

DESCRIPTION

Memory Accessing Method and Data Processing Apparatus

Technical Field

This invention relates to a memory accessing method for having access to a frame buffer by pixel interleaving and a data processing apparatus for performing pre-set data processing by accessing a frame buffer by the above memory accessing method. The memory accessing method and the data processing apparatus are usable for a video game machine or a graphic computer system as video equipment employing a computer.

Background Art

In a picture generating apparatus for generating data of a picture outputted to and displayed on a display device, that is output display picture data, for a television receiver, a monitor receiver or a cathode ray tube (CRT) display device, used in a video game device or a graphics computer system, a dedicated drawing device is provided between a central processing unit (CPU) and a frame buffer for enabling high-speed processing.

That is, in the above picture generating apparatus, the CPU when generating a picture performs geometry processing, such as coordinate transformation or clipping or light source calculations, instead of directly accessing the frame buffer. The CPU then defines a three-dimensional model as a combination of basis unit figures, such as triangular or quadrangular figures, formulates a drawing command for drawing a three-dimensional picture, and sends

the drawing command to the drawing device.

For example, if, in the above-described picture generating apparatus, a three-dimensional object is represented, the object is resolved into plural polygons, and a drawing command associated with each polygon is transmitted from a CPU to a drawing device. The drawing device interprets the drawing command sent from the CPU and executes rendering processing of computing colors and Z-values of all pixels making up the polygon from the Z-values for writing pixel data in a frame buffer for drawing a picture in a frame buffer.

The Z-value represents color data and depth of all pixels making up a polygon, that is, the Z-value is the information representing the distance along the depth from the viewing point.

Meanwhile, the above-described picture generating device has the function of executing pixel interleaving of writing plural pixel data at a time with plural addresses of a frame buffer as accessing unit, and is configured for accessing the frame buffer by this function.

However, with the pixel interleaving performed by the above picture generating device, plural simultaneously accessible addresses are fixed from one accessing unit to another.

Therefore, if a picture represented by the plural simultaneously accessible addresses and a picture drawn on the frame buffer, that is the shape on the memory of data actually desired to be accessed, are not coincident with each other, the

number of times of accessing the frame buffer is undesirably increased.

Moreover, if the number of times of accessing the frame buffer is increased, the data processing carried out on the picture generating device is worsened.

In view of the above-depicted status of the art, the present invention has the following objects.

Specifically, it is an object of the present invention to provide a memory accessing method in which simultaneously accessible addresses are changed over for enabling the memory to be accessed with a minimum number of times of accessing operations for raising the memory accessing efficiency.

It is another object of the present invention to provide a data processing apparatus in which simultaneously accessible addresses are changed over for enabling the memory to be accessed with a minimum number of times of accessing operations for raising the data processing efficiency.

Disclosure of the Invention

In one aspect, the present invention provides a memory accessing method for simultaneously accessing a plurality of addresses of a memory wherein addresses to be accessed simultaneously are determined depending on the shape on the memory of data desired to be accessed.

In the memory accessing method according to the present invention, the above data may be picture data.

In the memory accessing method according to the present invention, the addresses to be accessed may be determined depending on the aspect ratio of the shape.

In the memory accessing method according to the present invention, the addresses to be accessed may be selected from a plurality of patterns.

In another aspect, the present invention provides a data processing apparatus for performing pre-set data processing by simultaneously accessing a plurality of addresses on a memory including accessing means for simultaneously accessing a plurality of addresses on the memory, and control means for determining the addresses simultaneously accessed by the accessing means depending on the shape of input data.

In the data processing apparatus of the present invention, data is picture data.

In the data processing apparatus of the present invention, the control means determines the simultaneously accessed addresses depending on the aspect ratio of the shape of the input data.

In the data processing apparatus of the present invention, the control means selects the accessed addresses from plural patterns.

Brief Description of The Drawings

Fig.1 is a block diagram showing the structure of a video game machine employing a data processing device according to the present invention.

Fig.2 is a block diagram showing the structure of a picture

processing device of the video game machine and a frame buffer.

Fig.3 is a block diagram showing the structure of a second bus switcher of the picture processing device.

Fig.4 illustrates accessing the inside of a first polygon drawn on a memory bank of the frame buffer.

Fig.5 illustrates an interleaving pattern to be accessed when accessing the inside of a first polygon.

Fig.6 illustrates masking processing in case of accessing the inside of the first polygon on the address basis.

Fig.7 illustrates an accessing address obtained by the masking processing.

