. 4.
Fig. 6 shows relative positions of the pixels on the display screen controlled by each pixel processor constituting the picture memory apparatus shown in Fig. 5.
Fig. 7 shows another modified construction of the picture memory apparatus and the picture data generating apparatus making up the graphic engine apparatus shown in Fig. 4.
Fig. 8 shows relative positions of the pixels on the display screen controlled by each pixel processor constituting the picture memory apparatus shown in Fig. 7.
Fig. 9 shows a further modified construction of the picture memory apparatus and the picture data generating apparatus making up the graphic engine apparatus shown in Fig. 4.
Fig. 10 shows relative positions of the pixels on the display screen controlled by each pixel processor constituting the picture memory apparatus shown in Fig. 9.
Fig. 11 shows a concrete circuit arrangement of a pixel processor XP_i constituting the picture memory unit.
Fig. 12 shows a circuit arrangement of a main path circuit 21 consisting the pixel processor XP_i.
Fig. 13 is a diagrammatic view showing a source area and a destination area on a display screen.
Fig. 14 is a diagrammatic view showing the source area and the destination area for illustrating a formula for finding the number of the pixel processor XP_i of the destination.
Fig. 15 is a flow chart for illustrating the operation of a BITBLT command not accompanied by logical operation.
Fig. 16 is a flow chart for illustrating the operation of a BITBLT command accompanied by logical operation.
Fig. 17 is a timing chart for illustrating the operation of a BITBLT command not accompanied by logical operation.
Fig. 18 is a timing chart for illustrating the operation of a BITBLT command accompanied by logical operation.
Figs. 19 to 22 are diagrammatic views showing the source area and the destination area for illustrating the pixel transfer sequence in the data transfer command.
Fig. 23 is a diagrammatic view showing a source area for illustrating a start corner in the data transfer command.
Fig. 24 is a diagrammatic view showing a source area for illustrating a readout address for memory M_j constituting the picture memory unit.
Referring to the drawings, preferred embodiments of the picture memory apparatus according to the present invention will be explained in detail. Fig. 1 shows a circuit construction of a graphic engine apparatus to which the picture memory apparatus according to the present invention is applied.

First, the graphic engine apparatus is explained.

The graphic engine apparatus is made up of a main body of a workstation 11, a memory 12 connected to an inner bus of the main body of the workstation 11, a setup processor (SP) 13 for sequentially reading out commands from memory 12 for calculating parameters required for generating pixel data, a rendering processor (RP) 14 for supervising data flow of the graphic engine, a picture data generating circuit 15 for generating pixel data responsive to pixel data generating commands and parameters, a picture memory unit 16 for storing the pixel data from picture data generating circuit 15, a video processing circuit 17 for converting pixel data read out from picture memory unit 16 into RGB signals, and a Braun tube 18 for displaying the picture based on the RGB signals from the video processing circuit 17, as shown in Fig. 4.

With the present graphic engine apparatus, the main body of the workstation 11 and the memory 13 are interconnected by a Versa Module European (VME) bus, so that memory 12 temporarily stores commands concerning picture processing from the main body of the workstation 11, such as commands for drawing line segments or a plane, data transfer commands, fill commands, etc. The stored commands are sequentially read out by SP 13 and converted into commands for generating pixel data (referred to hereinafter simply as commands) and parameters. There commands and parameters are transmitted to picture data generating circuit 15. The picture data generating circuit 15 decides the commands to generate pixel data conforming to parameters, such as pattern information, color information, mask information, coordinate information on the display screen of the Braun tube 18, and commands, such as data transfer commands, these data being transmitted to picture memory unit 16. The picture memory unit 16 is a so-called bit map type memory and has a storage capacity corresponding to the resolution of a display screen of the Braun tube, such as 1024x1024 pixels. The picture memory unit 16 stores the pixel data of the respective pixels which are read out in the memory 12 in the order of scan, and transmitted to the video processing circuit 17. The video processing circuit 17 is constructed by e.g., a D/A converter and is adapted for converting the pixel data into RGB signals for displaying the picture based on the RGB signals on the Braun tube 18.

The graphic engine apparatus is also adapted for executing data transfer commands, such as bit block transfer (BITBLT) commands, by pixel data transfer within the picture memory unit 16.

Specifically, the picture memory unit 16 has an n number of memories M_i for storing pixel data and has a storage capacity of 1/n of a storage capacity corresponding to the resolution of the Braun tube 18, an n number of pixel processors XP_i (i = 1 ~ n - 1) each having first and second input/output ports IO_1, IO_2 for inputting and outputting pixel data and adapted for controlling reading and recording of the pixel data in and from memory M_i via first input port IO_1, and a bus (transfer bus) 19 for commonly connecting the second input/output ports IO_2 of the n number of pixel processors XP_i for effecting transfer of the pixel data between the pixel processors XP_i as shown in Fig. 4.

That is, the pixel memory unit 16 is constructed in accordance with the memory interleave system, and the memory 20 associated with the display screen of the bit map system is constituted by an n number of memories M_i. Each memory M_i is so constructed that, as described in connection with the prior art shown in Fig. 3, if the picture memory 20 is divided into 16 parts (N = 16), the pixels at the left upper corner of the display screen of the Braun tube 18 is an origin, the horizontal and vertical directions are denoted as x and y axes, and the pixels on the display screen are denoted by P_{x,y}, x and y being coordinates on the display screen, referred to hereinafter as pixel addresses, the memories M_0, M_1, M_2, M_3, ... M_15 store the pixel data for pixels P_{4q, 4r}, P_{4q+2, 4r}, P_{4q+5, 4r}, ... P_{4q, 4r+1}, P_{4q+1, 4r+1}, ... P_{4q+3, 4r+1} (q, r = 0, 1, 2, ...). Each memory M_i is of plural plane structure in case of a color display and is adapted for storing three color data, namely red color data, green color data, blue color data and depth data for three-dimensional display for each pixel P_{x,y}. One of the planes is used as a fill work buffer for executing filling.

On the other hand, the pixel processors XP_i are provided for memories M_i in a one-for-one relationship, such that, if the picture memory 20 is divided into 16 parts, the number of the pixel processors is also 16, as shown in Fig. 4. These pixel processors XP_0 ~ XP_15 supply pixel data from picture data generator 15 via input/output port IO_1 to memories M_0 ~ M_15, while generating associated addresses for controlling the recording in the associated memories M_i.

The pixel processors XP_i are adapted for being operated in synchronism with another one, such that, when data are transferred within the picture memory 20, the numbers j of the pixel processors XP_i of the pixel data receiver are found in synchronism, the pixel data read out via input/output ports IO_1 from memory M_i based on the number j are outputted at TBus 19 via input/output ports IO_2 in the increasing order of the number j of the receiver, the pixel data supplied from the pixel processor XP_i of the originate via TBus 19 and input/output ports IO_2 are received by the processor XP_j in the sequence of the increasing numbers j, and the received pixel data are simultaneously written in the memories M_j via input/output ports IO_1.

On the other hand, if the picture memory 20 is divided into 16 parts, the picture data generating circuit 15 is made
up of four picture data generators \( LP_0 \sim LP_3 \) and is adapted for decoding commands from SP 13 and generating pixel data responsive to parameters, as shown for example in Fig.4. Thus the pixel data generator \( LP_0 \) supply the generated pixel data to the pixel processors \( XP_0 \sim XP_3 \). the pixel data generator \( LP_1 \) supply the generated pixel data to the pixel processors \( XP_4 \sim XP_7 \). the pixel data generator \( LP_2 \) supply the generated pixel data to the pixel processors \( XP_8 \sim XP_{11} \) and the pixel data generator \( LP_3 \) supply the generated pixel data to the pixel processors \( XP_{12} \sim XP_{15} \).

It is noted that the picture memory unit 16 and the picture data generator 15 are not limited to the construction of dividing the picture memory 20 into 16 parts, as shown in Fig.4. For example, if the picture memory 20 is divided into four parts, a basic unit consisting of a picture data generator LP and four pixel processors XP as shown for example in Fig.5 is used, and each of the pixel processors \( XP_0 \sim XP_3 \) is adapted to control reading and writing of the associated pixel data of the corresponding pixels on the display screen of the picture memories, indicated by numbers 0 to 3, as shown in Fig.6.

For example, if the picture memory 20 is divided into eight parts, two of the above-mentioned basic units as shown for example in Fig.5 are used, and each of the pixel processors \( XP_0 \sim XP_7 \) is adapted to control reading and writing of the associated pixel data of the corresponding pixels on the display screen of the picture memories, indicated by numbers 0 to 7, as shown in Fig.8.

If the picture memory 20 is divided into 32 parts, eight of the above-mentioned basic units as shown for example in Fig.9 are used, and each of the pixel processors \( XP_0 \sim XP_{31} \) is adapted to control reading and writing of the associated pixel data of the corresponding pixels on the display screen of the picture memories, indicated by numbers 0 to 31, as shown in Fig.9. In sum, if the picture memory unit 16 is constructed in accordance with the memory interleaver system in which the picture memory 20 is divided into plural parts in connection with the resolution of the display screen and each of the pixel processors \( XP_j \) controls each memory \( M_j \) resulting from division. A concrete arrangement is hereinafter explained, in which the picture memory 20 is divided into 16 parts.

The pixel processor \( XP_j \) is made up of a main path circuit 21 for supplying pixel data supplied from the pixel data generators \( LP_0 \sim LP_3 \) via input/output port \( IO_0 \) to memory \( M_j \) and for outputting the pixel data read out from memory \( M_j \) via input/output port \( IO_j \) to TBus 19 via input/output port \( IO_0 \), an address decoder 22 for decoding addresses supplied from picture data generators \( LP_0 \) to \( LP_3 \) for controlling the so-called pipeline operation of the main path circuit 21, a sequencer 23 for controlling data flow through main path circuit, a memory controller 24 for controlling reading and recording of the pixel data in and from memory \( M_j \), a TBus controller 25 for finding the number \( j \) of the pixel processor \( XP_j \), of the receiver for deciding whether or not there is the transfer and for controlling the TBus 19, an address generator 26 for generating the addresses of a rectangular transfer area at the time of data transfer of e.g. BITBLT command for supplying the addresses to the sequencer 23, and a control register 27 for storing parameters for controlling the components from the main path circuit 21 to address generator 26, as shown for example in Fig.11.

The main path circuit 21 is essentially made up of transfer registers 31a, 31b for transiently storing pixel data read out via input/output port \( IO_j \) from memory \( M_j \) for outputting stored pixel data at TBus 19 via input/output port \( IO_0 \), a multiplexer 32 for changing over and selecting pixel data supplied read out from memory \( M_j \) via input/output port \( IO_j \) or pixel data supplied via inner data bus 46 provided in the main path circuit 21 and outputting the selected pixel data via input/output port \( IO_0 \) to TBus 19. reception registers 34a, 34b for transiently storing pixel data supplied from TBus 19 via input/output port \( IO_0 \) and supplying stored pixel data via input/output port \( IO_0 \) to memory \( M_j \), a multiplexer MUX 35 for changing over and selecting the pixel data supplied TBus 19 via input/output port \( IO_0 \) or the pixel data supplied via inner data bus 46, a multiplexer MUX 36 for changing over and selecting the pixel data read out from memory \( M_j \) via input/output port \( IO_j \) and the pixel data selected by multiplexer MUX 35, a multiplexer MUX 37 for changing over and selecting one of the pixel data from registers 34a, 34b, a shift register 38 for shifting data of a predetermined plane of the pixel data deselected by MUX 37 for detecting control data of changing over the background color display or the foreground color display, a color converter 39 for generating color-converted pixel data based on control data from shift register 38, a register 40 for transiently storing destination data read from memory \( M_j \) via input/output port \( IO_0 \), a logical operation circuit 41 for recording pixel data obtained after logical operation of the pixel data from color-converter 39 and pixel data from register 40 in memory \( M_j \) via input/output port \( IO_0 \), buffers 42a, 42b for changing over the directions of the input/output port \( IO_0 \) and buffers 43a, 43b for changing over the directions of the input/output port \( IO_0 \), as shown for example in Fig.12.

Meanwhile, buffers 44a, 44b are designed for increasing the number of fan-outs, while registers 45a, 45b, 45c and 45d are designed for performing a pipeline processing operation. For data transfer commands, multiplexers MUX 32, 35 select pixel data from buffer 44a, multiplexor MUX 36 is fixed for selecting pixel data from buffer 44b, multiplexor MUX 33 is controlled by control signal SSC from sequencer 23, and multiplexor 37 is controlled by control signal SRC from sequencer 23. Registers 31a, 31b, 34a, 34b are controlled by enabling signals permitting of writing from sequencer 23.

The above-described main path circuit 21 is controlled by sequencer 23 and memory controller 24, so that data transfer between memories \( M_j \) is carried out over TBus 19. Data transfer operation in case of non-coincidence of the area of the originat, referred to hereinafter as source area, and the area of the receiver, referred to hereinafter as
destination area, as viewed on the display screen, with the boundary of the block \(B_{x, y}\) shown in Fig. 3 and discussed in connection with the prior art, is hereinafter explained.

Each pixel processor \(XP_j\) is adapted for executing copy transfer of the rectangular area by the above-described PIXBLT command. With such command, the coordinate values \((x_{p}, y_{p})\) of a starting point which is a pixel at the left upper corner of the source area, the size \(W, H\) of the source area 51, and distances \(D_{xp}, D_{yp}\) up to the destination area 52, stored in memory 12, as shown in Fig. 13, are supplied via SP 13 and picture data generators LP\(_0\) \(\sim\) LP\(_3\).

The pixel processor \(XP_j\) finds the number \(j\) of the pixel processor \(XP_j\) of the destination by the formula (1):

\[
j = (((D_{yp} \ll 2) + i) \land \ln) \lor (((D_{xp} + i) \lor H) 3H)
\]

In the above formula (1), symbols \(\ll 2, \land\) and \(\lor\) indicate shift left by 2 bits, logical product and logical sum, respectively. It is noted that \(D_{yp}, D_{xp}\) in and \(3H\) are expressed by a hexadecimal notation, while the value of \(i\) is \(OH\ (O), 4H\ (4), CH\ (12), 1CH (20)\), where the bracketed number is a decimal number, in association with 4, 8, 16 and 32 of the numbers \(p\) of the pixel processors \(XP_j\) which number is the number of division of the picture memory 20.

Specifically, if, with the coordinate values \((x_{p}, y_{p})\) of a starting point of the source area 51 of \((10, 10)\), the size of the source area \(W, H\) of 8, and the distances \(D_{xp}, D_{yp}\) up to the destination area 52 of 9 or 13, the boundary of the source area 51 is not coincident with the boundary of the block \(B_{x, y}\) as shown for example in Fig. 14, each TBUS controller 25 finds the number \(j\) of the pixel processor \(XP_j\) of the destination area from formula 1 as \(5H \sim FH\ (15), OH\ (0) \sim 4H\ (4)\), to control the TBUS 19 based on this number \(j\). On the other hand, each address decoder 21 checks as to which is the number of times required of the transfer operations on the basis of the size \(W, H\) of the source area 51 to advance the sequencer 23 of the start of transfer, while advising it of the end of transfer when the required number of the transfer operations have come to an end.

On reception of the notice of start of transfer from address decoder 21, sequencer 23 starts the control of the main path circuit 21. In the BITBLT command which is not accompanied by a logical operation in which a predetermined processing is carried out on the pixel data transferred and the pixel data of the destination area 52 and the produced pixel data is adopted as pixel data of new destination area 52, sequencer 23 controls the flow of data in the main path circuit 21 in operative association with memory controller 24 and TBUS controller 25 in accordance with the flowchart shown in Fig. 15. Meanwhile, the left-hand side and the right-hand side of the flow chart show accessing operations to the TBUS 19 and to memory \(M_j\) respectively. The pixel data read from memory \(M_j\) of the origin are source data \(SD_{b,k}\) where \(k = 0, 1, 2, \ldots\), while pixel data received by the pixel processor \(XP_j\) of the destination is the reception data or received data.

The source data \(SD_{b,k}\) read out from memory \(M_j\) at timing \(T_0\) are latched by register 31b via input/output port \(IO_2\), buffers 42a, 44a and multiplexer MUX 32. The latched source data \(SD_{b,k}\) are outputted via multiplexer MUX 33, buffer 43a and input/output port \(IO_2\) at TBUS 19. The outputting sequence of the pixel processors \(XP_j\) is in the sequence of the increasing numbers \(j\) indicating the number of the pixel processor \(XP_j\) of the destination as found based on the above formula (1).

At timing \(T_1\), the source data \(SD_{b,k}\) as read out from memory \(M_j\) in readiness for data transfer for the next pixel is latched at register 31a. The reception data transferred over TBUS 19 from other pixel processors \(XP_j\), at the above-mentioned timing \(T_0\) is latched at register 34b via input/output port \(IO_2\), buffers 43a, 44b and multiplexer MUX 36. After the end of data transfer between the pixel processors \(XP_j\), the next source data latched by the register 31a is outputted over TBUS 19.

At timing \(T_2\), the reception data \(RD_{b,k}\) latched at register 34b at timing \(T_1\) is color-converted, if need be, and reception data \(RD_{b,k}\), for example, are directly set as write data \(WD_{b,k}\) which is supplied to memory \(M_j\) via logical operation circuit 41, register 45c, buffer 42b and input/output port \(IO_2\). That is, with the BITBLT command not accompanied by logical operation, the logical operation circuit 41 directly transmits the reception data \(RD_{b,k}\) to memory \(M_j\).

At timing \(T_3\), source data \(SD_{b,k}\) is latched in register 31b in readiness for next pixel data transfer, as in the above-mentioned operation at timing \(T_1\). Reception data \(RD_{b,k}\) transmitted from other pixel processors \(XP_j\), is latched at this time at register 34a. After the end of data transfer between the pixel processors \(XP_j\), the source data latched in register 31b is outputted at TBUS 19.

At timing \(T_4\), reception data \(RD_{b,k}\) from other pixel processors \(XP_j\) latched at timing \(T_3\) in register 34a, is color-converted, if need be, and reception data \(RD_{b,k}\), for example, is directly set as write data \(WD_{b,k}\) which is supplied to memory \(M_j\).

The above-described operations of reading out source data \(SD_{b,k}\) from memory \(M_j\) and outputting source data previously read out to TBUS 19 simultaneously and alternately via registers 31a, 31b and of receiving reception data \(RD_{b,k}\) transmitted over TBUS 19 and writing the previously received reception data as write data \(WD_{b,k}\) in memory \(M_j\) simultaneously and alternately by registers 34a, 34b, are repeatedly performed until data transfer to the totality of the
pixels within the source area 51 comes to an end. The result is that data transfer between the pixel processors $X_{P_i}$ may be completed substantially within the memory accessing time. In other words, accessing (reading and writing) of memory $M_1$ and accessing (data transmission and reception) of TBUs 19 may be performed simultaneously to enable high speed data transfer between the pixel processors $X_{P_i}$.