Fig.8 illustrates accessing the inside of the shape of a second polygon drawn on a memory bank of the frame buffer with a (4×4) interleaving pattern.

Fig.9 illustrates an interleaving pattern to be accessed when having access to the inside of the second polygon with the (4×4) interleaving pattern.

Fig.10 illustrates masking processing when having access to the inside of the shape of the second polygon on the basis of the (4×4) interleaving pattern.

Fig.11 illustrates accessing the inside of the shape of the second polygon with a (8×2) interleaving pattern.

Fig.12 illustrates an interleaving pattern to be accessed when having access to the inside of of the shape of the second polygon with the (8×2) interleaving pattern.

Fig.13 illustrates masking processing when having access to the inside of the shape of the second polygon on the address basis within the (8×2) interleaving pattern.

Fig.14 illustrates accessing the inside of the shape of the second polygon with a (16×1) interleaving pattern.

Fig.15 illustrates an interleaving pattern to be accessed when accessing the inside of the shape of the second polygon with a (16×1) interleaving pattern.

Fig.16 illustrates masking processing when having access to the inside of the shape of the second polygon on the address basis within the (16×1) interleaving pattern.

Fig.17 illustrates processing for computing the aspect ratio of the shape of the polygon drawn on a memory bank of the above frame buffer.

Fig.18 is a pattern diagram showing five interleaving patterns each having 16 addresses.

Fig.19 is a plan view showing a video game machine according to the present invention.

Fig.20 is a back side view of the video game machine.

Fig.21 is a side view showing the video game machine.

Fig.22 is a plan view of a CD-ROM loaded on the video game machine.

Best Mode for Carrying Out the Invention

Referring to the drawings, preferred embodiments of the present invention will be explained in detail.

The memory accessing method according to the present invention is carried out by a video game machine configured as shown in Fig.1. The video game machine is an apparatus employing a data processing device according to the present invention.

The video game machine executes a game in accordance with instructions by the user by reading and executing a game program stored in an auxiliary memory device such as an optical disc, and has two sorts of buses, namely a main bus 1 and a sub-bus 2. The main bus 1 and the sub-bus 2 are interconnected via a bus controller 16.

To the main bus 1 are connected a central processing unit (CPU) 11, made up of a micro-processor, a main memory 12, made up of a random access memory (RAM), a main dynamic memory access controller or main DMAC 13, a MPEG decoder 14 and a picture processing unit or graphic processing unit (GPU) 15.

To the sub-bus 2 are connected a subsidiary central processing unit or sub-CPU 21, made up of a micro-processor, a subsidiary memory 22, made up of a random access memory (RAM), a subsidiary dynamic memory accessing controller or sub DMAC 23, a read-only memory (ROM) 24, having stored therein a program, such as an operating system, a sound processing unit (SPU) 25, a communication controller or asynchronous transmission mode (ATM) 26, an auxiliary storage device 27, an input device 28 and a CD-ROM driver 30.

First, on the side of the main bus 1, the bus controller 16 is a device on the main bus 1 for switching between the main bus 1 and

the sub-bus 2, and is opened in an initial state.

The main CPU 11 is a device on the main bus 1 operating by a program on the main memory 12. Since the bus controller 16 is open on start-up, the Main CPU 11 reads in a boot program from the ROM 24 on the sub-bus 2 in order to load an application program and necessary data from the auxiliary storage device 27 on the main memory 12 or on devices on the sub-bus 2.

On the main CPU 11 is loaded a geometry transfer engine (GTE) 17 for performing processing such as coordinate transformation. This GTE 17 has a parallel computing mechanism for carrying out plural computing processing operations in parallel and is responsive to a request for computing processing from the CPU 11 in order to carry out fast processing operations, such as coordinate transformation, light source calculations, matrix or vector operations. Based on the results of computing processing operations by the GTE 17, the main CPU 11 defines a three-dimensional model as a combination of basic unit figures, such as triangles or quadrangles, formulates a drawing command associated with each polygon for drawing a three-dimensional picture and packetizes the drawing command in order to route the resulting command packet to the GPU 15.

The main DMAC 13 is a device on the main bus 1 for performing control such as DMA transfer on the devices on the main bus 1. If the bus controller 16 is open, the main DMAC 13 also controls the devices on the sub-bus 2.

The CPU 15 is a device on the main bus 1 functioning as a rendering processor. This GPU 15 interprets the drawing command sent from the main CPU 11 or main DMAC 13 as command packet and performs rendering processing of calculating Z-values and colors of all pixels making up a polygon for writing pixel data in the frame buffer 18.