Meanwhile, with the BITBLT command, accompanied by the logical operation, writing in register 40 of destination data $D_{D_1}$, which is the pixel data of the destination from memory $M_1$ at timings T1, T4 or T7, is annexed, while a logical operation between destination data $D_{D_1}$ prior to writing of write data $D_{W_1}$ at timings T3, T6 and T9 in memory $M_1$ and optionally color-converted reception data $D_{R_1}$, for example, a logical operation between destination data $D_{D_1}$ and pixel data resulting from color conversion of the reception data $D_{R_1}$, is also annexed, as shown in Fig. 16. However, these operations may also be completed within the memory access time, so that, similarly to the case in which the BITBLT command is not accompanied by the above-mentioned logical operation, data transfer may be executed at a higher speed.

The accessing timing to the memory $M_1$ and TBUs 19 is explained in detail.

With the BITBLT command not accompanied by the logical operation, accessing of memory $M_1$ is started by the low (L) level of a request signal MREQ supplied from sequencer 23 to memory controller 24, as shown at $a$ in Fig. 17, while reading and writing of pixel data in and from memory $M_1$ are controlled by read/write signal R/W from memory controller 24, as shown at $b$ in Fig. 17, so that, when the read/write signal R/W is at the H level, memory $M_1$ is in the read mode. Memory controller 24 sets the read/write signal R/W to an H level, while sequentially supplying readout addresses indicating the pixel positions of the source area 51 to memory $M_1$, for sequentially reading the source data $S_{D_0}, S_{D_1}, S_{D_2}, ...$, as shown at $c$ in Fig. 17. This reading is equivalent to the above-mentioned reading operation at the timings T0, T1, T3, T5 - shown in Fig. 15. Memory controller 24 sets the read/write signal R/W to the L level and sequentially transmits write addresses indicating pixel positions in the destination area 52 as later explained to memory $M_1$, for sequentially writing data write data $W_{D_0}, W_{D_1}, W_{D_2}, ...$, as shown in Fig. 10c. This writing operation is equivalent to the writing operation at the above-mentioned timings T2, T4, T6, ... On the other hand, memory controller 24 transmits a next signal NEXT indicating the timing of the next operation as a low-level signal to sequencer 23 to permit the next operation, as shown in Fig. 10c.

On the other hand, the buffers $42a, 42b$ of the main path circuit 21 are controlled in timed relation to the above-mentioned read/write signal R/W so that buffer $42a$ is rendered operative when the read/write signal R/W is at the H level and buffer $42b$ is rendered operative when the signal R/W is at the L level. When enable signals $E_{N_1}, E_{N_2}$ supplied from sequencer 23 are at L level, as shown at $e, f$ in Fig. 17, registers $31a, 31b$ are enabled for writing, and alternately latch source data $S_{D_0}$, read out from memory $M_1$ via buffer $42a$, by clocks, not shown, which positively latch source data $S_{D_0}$ from sequencer 23 shown at $g$ in Fig. 17. That is, the source data $S_{D_0}, S_{D_1}, S_{D_2}, ...$ are latched alternately, similarly to the latching operation at the aforementioned timings T0, T1, T3, T5, ... The source data $S_{D_0}$, thus alternately latched by registers $31a, 31b$, are alternately selected by multiplexer MUX 33, so as to be outputted over TBUs 19 via buffer 43a and input/output port IO$_2$. Specifically, multiplexer MUX 33 selects source data $S_{D_0}, S_{D_1}$, k being an even number, from register $31b$ and source data $S_{D_0}$, k being an odd number, from register $31a$, when control signal SSC from sequencer 23 is at the L level or at the H level, respectively, as shown at $g$ in Fig. 17, and outputs the selected source data $S_{D_0}$ to TBUs 19 via buffer $43a$ controlled by controller 25.

TBUs controller 25 causes the number of clocks corresponding to the data transfer rate on TBUs 19 to be counted by the number $j$ indicating the number of the pixel processor $X_{P_j}$ of the destination area, with the L-level of the control signal TS controlling the start of data transfer supplied from sequencer 23 based on the aforementioned NEXT signal as a reference, as shown at $k$ in Fig. 17. As soon as the counting is terminated, buffer $43a$ is rendered operative to effect the outputting of the source data $S_{D_0}$ to TBUs 19, which is outputting is effected after the end of writing of the source $S_{D_0}$ to registers $31a, 31b$ at the aforementioned timings T0, T1, T3, ... Specifically, if the number $j$ of the pixel processor $X_{P_j}$ is 12 and the number $j$ of the pixel processor $X_{P_j}$ of the destination is 5, TBUs controller 25 of the pixel processor $X_{P_j}$ as indicated sets a control signal DIR controlling the direction of the input/output port IO$_2$ to L level within the time corresponding to the fifth time slot of the transfer data on TBUs 19, while setting enable signal $E_{N_5}$ to the L level, thereby rendering buffer $43a$ operative to output source data $S_{D_0}$ alternately latched by registers $31a, 31b$ at TBUs 19 via multiplexer MUX 33 and buffer $43a$, as shown at $m$ in Fig. 17. Since data transfer for each pixel is completed each time a source data $S_{D_0}$ is outputted, TBUs controller 25 sets control signal TH to L level to supply the L-level signal to sequencer 23 to permit the transfer of the next data as shown at $g$ in Fig. 17.

Since each pixel processor $X_{P_j}$ performs the above-described operation based on the number $j$ of the pixel processor $X_{P_j}$ of the destination, pixel data outputted from the pixel processor $X_{P_j}$ of the originate is outputted at an array conforming to the sequence of the increasing numbers $j$ of the pixel processor $X_{P_j}$ of the destination. Meanwhile, the number shown at $g$ in Fig. 17 corresponds to the above-mentioned number $j$.

The pixel data outputted over TBUs 19 in this manner are received by each pixel processor $X_{P_j}$ in the following manner.

Buffer 43b is also controlled by control signals DIR from TBUs controller 25 and enable signal $E_{N_5}$ so that, if the
control signal DIR is at H level and enable signal EN$_H$ is at L level, buffer 43b is rendered operative. TBus controller 25 counts, with the L-level of the control signal TS from sequencer 23 as a reference, the number of clocks corresponding to the data transfer rate on TBus 19 by a number of the pixel processor XP, of the associated pixel processor SP$_j$, and renders the buffer 43b operative at the point the counting is terminated, thereby latching in registers 34a, 34b of the reception data RD$_k$ timed to the outputting at the TBus 19 of the source data SD$_k$ which is executed after the aforementioned timings T0, T1, T3 ... Specifically, if the number of the pixel processor XP is 12, as mentioned above, TBus controller 25 sets the control signal DIR to H level at the time corresponding to the 12th time slot of the transfer data on the TBus 19, while setting the enable signal EN$_H$ to the L level to render buffer 43b operative, as shown at $m, g$ in Fig.17. Besides, TBus controller 25 alternately sets the enable signals EN$_H$, EN$_L$ supplied to registers 34a, 34b, to L level, as shown at $h, j$ in Fig.17, for alternately latching reception data RD$_k$ received over TBus 19.

The source data SD$_k$, alternately latched by registers 34a, 34b, is alternately selected by multiplexor MUX 37. The selected reception data RD$_k$ are color-converted, if need be, so as to be stored in memory $M_j$ via logical operation circuit 41, register 45c and buffer 42b. Specifically, multiplexor MUX 37 selects reception data RD$_k$' being an even number, from register 34a, and reception data RD$_k$' being an odd number, from register 34a, when control signal SRC from sequencer 23 is at the L-level and at the H level, respectively, as shown at $j$ in Fig.17, and transmits the selected reception data RD$_k$ to shift register 38. Shift register 28 shifts the reception data RD$_k$ by a predetermined number of bits to supply the shifted data via register 45a to color converter 39. Color converter 39 converts, based on the lower most bit (LSB) of the shifted pixel data, the reception data RD$_k$ into the background color and foreground color when the LSB is 0 or 1, respectively, and transmits the optionally color-converted reception data RD$_k$ via register 45b to logical operation circuit 41. With the BITBLT command not accompanied by the logical operation, logical operating circuit 41 transmits the optionally color-converted reception data RD$_k$ directly to buffer 42b via register 45c. Buffer 42b is rendered operative in timed relation to writing into memory $M_j$, as mentioned above, and transmits the optionally color-converted reception data RD$_k$ as write data WD$_k$ to memory $M_j$. Memory $M_j$ stores the reception data RD$_k$ in a manner equivalent to writing at the aforementioned timings T2, T4, T6, ...

The accessing timing of the memory $M_j$ and TBus 19 at the BITBLT command, accompanied by logical operation, is explained in detail.

Similarly to the BITBLT command not accompanied by the above-mentioned logical operation, accessing of the memory $M_k$ is started by the L-level of the request signal MREQ, as shown at $g$ in Fig.18, while reading and writing of pixel data in and from memory $M_k$ is controlled by read/write signal R/W, as shown at $b$ in Fig.18, so that, if the read/write signal R/W is at the H level, memory $M_k$ is in the read mode. Memory controller 24 sets the read/write signal R/W to H level, and transmits readout address indicating the pixel position of the source area 51 sequentially to memory $M_k$ so that, during the data transfer accompanied by logical operation, memory controller 24 effects reading of source data SD$_0$, SD$_1$, SD$_2$, SD$_3$, in a manner equivalent to the readout operation at the timings T0, T2, T5, T6, ... shown in Fig.9, while sequentially supplying the readout addresses indicating pixel positions of the destination area 52 as shown at $d$ in Fig.18, for sequentially reading destination data DD$_0$, DD$_1$, DD$_2$, ..., in a manner equivalent to the readout operation of the pixel data at the aforementioned timings T1, T4, T7, ... On the other hand, memory controller 24 sets the read/write signal R/W to the L-level, while sequentially transmitting write addresses indicating the pixel positions in the destination area 52 to memory $M_j$ for sequentially writing the write data WD$_0$, WD$_1$, WD$_2$, ... in a manner equivalent to the writing operation at the aforementioned timings T3, T6, T9, ... Memory controller 24 transmits to sequencer 23 the signal NEXT advising of the readout and write timings each time reading and writing are performed, as shown at $c$ in Fig.18.

On the other hand, buffers 42a, 42b of the main path circuit 21 are controlled in timed relation to the read/write signal R/W, so that the buffers 42a, 42b are rendered operative when the signal R/H is at the H and L levels, respectively. Registers 31a, 31b are alternately selected as to writing therein when the enable signals EN$_H$, EN$_L$ supplied from sequencer 23 are at the L level, as shown at $e$ and $f$ in Fig.18, for alternately latching source data SD$_k$ read out from memory $M_j$ via buffer 42a. That is, alternate latching of the source data SD$_0$, SD$_1$, SD$_2$, SD$_3$, ... corresponding to the latching operation at the aforementioned timings T0, T2, T5, T8, ... is carried out sequentially.

On the other hand, register 40 is allowed as to writing therein when the enable signal EN$_H$ from sequencer 23 is at the L level, for sequentially latching the destination data DD$_0$, read out from memory MN$_j$, as shown at $k$ in Fig.18. That is, alternate latching of the destination data DD$_0$, DD$_1$, DD$_2$, DD$_3$, ... corresponding to the latching operation at the aforementioned timings T1, T4, T7, ... is carried out sequentially.

The source data SD$_k$, thus latched alternately by registers 31a, 31b, are alternately selected by multiplexor MUX 33, so as to be outputted at TBus 19 via buffer 43a and input/output port IO$_2$. Specifically, multiplexor MUX 33 selects source data SD$_k$ from register 31a and source data SD$_k$ from register 31b when the control signal SSC from sequencer 23 is at the L and H levels, respectively, as shown at $g$ in Fig.18, and outputs the selected source data SD$_k$ to TBus 19 via buffer 43a controlled by TBus controller 25. TBus controller 25 counts, with the L-level of the control signal, controlling the start of data transfer supplied from sequencer 23, as a reference, the number of clocks corresponding to the data transfer rate on TBus 19, by the number
indicating the pixel processor \(XP_j\) of the destination, and renders buffer 43a operative on termination of counting, for effecting the outputting of the source data \(SD_j\) to TBus 19, which outputting is performed after the end of the writing of the source data \(SD_i\) in registers 31a, 31b at the aforementioned timings T0, T2, T5, T8 ... If the number \(j\) of the pixel processor \(XP_j\) is 12 and the number \(i\) of the pixel processor \(XP_i\) of the destination is 5, TBus controller 25 sets the control signal DIR and enable signal \(EN_{i}\) to the L-level, within a time interval the time corresponding to the fifth time slot of the transfer data on the TBus 19, as shown at \(m\) in Fig. 18, to render the buffer 43a operative to output source data \(SD_i\) alternately latched in registers 31a, 31b at TBus 19 via multiplexor MUX 33 and buffer 43a. TBus controller 25 transmits a control signal TH advising the completion of data transfer for each pixel as a low level signal to sequencer 23 each time source data \(SD_i\) is outputted, as shown at \(n\) in Fig. 18.

Each pixel processor \(XP_j\) performs the above-described operation based on the number \(j\) of the pixel processor \(XP_j\) of the destination so that the pixel data outputted from the pixel processor \(XP_j\) of the destination are arrayed in the increasing order of the number \(j\) of the pixel processor \(XP_j\) of the destination, so as to be outputted on the TBus 19 in this sequence, as shown at \(j\) in Fig. 18.

The pixel data outputted on TBus 19 in this manner is received by each pixel processor \(XP_j\) in the following manner.

Buffer 43b is also controlled by control signals DIR from TBus controller 25 and enable signal \(EN_{j}\) so that, if the control signal DIR is at H level and enable signal \(EN_{j}\) is at L level, buffer 43b is rendered operative. TBus controller 25 counts, with the L-level of the control signal \(TS\) from sequencer 23 as a reference, the number of clocks corresponding to the data transfer rate on TBus 19 by a number \(j\) of the pixel processor \(XP_j\) of the TBus controller, and renders the buffer 43b operative at the time point the counting is terminated, for latching to registers 34a, 34b of the reception data \(RD_j\) timed to the outputting to the TBus 19 of the source data \(SD_i\) which outputting is executed after the aforementioned timings T0, T1, T3 ... Specifically, if the number \(j\) of the pixel processor \(XP_j\) is 12, as mentioned above, TBus controller 25 sets the control signal DIR to H level within a time interval corresponding to the 12th time slot of the transfer data on the TBus 19, while setting the enable signal \(EN_{12}\) to the L level to render buffer 43b operative, as shown at \(p\) in Fig. 18. Besides, TBus controller 25 alternately sets the enable signals \(EN_{12}, EN_{5}\), supplied to registers 34a, 34b, to L level, as shown at \(q\) in Fig. 11, for alternately latching reception data \(RD_j\) received over TBus 19.

The reception data \(RD_{12}\) is alternately latched by registers 34a, 34b in this manner, are alternately selected by multiplexor MUX 37, which is controlled by control signal \(SRC\) shown at \(j\) in Fig. 18, so that the selected data is supplied to shift register 38. The reception data \(RD_{12}\) are optionally color-converted by shift register 38 and color-conversion circuit 39 so that the optionally color-converted reception data \(RD_{12}\) are supplied via register 45b to local operation circuit 41.

The logical operation circuit 41 performs a logical processing between the pixel data of the destination area latched by register 40 and the optionally color-converted reception data \(RD_{12}\) to generate new pixel data which are supplied as write data \(WD_{12}\) to memory \(M_i\) via register 45c and buffer 42b. That is, buffer 42b is rendered operative in timed relation to writing in the memory \(M_i\) as mentioned above, to transmit new pixel data from logical operation circuit 41 as write data \(WD_{12}\) to memory \(M_i\) for storage therein in a manner equivalent to writing at the aforementioned timings T3, T6, T9.

Meanwhile, if the pixel processor \(XP_j\) of the destination area, as found by the above formula (1), coincides with the pixel processor \(XP_j\) of the originate, it means transmission to itself, in which case pixel processor \(XP_j\) writes pixel data read out from memory \(N_{12}\) in its own memory \(M_j\) without transmission over TBus 19. Specifically, multiplexor MUX 36 is controlled to select pixel data from multiplexor MUX 35 and to reciprocate the pixel data read from memory \(M_j\).

In this manner, each pixel processor \(XP_j\) finds the number \(j\) of the pixel processor \(XP_j\) of the destination area, and simultaneously effects readout of the pixel data from memory \(M_j\) and transmission of the readout pixel data to TBus 19 in the increasing order of the number \(j\), while simultaneously effecting reception of pixel data from TBus 19 and writing of the pixel data in memory \(M_j\), so that, with the data transfer command, such as BITBLT command, high speed data transfer may be achieved even if the boundary of the source area 51 is not coincident with the boundary of the block \(B_{x,y}\). In other words, accessing the memory \(M_j\) and TBus 19 may proceed simultaneously so that data transfer may be achieved in substantially the same time as that for data transfer within its own memory \(M_j\) for which data transfer between the pixel processors \(XP_j\) is not necessary.

The readout address for each pixel processor \(XP_j\) to read out pixel data from memory \(M_j\) is explained.

The transfer sequence on the display screen of source data \(SD_{12}\) outputted on TBus 19 by each pixel processor \(XP_j\), at each data command such as BITBLT command is determined by the codes of the distances \(D_{Xd}, D_{Yd}\) from the source area 51 up to the destination area 52. Thus, in order that pixel data of the source area 51 are not rewritten by new pixel data before transfer, data transfer is started at a 4x4 pixel block containing the right lower corner pixel, referred to as a start corner, of the source area 51, if \(D_{Xd} \geq 0\) and \(D_{Yd} \geq 0\), as shown in Fig. 19. It is noted that the above pixel block is different from the above-mentioned block \(B_{x,y}\). Data transfer is performed from the start corner block sequentially to the left side block and then from the right corner block one line above towards the left side block.

If \(D_{Xd} \geq 0\) and \(D_{Yd} < 0\), data transfer is started from the right upper corner block of the source area 51 and proceeds from this start corner block sequentially towards the left side block and, after the end of the blocks of the line is reached, data transfer is performed sequentially from the right corner block one line below towards the left side block, as shown
in Fig. 20.

If \( D_{x_d} \leq 0 \) and \( D_{y_d} \geq 0 \), data transfer is started from the left lower corner block of the source area 51 and proceeds from this start corner block sequentially towards the right side block and, after the end of the blocks of the line is reached, data transfer is performed sequentially from the left corner block one line above towards the right side block, as shown in Fig. 21.

If \( D_{x_d} \leq 0 \) and \( D_{y_d} < 0 \), data transfer is started from the left upper corner block of the source area 51 and proceeds from this start corner block sequentially towards the right side block and, after the end of the blocks of the line is reached, data transfer is performed sequentially from the left corner block one line below towards the right side block, as shown in Fig. 22.