The GPU 15 will be explained subsequently in detail.

The MDEC 14 is an I/O connection device operable in parallel with the main CPU 11 and is a device functioning as a picture expansion engine. The MDEC 14 decodes picture data compressed and encoded by orthogonal transform, such as discrete cosine transform.

On the sub-bus 2, the sub-CPU 21 is a device on the sub-bus 2 operating in accordance with a program on the sub-memory 22.

The sub-DMAC 23 is a device on the sub-bus 2 for performing control such as DMAC transfer for devices on the sub-bus 2. This sub-DMAC 23 can acquire bus rights only when the bus controller 16 is closed.

The SPU 25 is a device on the sub-bus 2 functioning as a sound processor. This SPU 25 is responsive to a sound command sent as a command packet from the sub-CPU 21 or sub-DMAC 23 as a command packet to read out sound source data from the sound memory 29 to output the read-out data.

The ATM 26 is a communication device on the sub-bus 2.

The auxiliary storage device 27 is a data input/output device

on the sub-bus 2 and is made up of a non-volatile memory, such as a flash memory. This auxiliary storage device 27 temporarily holds on memory data such as game progress or scores.

The input device 28 is an input device from other equipments, such as a control pad on the sub-bus 2, man-machine interface, picture input or speech input.

The CD-ROM driver 30 is an input device on the sub-bus 2 and reproduces an application program and necessary data from the CD-ROM.

That is, in the above-described video game machine, the geometry processing system, executing geometry processing, such as coordinate transformation, clipping or light source calculations, formulating a drawing command for defining a three-dimensional model as combination of basic unit figures (polygons) such as triangles or quadrangles for drawing a three-dimensional picture, and sending out the drawing command associated with each polygon as command packet to the main bus 1, is made up of the main CPU 11 and the GTE 17 on the main bus 1. Also, in the video game machine, the rendering processing system for formulating pixel data of each polygon based on the drawing command from the geometry processing system for writing a figure in the frame buffer 18 by way of rendering processing for drawing a figure in the frame buffer 18 is constituted by the GPU 15.

The above-mentioned GPU 15 is explained in detail.

Referring to Fig.2, the GPU 15 includes a packet engine 31

connected to the main bus 1 shown in Fig.1 and performs rendering processing of writing pixel data of each polygon in the frame buffer 18 by the pre-processor 32 and the drawing engine 33 in accordance with the drawing command sent as a command packet from the main CPU 11 or main DMAC 13 shown in Fig.1 over the main bus 1 to the packet engine 31, reading out pixel data of the picture drawn in the frame buffer 18 and furnishing the read-out pixel data via CRT controller 34 as video signals to a television receiver or a monitor receiver, not shown.

The packet engine 31 develops the command packet sent from the main CPU 11 or main DMAC 13 shown in Fig.1 over the main bus 1 on a register, not shown.

The pre-processor 32 generates polygon data in accordance with the drawing command sent to the packet engine 31 as command packet and performs pre-set pre-processing, such as polygon division as later explained, on the polygon data, while generating various data such as apex point coordinate information for the respective polygons required by the drawing engine 33, address information such as texture or MIP map texture, or control information, such as pixel interleaving.

The drawing engine 33 includes N polygon engines 33A1, 33A2, ..., 33AN, connected to the pre-processor 32, N texture engines 33B1, 33B2, ..., 33BN, connected to the polygon engines 33A1, 33A2, ..., 33AN, a first bus switcher 33C, connected to the texture engines 33B1, 33B2, ..., 33BN, M pixel engines 33D1, 33D2, ...,

33DM, connected to the first bus switcher 33C, a second bus switcher 33E; connected to the pixel engines 33D1, 33D2, ..., 33DM, a texture cache 33F connected to the second bus switcher 33E, and a CLUT cache 33G connected to the texture cache 33F.

In the drawing engine 33, the N polygon engines 33A1, 33A2, ..., 33AN sequentially generate polygons in accordance with the drawing command on the basis of the polygon data pre-processed by the pre-processor 32 for performing parallel shading processing from one polygon to another.

The N texture engines 33B1, 33B2, ..., 33BN perform texture mapping or MIP map processing in parallel, based on the texture data supplied from the texture cache 33F via color lookup table (CLUT) cache 33G, on the polygons generated by the polygon engines 33A1, 33A2, ..., 33AN.