Specifically, each pixel processor \( XP_i \) finds, on the basis of the coordinate values \((x_c, y_c)\) of a starting point of the source area 51 and its size \( W, H \), the coordinate values of an end point \((x_e, y_e) = (x_c + W - 1, y_c + H - 1)\) at the diagonally opposite corner of the starting point, and generates a readout address of the memory \( M_i \) based on the sign of the distance \( D_{x_d}, D_{y_d} \) up to the destination area 52 and these coordinates.

If, as shown for example in Fig. 23, the number \( q \) of the pixel processors \( XP_i \) is 16, the coordinate values of the starting point \((x_c, y_c)\) are \((10, 10)\), and its size \( W, H \) are 21, 18, and if, when the coordinate values \((x_c, y_c)\) are \((30, 27)\), \( D_{x_d} \geq 0 \) and \( D_{y_d} \geq 0 \), each pixel processor \( XP_i \) divides the source area 51 into \( 4 \times 4 \) pixel blocks having the terminal point as the start corner and having the terminal point as the point of origin, as shown for example in Fig. 24. The block addresses for discriminating these blocks are indicated as \((X, Y)\) \((X, Y = 0, 1, 2, 3, \ldots)\). Therefore, these blocks differ from the blocks \( B_{x,y} \) bounded by the number \( j \) as discussed in connection with the prior art, as shown in Fig. 13. Meanwhile, the start corners for \( D_{x_d} \geq 0 \) and \( D_{x_d} < 0 \), \( D_{y_d} = 0 \) and \( D_{y_d} \geq 0 \), \( D_{x_d} < 0 \) and \( D_{y_d} < 0 \) are also shown and each start corner is necessarily at an inner corner of the source area 51.

Each pixel processor \( XP_i \) finds, at a block having a block address \((X, Y)\) of \((0, 0)\) containing the start corner, the coordinate values \((x_0, y_0)\) on the display screen of the pixels under its control by the formula shown in Table 1.

<table>
<thead>
<tr>
<th>( D_{x_d} )</th>
<th>( D_{y_d} )</th>
<th>( X_0 )</th>
<th>( Y_0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \geq 0 )</td>
<td>( &gt; 0 )</td>
<td>( x_e + (1 \wedge 3H) )</td>
<td>( y_e + (b&gt;2) )</td>
</tr>
<tr>
<td>( \geq 0 )</td>
<td>( \leq 0 )</td>
<td>( x_e + (1 \wedge 3H) \cdot 4 )</td>
<td>( y_e + (b&gt;2) - L )</td>
</tr>
<tr>
<td>( &lt; 0 )</td>
<td>( \geq 0 )</td>
<td>( x_e + (1 \wedge 3H) )</td>
<td>( y_e + (b&gt;2) )</td>
</tr>
<tr>
<td>( &lt; 0 )</td>
<td>( \leq 0 )</td>
<td>( x_e + (1 \wedge 3H) \cdot 4 )</td>
<td>( y_e + (b&gt;2) - L )</td>
</tr>
<tr>
<td>( \leq 0 )</td>
<td>( \geq 0 )</td>
<td>( x_e + (1 \wedge 3H) )</td>
<td>( y_e + (b&gt;2) )</td>
</tr>
<tr>
<td>( \leq 0 )</td>
<td>( \leq 0 )</td>
<td>( x_e + (1 \wedge 3H) \cdot 4 )</td>
<td>( y_e + (b&gt;2) + L )</td>
</tr>
</tbody>
</table>

while finding, using the coordinate values \((x_0, y_0)\) as the reference, the coordinate values \((x_m, y_n)\) \((m, n = 0, 1, 2, \ldots)\) of the pixels under its control by the formula shown in Table 2.

<table>
<thead>
<tr>
<th>( D_{x_d} )</th>
<th>( D_{y_d} )</th>
<th>( x_{m+1} )</th>
<th>( y_{n+1} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \geq 0 )</td>
<td>( \geq 0 )</td>
<td>( x_{m} + 4 )</td>
<td>( y_{n} - L )</td>
</tr>
<tr>
<td>( \geq 0 )</td>
<td>( &lt; 0 )</td>
<td>( x_{m} + 4 )</td>
<td>( y_{n} + L )</td>
</tr>
<tr>
<td>( &lt; 0 )</td>
<td>( \geq 0 )</td>
<td>( x_{m} + 4 )</td>
<td>( y_{n} - L )</td>
</tr>
<tr>
<td>( &lt; 0 )</td>
<td>( &lt; 0 )</td>
<td>( x_{m} + 4 )</td>
<td>( y_{n} + L )</td>
</tr>
</tbody>
</table>

and, using the coordinate values \((x_0, y_0)\) as readout addresses, read out the pixel data of the respective pixels in the sequence of the block addresses \((X, Y)\) of \((0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (5, 0), (0, 1), (1, 1), \ldots\), simultaneously for each of the blocks.

Meanwhile, symbols \( >>2 \) and \( \wedge \) indicate 2 bits shift to right and logical product, respectively, \( i, X_{eH}, x_{eH}, y_{eH}, y_{eH} \), \( L, 3H, 4H \) are represented by the hexadecimal notation, \( X_{eH}, x_{eH}, y_{eH}, y_{eH} \) are values obtained by the formulas 2 to 5.
the values of L are 1H (1), 2H (2), 4H (4) and 8H (8) in association with the number \( n \) of the pixel processor \( XP_i \) of 4, 8, 16 and 32, respectively, and selection between the formulas at the upper and lower parts of Table 1 is governed by the conditions shown by the following Tables 3 and 4.

<table>
<thead>
<tr>
<th>( DX_d )</th>
<th>Conditions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>≥0</td>
<td>((i \land 3H) \leq (x_{sl} \land 3H))</td>
<td>Upper part of Table 1</td>
</tr>
<tr>
<td></td>
<td>((i \land 3H) &gt; (x_{sl} \land 3H))</td>
<td>Lower part of Table 1</td>
</tr>
<tr>
<td>&lt;0</td>
<td>((i \land 3H) \leq (x_{sl} \land 3H))</td>
<td>Upper part of Table 1</td>
</tr>
<tr>
<td></td>
<td>((i \land 3H) \geq (x_{sl} \land 3H))</td>
<td>Lower part of Table 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( DY_d )</th>
<th>Conditions</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>≥0</td>
<td>((i&gt;&gt;2) \leq (y_{sl} \land Ln))</td>
<td>Upper part of Table 1</td>
</tr>
<tr>
<td></td>
<td>((i&gt;&gt;2) &gt; (y_{sl} \land Ln))</td>
<td>Lower part of Table 1</td>
</tr>
<tr>
<td>&lt;0</td>
<td>((i&gt;&gt;2) \leq (y_{sl} \land Ln))</td>
<td>Upper part of Table 1</td>
</tr>
<tr>
<td></td>
<td>((i&gt;&gt;2) &lt; (y_{sl} \land Ln))</td>
<td>Lower part of Table 1</td>
</tr>
</tbody>
</table>

Meanwhile, in these formulas 2 to 5 and Tables 3 and 4, symbols \( >>2 \), \( \land \) and \( \sim \) denote shift 2 bits left, logical product and negative logic i, \( x_{sl}, x_{sh}, y_{sl}, y_{sh}, Ln \), 3H denote hexadecimal numbers and \( x_{sl}, x_{sh} \) are lower 2 bits of \( x_e \), \( x_e \) obtained based on formulas 6 and 7

\[
x_{sl} = x_e - x_{sh} \tag{6}
\]

\[
x_{sl} = x_e - x_{sh} \tag{7}
\]

\( y_{sl}, y_{sh} \) are obtained from the formulas 8 and 9

\[
y_{sl} = x_e - x_{sh} \tag{8}
\]

\[
y_{sl} = x_e - y_{sh} \tag{9}
\]

and are lower 0th bit, lower 1st bit, lower 2nd bits and lower 3rd bit of \( y_e \) and \( y_e \) in association with the numbers \( n \) of the pixel processor \( XP_i \) of 4, 8, 16 and 32, respectively. The value of Ln are OH (0), 1H (1), 3H (3) and 7H (7), in association with the numbers \( n \) of 4, 8, 16 and 32 of the pixel processor \( XP_i \), respectively.
Specifically, if assumed that the number \( n \) of the pixel processors \( XP_i \) is 16, and the coordinate values \((x_n, y_n)\) of a starting point is \((10, 10)\) that is \((AH, AH)\) and its size is 21, 16, each pixel processor \( XP_i \) finds the coordinate values \((x_{N}, y_{N})\) as \((30, 27)\) that is \((1EH, 1BH)\). With \( D_{N} \geq 0 \) and \( D_{N} \leq 0 \), the pixel processors \( XP_{3}, XP_{4}, XP_{11}, XP_{12} \) find the \( x \) coordinate in the block having the block address \((X, Y)\) of \((0, 0)\) of a coordinate \((x_{N}, y_{N})\) on a display screen, employed as the readout address for memory \( M_{N} \) as \(1BH (27)\) from the formula \( y_{N} = (i \land 3H) \cdot 4H \) shown in Table 1 because the formula \( i \land 3H > (x_{N} \land 3H) \) is shown in Table 3 is satisfied. Similarly, pixel processors \( XP_{5}, XP_{6}, XP_{7}, XP_{13} \) find the \( x \) coordinate as \(1CH (28)\) from the formula \( x_{N} = (i \land 3H) \cdot 4H \) shown in Table 1 because the formula \((i \land 3H) \cdot 4H \leq (x_{N} \land 3H) \) is shown in Table 3 is satisfied. Pixel processors \( XP_{1}, XP_{2}, XP_{8}, XP_{10} \) find the \( x \) coordinate as \(1DH (29)\) from the formula \( x_{N} + (i \land 3H) \) shown in Table 1 because the formula \((i \land 3H) \leq (x_{N} \land 3H) \) is shown in Table 3 is satisfied.

On the other hand, pixel processors \( XP_{3}, XP_{4}, XP_{11}, XP_{12} \) find the \( y \) coordinate of the coordinates \((x_{N}, y_{N})\) as \(18H (24)\) by the formula \( y_{N} = (i \land 3H) \cdot 4H \) shown in Table 1 because the formula \((i \land 3H) \leq (y_{N} \land L) \) shown in Table 4 is satisfied. Pixel processors \( XP_{5}, XP_{6}, XP_{7}, XP_{8} \) find the \( y \) coordinate of the coordinates \((x_{N}, y_{N})\) as \(19H (25)\) by the formula \( y_{N} + (i \land 3H) \) shown in Table 1 because the formula \((i \land 3H) \leq (y_{N} \land L) \) shown in Table 4 is satisfied. Finally, pixel processors \( XP_{1}, XP_{2}, XP_{11}, XP_{12} \) find the \( y \) coordinate of the coordinates \((x_{N}, y_{N})\) as \(1AH (26)\) by the formula \( y_{N} + (i \land 3H) \) shown in Table 1 because the formula \((i \land 3H) \leq (y_{N} \land L) \) shown in Table 4 is satisfied.

Each pixel processor \( XP_i \) finds the coordinate values \((x_{N}, y_{N})\) of other blocks by the formula shown in Table 2, with the coordinate values \((x_{N}, y_{N})\) shown in Table 4, as the reference. That is, each pixel processor \( XP_i \) finds the coordinate values of the respective blocks \((x_{N}, y_{N})\) by subtracting 4 from the coordinate value each time the value of the \( X \) coordinate of the block address \((X, Y)\) is incremented by 1 and by subtracting \( L \) (herein \( L = 4 \)) each time the coordinate value of the \( Y \) coordinate is incremented by 1 and reads out pixel data from memory \( M_{N} \) with these coordinates \((x_{N}, y_{N})\) as readout addresses.

On the other hand, storage of the received pixel data in memory \( M_{N} \) is carried out based on the coordinates of the pixels of the destination area 52 on the display screen. That is, the coordinate values of each pixel of the destination area 52 is found by the formula similar to that for the readout address and, using these coordinate addresses as the write addresses, write the received pixel data. Meanwhile, pixel data not included in the source area 51 of the blocks having the block addresses \((X, Y)\) shown in Fig. 24 of \((5, 0), (5, 1), \ldots, (5, 4), (0, 4), (1, 4) \ldots (4, 4)\) are transferred between the pixel processors \( XP_i \) without being written in memory \( M_{N} \).

It will be seen from above that, in the graphics engine according to the present invention, each pixel processor \( XP_i \) finds the number \( j \) of the pixel processor \( XP_i \) of the pixel object that reads in the read pixel data in the sequence of the increasing number \( j \) while it receives pixel data from TBus 19 at the same time that it reads the pixel data in memory \( M_{N} \) so that, if the boundary of the source area 5 is not coincident in the data transfer command, such as BITBLT command, with the boundary of the block \( B_{x, y} \) data transfer can be performed speedily. In other words, accessing the memory \( M_{N} \) and accessing TBus 19 can proceed simultaneously so that data transfer may be achieved within the same time as that for data transfer within the own memory \( M_{N} \) is not necessary.

In the above-described embodiment, each pixel processor \( XP_i \) of the origin finds the number \( j \) of the pixel processor \( XP_i \) that transfers pixel data between the pixel processors \( XP_i \) in the data transfer command such as BITBLT command in the order of the increasing number \( j \) while pixel processor \( XP_i \) of the receiver receives the data in the order of the increasing number \( j \). However, it is also possible for the pixel processor \( XP_i \) of the origin to transmit pixel data in the order of the increasing number \( j \) and for the pixel processor \( XP_i \) of the receiver to find the number \( j \) of the pixel processor \( XP_i \) to receive the data in the order of the increasing number \( j \).

That is, each pixel processor \( XP_i \) reads pixel data from memory \( M_{N} \) at the same time that it transmits the pixel data to TBus 19 in the order of the increasing number \( j \) while finding the number \( j \) of the pixel processor \( XP_i \) of the origin and receives pixel data from TBus 19 at the same time that it writes the received data in memory \( M_{N} \) in the order of the increasing number \( j \), so that high speed data transfer may be achieved even if the boundary of the source area 51 is not coincident with the boundary of the block \( B_{x, y} \). In other words, accessing the memory \( M_{N} \) and accessing TBus 19 may proceed simultaneously and data transfer may be made within substantially the same time as that for data transfer within own memory \( M_{N} \) for which data transfer between pixel processors \( XP_i \) is not required.

It is seen from above that, in accordance with the present invention, the number of controlling means of the receiver of the pixel data is found in synchronism and pixel data read out via first input/output port from memory based on this number is supplied via second input/output port and bus to control means of the receiver while pixel data supplied from control means of the origin via bus and second input/output port are written in memory means via first input/output port to effect pixel data transfer between memory means, so that pixel data having an optional size on the display screen may be transferred at an elevated speed to any desired position on the display screen even although the
memory interleave system is adopted.
On the other hand, pixel data read via first input/output port from memory means are supplied via second input/output port and bus to control means of the receiver, while the number of the receiver is found in timed relation thereto, and pixel data supplied from the control means of the originate via ons and second input/output port is written in memory means via first input/output port to effect data transfer between memory means, so that pixel data having an optional size on the display screen may be transferred at an elevated speed to any desired position on the display screen even although the memory interleave system is adopted.

Claims

1. A picture memory apparatus comprising

an n number of memory means (M) each having a storage capacity equal to 1/n of a storage capacity corresponding to the resolution of a display screen and each being adapted for storing a pixel data,
an n number of control means (XP) each having first and second input/output ports (IO₁, IO₂) for inputting and outputting the pixel data and each adapted for controlling readout and writing of pixel data in and from said memory means (M) via said first input/output port (IO₁), and
bus connection means for commonly interconnecting said second input/output ports (IO₂) of said n number of said control means (XP) for transfer of pixel data between the control means,

characterized in

that said n number of control means (XP) perform in synchronism a control operation of finding, from all of said control means, the number of control means to receive said pixel data said control means (XP) also performing a control operation of supplying pixel data read out from said memory means (M) via said first input/output port (IO₁) to controlling means to pixel data via second input/output port (IO₂) and bus connection means, and of causing said control means to receive said pixel data, to write pixel data supplied thereto via said bus connection means and said second input/output port (IO₂) in said memory means (M) via said first input/output port (IO₁).

2. A picture memory apparatus comprising

an n number of memory means (M) each having a storage capacity equal to 1/n of a storage capacity corresponding to the resolution of a display screen and each being adapted for storing a pixel data,
an n number of control means (XP) each having first and second input/output ports (IO₁, IO₂) for inputting and outputting the pixel data and each adapted for controlling readout and writing of pixel data in and from said memory means (M) via said first input/output port (IO₁), and
bus connection means for commonly interconnecting said second input/output ports (IO₂) of said n number of said control means (XP) for transfer of pixel data between the control means,

characterized in

that said n number of control means (XP) perform a controlling operation for supplying pixel data read out from said memory means (M) via said first input/output port (IO₁) and bus connection means to control means to receive said pixel data via said second input/output port (IO₂) and said bus connection means, said control means (XP) also performing in synchronism a controlling operation of finding the number of the control means from which the pixel data originated and of causing control means to receive said pixel data to write pixel data supplied thereto from control means (XP) from which pixel data originated via said bus connection means and said second input/output port (IO₂) in said memory means (M) via said first input/output port (IO₁).

3. A graphic engine apparatus comprising a picture memory apparatus according to claim 1, furthermore comprising

a memory (12) for storage of commands for picture processing,
a setup processor (13) for sequentially reading out the commands stored in said memory for calculating parameters necessary for generation of pixel data,
a rendering processor (14) for supervising data flow for graphic processing,
a pixel data generating circuit (15) for generating pixel data responsive to parameters and commands for generating pixel data from said setup processor (13), and
a video processing unit (17) for converting pixel data read out from said pixel data generating circuit (15) readout and writing of pixel data in and from said memory means (M) via said first input/output port (IO₁), and
EP 0 563 855 B1

bus connection means for commonly interconnecting said second input/output ports (IO₂) of said n number of said control means (XP),

wherein

that said n number of control means (XP) perform in synchronism a control operation of finding, from all of said control means, the number of control means to receive said pixel data said control means (XP) also performing a control operation of supplying pixel data read out from said memory means (M) via said first input/output port (IO₁) to controlling means to pixel data via second input/output port (IO₂) and bus connection means, and of causing said control means to receive said pixel data to write pixel data supplied thereto via said bus connection means and said second input/output port (IO₂) in said memory means (M) via said first input/output port (IO₁).

4. A graphic engine apparatus comprising a picture memory apparatus according to claim 2, furthermore comprising

a memory (12) for storage of commands for picture processing,
a setup processor (13) for sequentially reading out the commands stored in said memory for calculating parameters necessary for generation of pixel data,
a rendering processor (14) for supervising data flow for graphic processing,
a pixel data generating circuit (15) for generating pixel data responsive to parameters and commands for generating pixel data from said setup processor (13), and
a video processing unit (17) for converting pixel data read out from said pixel data generating circuit (15), an n number of control means (XP) each having first and second input/output ports (IO₁, IO₂) for inputting and outputting the pixel data and each adapted for controlling readout and writing of pixel data in and from said memory means (M) via said first input/output port (IO₁), and
bus connection means for commonly interconnecting said second input/output ports (IO₂) of said n number of said control means (XP) for transfer of pixel data between the control means (XP),

wherein

that said n number of control means (XP) perform a controlling operation for supplying pixel data read out from said memory means (M) via first input/output port (IO₁) and bus connection means to control means to receive said pixel data via said second input/output port (IO₂) and said bus connection means, said control means (XP) also performing in synchronism a controlling operation of finding the number of the control means from which the pixel data originated and of causing control means to receive said pixel data to write pixel data supplied thereto from control means (XP) from which pixel data originated via said bus connection means and said second input/output port (IO₂) in said memory means (M) via said first input/output port (IO₁).

Patentansprüche

1. Bildspeichervorrichtung, welche umfaßt

   eine Anzahl von n Speichereinrichtungen (M), von denen jede eine Speicherkapazität besitzt, die gleich 1/n einer Speicherkapazität ist, welche der Auflösung eines Wiedergabebildschirms entspricht, und von denen jede zur Speicherung von Pixelwerte ausgelegt ist,

   eine Anzahl von n Steuerungseinrichtungen (XP), von denen jede erste und zweite Eingangs-/Ausgangsanschlüsse (IO₁, IO₂) zum Eingehen und Ausgeben der Pixelwerte besitzt, und von denen jede zur Steuerung des Auslesevervorgangs und des Schreibvorgangs von Pixelwerte in und aus den Speichereinrichtungen (M) über den ersten Eingangs-/Ausgangsanschluß (IO₁) ausgelegt ist, und

   eine Busverbindungseinrichtung zur gemeinsamen Verbindung der zweiten Eingangs-/Ausgangsansschlüsse (IO₂) der Anzahl von n Steuerungseinrichtungen (XP) zur Übertragung der Pixelwerte zwischen den Steuerungseinrichtungen,

dadurch gekennzeichnet,

die Anzahl von n Steuerungseinrichtungen (XP) synchron eine Steuerungstätigkeit zum Herausfinden der Nummer der Steuerungseinrichtungen für den Empfang der Pixelwerte aus allen Steuerungseinrichtungen durchführen, wobei die Steuerungseinrichtungen (XP) außerdem eine Steuerungstätigkeit zur Zuführung der über den ersten Eingangs-/Ausgangsanschluß (IO₁) aus den Speichereinrichtungen (M) ausgelesenen Pixelwerte über den
zweiten Eingangs-/Ausgangsanschluß (IO₂) und der Busverbindungseinrichtung an Steuerungseinrichtungen für die Pixel daten durchführen, und um die Steuerungseinrichtungen zu ver anlassen, die Pixel daten zu empfangen, um die ihnen über die Busverbindungseinrichtung und den zweiten Eingangs-/Aus gangsanschluß (IO₂) zugeführten Pixeldaten über den ersten Eingangs-/Ausgangsanschluß (IO₁) in die Speiche reinrichtung (M) zu schreiben.

2. Bildspeichervorrichtung, welche umfaßt
eine Anzahl von n Speichereinrichtungen (M), von denen jede eine Speicherkapazität besitzt, die gleich 1/n einer Speicherkapazität ist, welche der Auflösung eines Wiedergabebildschirms entspricht, und von denen jede zur Speicherung von Pixeldaten ausgelegt ist,
eine Anzahl von n Steuerungseinrichtungen (XP), von denen jede erste und zweite Eingangs-/Ausgangsanschlüsse (IO₁, IO₂) zum Eingeben und Ausgeben der Pixel daten besitzt, und von denen jede zur Steuerung des Auslesew- und des Schreibvorgangs von Pixeldaten in und aus den Speichereinrichtungen (M) über den ersten Eingangs-/Ausgangsanschluß (IO₁) ausgelegt ist, und
eine Busverbindungseinrichtung zur gemeinsamen Verbindung der zweiten Eingangs-/Ausgangsanschlüsse (IO₂) der Anzahl von n Steuerungseinrichtungen (XP) zur Übertragung der Pixel daten zwischen den Steuerungseinrichtungen,
dadurch gekennzeichnet,
die Anzahl n der Steuerungseinrichtungen (XP) synchron eine Steuerungstätigkeit zum Zuführen der über den ersten Eingabe/Ausgabeanschluß (IO₁) und der Busverbindungseinrichtung aus der Speichereinrichtung (M) ausgelesenen Pixel daten über den zweiten Eingangs-/Ausgangsanschluß (IO₂) und die Busverbindungseinrichtung an die Steuerungseinrichtungen für den Empfang der Pixel daten durchführen, wobei die Steuerungseinrichtungen (XP) außerdem eine Steuerungstätigkeit zum Herausfinden der Nummer der Steuerungseinrichtungen durchführen, von welcher die Pixel daten herühren, und um die Steuerungseinrichtungen zu ver anlassen, die Pixel daten zu empfangen, um die ihnen von den Steuerungseinrichtungen (XP), von welchen die Pixel daten herühren, über die Busverbindungseinrichtung und den zweiten Eingangs-/Ausgangsanschluß (IO₂) zugeführten Pixeldaten über den ersten Eingangs-/Ausgangsanschluß (IO₁) in die Speichereinrichtung (M) zu schreiben.

3. Grafsche Anzeigevorrichtung, welche eine Bildspeichervorrichtung nach Anspruch 1 umfaßt, welche weiterhin umfaßt
einen Speicher (12) zur Speicherung von Befehlen für eine Bildverarbeitung,
einen Setup-Prozessor (13) zum aufeinanderfolgenden Auslesen der in dem Speicher gespeicherten Befehle zur Berechnung von Parametern, welche zur Erzeugung von Pixel daten erforderlich sind,
ein Übergabeprozessor (14) zur Überwachung des Datenflusses für die grafische Verarbeitung,
eine Erzeugungsschaltung (15) für Pixel daten zur Erzeugung von Pixel daten als Reaktion auf Parameter und Befehle von dem Setup-Prozessor (13) zur Erzeugung von Pixel daten und
eine Video-Verarbeitungseinheit (17) zur Umsetzung von aus der Pixel daten-Erzeugungsschaltung (15) ausgelesenen Pixel daten und Schreiben von Pixel daten in und aus den Speichereinrichtungen (M) über den ersten Eingangs-/Ausgangsanschluß (IO₁), und
eine Busverbindungseinrichtung zur gemeinsamen Verbindung der zweiten Eingangs-/Ausgangsanschlüsse (IO₂) der Anzahl von n Steuerungseinrichtungen (XP),
dadurch gekennzeichnet,
die Anzahl von n Steuerungseinrichtungen (XP) synchron eine Steuerungstätigkeit zum Herausfinden der Nummer der Steuerungseinrichtungen für den Empfang der Pixel daten aus allen Steuerungseinrichtungen durchführen, wobei die Steuerungseinrichtungen (XP) außerdem eine Steuerungstätigkeit zur Zuführung der über den ersten Eingangs-/Ausgangsanschluß (IO₁) aus den Speichereinrichtungen (M) ausgelesenen Pixel daten über den zweiten Eingangs-/Ausgangsanschluß (IO₂) und der Busverbindungseinrichtung an Steuerungseinrichtungen für die Pixel daten durchführen, und um die Steuerungseinrichtungen zu ver anlassen, die Pixel daten zu empfangen,
EP 0 563 855 B1

um die ihnen über die Busverbindungseinrichtung und den zweiten Eingangs-/Ausgangsan schluss (IO₂) zugeführten Pixeldaten über den ersten Eingangs-/Ausgangsanschluss (IO₁) in die Speichereinrichtung (M) zu schreiben.

4. Grafische Anzeigevorrichtung, welche eine Bildspeichervorrichtung nach Anspruch 2 umfaßt, welche weiterhin umfaßt
   einen Speicher (12) zur Speicherung von Befehlen für eine Bildverarbeitung,
   einen Setup-Prozessor (13) zum aufeinanderfolgenden Auslesen der in dem Speicher gespeicherten Befehle zur Berechnung von Parametern, welche zur Erzeugung von Pixeldaten erforderlich sind,
   ein Übergabeprozessor (14) zur Überwachung des Datenflusses für die grafische Verarbeitung,
   eine Erzeugungsschaltung (15) für Pixeldaten zur Erzeugung von Pixeldaten als Reaktion auf Parameter und Befehle von dem Setup-Prozessor (13) zur Erzeugung von Pixeldaten und
   eine Video-Verarbeitungseinheit (17) zur Umsetzung von aus der Pixeldaten-Erzeugungsschaltung (15) ausgelesenen Pixeldaten,
   eine Anzahl von n Steuerungseinrichtungen (XP), von denen jedes erste und zweite Eingangs-/Ausgangsanschlüsse (IO₁, IO₂) zum Eingeben und Ausgeben der Pixeldaten besitzt, und von denen jede zur Steuerung des Auslese- und des Schreibvorgangs von Pixeldaten in und aus den Speichereinrichtungen (M) über den ersten Eingangs-/Ausgangsanschlüße (IO₁) ausgelagert ist, und
   eine Busverbindungseinrichtung zur gemeinsamen Verbindung der zweiten Eingangs-/Ausgangsanschlüsse (IO₂) der Anzahl von n Steuerungseinrichtungen (XP) zur Übertragung der Pixeldaten zwischen den Steuerungseinrichtungen (XP),

bei welcher
   die Anzahl n der Steuerungseinrichtungen (XP) synchron eine Steuerungstätigkeit zum Zuführen der über den ersten Eingabe/Ausgabeanschluss (IO₁) und der Busverbindungseinrichtung aus der Speichereinrichtung (M) ausgelesenen Pixeldaten über den zweiten Eingangs-/Ausgangsanschluss (IO₂) und die Busverbindungseinrichtung an die Steuerungseinrichtungen für den Empfang der Pixeldaten durchführen, wobei die Steuerungseinrichtungen (XP) außerdem eine Steuerungstätigkeit zum Herausfinden der Nummer der Steuerungseinrichtungen durchführen, von welcher die Pixeldaten herrühren, und um die Steuerungseinrichtungen zu veranlassen, die Pixeldaten zu empfangen, um die ihnen von den Steuerungseinrichtungen (XP), von welchen die Pixeldaten herrühren, über die Busverbindungseinrichtung und den zweiten Eingangs-/Ausgangsanschluss (IO₂) zugeführten Pixeldaten über den ersten Eingangs-/Ausgangsanschlüße (IO₁) in die Speichereinrichtung (M) zu schreiben.

Revendications

1. Un dispositif de mémoire d'images comprenant

   un nombre n de moyens de mémoire (M) ayant chacun une capacité d'enregistrement égale à la fraction 1/n d'une capacité d'enregistrement correspondant à la résolution d'un écran de visualisation, et chacun d'eux étant adapté pour enregistrer des données de pixels;
   un nombre n de moyens de commande (XP), ayant chacun des premier et second ports d'entrée/sortie (IO₁, IO₂) pour recevoir et émettre les données de pixels, et chacun d'eux étant adapté pour commander la lecture et l'écriture de données de pixels dans les moyens de mémoire (M), par l'intermédiaire du premier port d'entrée/sortie (IO₁), et des moyens de connexion à bus pour interconnecter en commun les seconds ports d'entrée/sortie (IO₂) des moyens de commande (XP) au nombre de n, pour le transfert de données de pixels entre les moyens de commande,

 caractérisé en ce que
   les moyens de commande (XP) au nombre de n accomplissent en synchronisme une opération de commande qui consiste à trouver, parmi tous les moyens de commande, le numéro de moyens de commande devant recevoir
les données de pixels, ces moyens de commande (XP) accomplissent également une opération de commande qui consiste à fournir des données de pixels lues dans les moyens de mémoire (M), par l’intermédiaire du premier port d’entrée/sortie (IO₁), à des moyens de commande pour des données de pixels, par l’intermédiaire du port d’entrée/sortie (IO₂) et des moyens de connexion à bus, et à faire en sorte que les moyens de commande devant recevoir les données de pixels écrivent dans les moyens de mémoire (M), par l’intermédiaire du premier port d’entrée/sortie (IO₁), les données qui leur sont fournies par l’intermédiaire des moyens de connexion à bus et du second port d’entrée/sortie (IO₂).

2. Un dispositif de mémoire d’images comprenant

un nombre n de moyens de mémoire (M) ayant chacun une capacité d’enregistrement égale à la fraction 1/n d’une capacité d’enregistrement correspondant à la résolution d’un écran de visualisation, et chacun d’eux étant adapté pour enregistrer des données de pixels,

un nombre n de moyens de commande (XP) ayant chacun des premier et second ports d’entrée/sortie (IO₁, IO₂) pour recevoir et émettre les données de pixels, et chacun d’eux étant adapté pour commander la lecture et l’écriture de données de pixels dans les moyens de mémoire (M), par l’intermédiaire du premier port d’entrée/sortie (IO₁), et des moyens de connexion à bus pour interconnecter en commun les seconds ports d’entrée/sortie (IO₂) des moyens de commande (XP) au nombre de n, pour le transfert de données de pixels entre les moyens de commande,

caractérisé en ce que

les moyens de commande (XP), au nombre de n, accomplissent une opération de commande pour fournir des données de pixels lues dans les moyens de mémoire (M), par l’intermédiaire du premier port d’entrée/sortie (IO₁) et des moyens de connexion à bus, à des moyens de commande devant recevoir les données de pixels, par l’intermédiaire du second port d’entrée/sortie (IO₂) et des moyens de connexion à bus, les moyens de commande (XP) accomplissant également en synchronisme une opération de commande qui consiste à trouver le numéro des moyens de commande desquels proviennent les données de pixels, et à faire en sorte que les moyens de commande devant recevoir les données de pixels écrivent dans les moyens de mémoire (M), par l’intermédiaire du premier port d’entrée/sortie (IO₁), les données de pixels qui leur sont fournies par des moyens de commande (XP) à partir desquels les données de pixels sont arrivées, par l’intermédiaire des moyens de connexion à bus et du second port d’entrée/sortie (IO₂).

3. Un moteur graphique comprenant un dispositif de mémoire d’images selon la revendication 1, comprenant en outre

une mémoire (12) pour l’enregistrement d’ordres pour le traitement d’images,

un processeur de préparation (13) pour lire séquentiellement les ordres qui sont enregistrés dans la mémoire, pour calculer des paramètres nécessaires pour la génération de données de pixels,

un processeur de rendu (14) pour superviser le flux de données pour le traitement graphique,

un circuit de génération de données de pixels (15) pour générer des données de pixels sous la dépendance de paramètres et d’ordres pour la génération de données de pixels, provenant du processeur de préparation (13),

une unité de traitement vidéo (17) pour convertir des données vidéo qui sont lues dans le circuit de génération de données de pixels (15), et pour lire et écrire des données de pixels dans les moyens de mémoire (M) par l’intermédiaire du premier port d’entrée/sortie (IO₁), et des moyens de connexion à bus pour interconnecter en commun les seconds ports d’entrée/sortie (IO₂) des moyens de commande (XP) au nombre de n,

dans lequel

les moyens de commande (XP) au nombre de n accomplissent en synchronisme une opération de commande qui consiste à trouver, parmi tous les moyens de commande, le numéro de moyens de commande devant recevoir les données de pixels, ces moyens de commande (XP) accomplissant également une opération de commande qui consiste à fournir des données de pixels lues dans les moyens de mémoire (M), par l’intermédiaire du premier port d’entrée/sortie (IO₁), à des moyens de commande pour des données de pixels, par l’intermédiaire du second port d’entrée/sortie (IO₂) et des moyens de connexion à bus, et à faire en sorte que les moyens de commande devant recevoir les données de pixels écrivent dans les moyens de mémoire (M), par l’intermédiaire du premier port d’entrée/sortie (IO₁), les données qui leur sont fournies par l’intermédiaire des moyens de connexion à bus et du second port d’entrée/sortie (IO₂).
4. Un moteur graphique comprenant un dispositif de mémoire d'images selon la revendication 2, comprenant en outre

une mémoire (12) pour l’enregistrement d’ordres pour le traitement d’images,
un processeur de préparation (13) pour lire séquentiellement les ordres qui sont enregistrés dans la mémoire,
pour calculer des paramètres nécessaires pour la génération de données de pixels,
un processeur de rendu (14) pour superviser le flux de données pour le traitement graphique,
un circuit de génération de données de pixels (15) pour générer des données de pixels sous la dépendance de paramètres et d’ordres pour la génération de données de pixels, provenant du processeur de préparation (13), et
une unité de traitement vidéo (17) pour convertir les données de pixels qui sont lues dans le circuit de génération de données de pixels (15),
un nombre n de moyens de commande (XP) ayant chacun des premier et second ports d’entrée/sortie (IO₁, IO₂) pour recevoir et émettre les données de pixels, et chacun d’eux étant adapté pour commander la lecture et l’écriture de données de pixels dans les moyens de mémoire (M) par l’intermédiaire du premier port d’entrée/sortie (IO₁), et
les moyens de connexion à bus pour interconnecter en commun les seconds ports d’entrée/sortie (IO₂) des moyens de commande (XP) au nombre de n, pour le transfert de données de pixels entre les moyens de commande (XP),

dans lequel
les moyens de commande (XP) au nombre de n accomplissent une opération de commande pour fournir des données de pixels qui sont lues dans les moyens de mémoire (M), par l’intermédiaire du premier port d’entrée/sortie (IO₁) et des moyens de connexion à bus, à des moyens de commande devant recevoir les données de pixels par l’intermédiaire du second port d’entrée/sortie (IO₂) et des moyens de connexion à bus, les moyens de commande (XP) accomplissant également, en synchronisme, une opération de commande qui consiste à trouver le numéro des moyens de commande à partir desquels les données de pixels sont arrivées, et à faire en sorte que les moyens de commande devant recevoir les données de pixels écrivent dans les moyens de mémoire (M), par l’intermédiaire du premier port d’entrée/sortie (IO₁), des données qui leur sont fournies par les moyens de commande (XP) à partir desquels les données de pixels sont arrivées, par l’intermédiaire des moyens de connexion à bus et du second port d’entrée/sortie (IO₂).
(ACCESS TBus)

START

T0
REGISTER 31b-(SD0) → TBus

T1
MEMORY MI-(DD0) → REGISTER 40

T2
MEMORY MI-(SD1) → REGISTER 31a

T3
PERFORM LOGICAL OPERATION OF RD0 AND DD0 AND STORE IN MEMORY MI(INDICATED REGISTER 34b-(WD0) → MEMORY MI)

T4
MEMORY MI-(DD1) → REGISTER 40

T5
MEMORY MI-(SD2) → REGISTER 31b

T6
REGISTER 34a-(WD1) → MEMORY MI

T7
MEMORY MI-(DD2) → REGISTER 40

T8
MEMORY MI-(SD3) → REGISTER 31a

T9
REGISTER 34b-(WD2) → MEMORY MI

END

FIG.16