The pre-processor 32 previously furnishes the address information, such as texture or MIP map texture, bonded to the polygons processed by the N texture engines 33B1, 33B2, ..., 33BN, to the texture cache 33F. Based on the above address information, the necessary texture data is transferred from the texture area on the frame buffer 18. To the CLUT cache 33G CLUT data to be referred to during texture drawing is transferred from the CLUT area on the frame buffer 18.

The polygon data, processed with texture mapping or MIP map processing by the N texture engines 33B1, 33B2, ..., 33BN, are transferred via first bus switcher 33C to M pixel engines 33D1,

33D2, ..., 33DM.

The M pixel engines 33D1, 33D2, ..., 33DM perform various picture processing operations, such as Z-buffer processing or anti-aliasing processing, in parallel, for generating M pixel data.

The M pixel data, generated by the M pixel engines 33D1, 33D2, ..., 33DM, are written in the frame buffer 18 via second bus switcher 33E.

The second bus switcher 33E is fed with the pixel interleaving control information from the pre-processor 32. The second bus switcher 33E as a function of performing pixel interleaving processing of selecting L pixel data from the M pixel data generated by the M pixel engines 33D1, 33D2, ..., 33DM in accordance with the above control information for writing M pixel data using the M storage locations conforming to the shape of the polygon drawn on the frame buffer 18 as accessing unit.

The pixel interleaving processing performed by the second bus switcher 33E will be explained subsequently.

The drawing engine 33 generates and writes all pixel data of each polygon in the frame buffer 18, based on the polygon data pre-processed by the pre-processor 32, for drawing a picture defined as the combination of the polygons by the drawing command on the frame buffer 18.

Thus the pixel data of the picture drawn on the frame buffer 18 is read out by the second bus switcher 33E so as to be furnished via CRTC 34 as video signals to a television receiver or to a

monitor receiver, not shown.

The pixel interleaving processing performed by the above-mentioned second bus switcher 33E is now specifically explained.

Referring to Fig.3, the second bus switcher 33E includes a control circuit 101 fed with an output of the pre-processor 32 shown in Fig.2, a selector 102 fed with an output of the control circuit 101 and plural multiplexer/demultiplexers (MUX/DMUX) 103a, 103b, 103c, 103d, ... each fed with an output of the selector 102.

The MUX/DMUX 103a, 103b, 103c, 103d, ... are connected to the frame buffer 18 and to the drawing engine 33 shown in Fig.2.

The frame buffer 18 is made up of plural memory banks [1], [2], ..., [X], ..., [L], as shown in Fig.2. Each of the memory banks [1], [2], ..., [X], ..., [L] is made up of a rectangular area represented by 16 addresses (interleaving patterns), so that the 16 addresses can be accessed simultaneously.

Therefore, the memory bank [X], for example, of the frame buffer 18, has 16 input/output ports P_0 to P_{15} for accessing the addresses A_0 to A_{15} . Of the MUX/DMUX 103a, 103b, 103c, 103d, ... , the four MUX/DMUX 103a, 103b, 103c, 103d are each connected to 16 input/output ports P_0 to P_{15} .

Also, the four MUX/DMUX 103a, 103b, 103c, 103d are connected in association with four pixel engines $33D_{x1}$, $33D_{x2}$, $33D_{x3}$ and $33D_{x4}$ of the drawing engine 33.

Since the memory banks other than the memory bank [X] are

configured similarly to the above-mentioned memory bank [X], the detailed description therefor is not made for simplicity. The accessing operation performed by the second bus switcher 33E on the other memory banks is similar to that performed by the second bus switcher 33E on the memory bank [X] as later explained. Therefore, in the following description, only the accessing operation performed by the second bus switcher 33E on the memory bank [X] is explained.

First, a series of operations by the second bus switcher 33E is explained.

If, for example, the shape of a polygon drawn on the memory bank [X] is a triangle T_{ABC} (the shape of the first polygon) as shown in Fig.4, the control circuit 101 is first fed from the pre-processor 32 with the pixel interleaving control information.

Based on the pixel interleaving control information from the pre-processor 32, the control circuit 101 changes over an interleaving pattern, used for accessing the inside of the triangle T_{ABC} , to, for example, a (4×4) interleaving pattern.

The method for changing over the interleaving pattern in the control circuit 101 will be later explained in detail.

Of plural interleaving patterns formed on the memory bank [X], such interleaving pattern to be accessed, that is, such interleaving pattern as permits the inside of the triangle T_{ABC} to be accessed in its entirety, is detected by the control circuit 101, with the aid of the (4×4) interleaving pattern.

Thus, in the triangle T_{ABC} , if each interleaving pattern on the memory bank [X] is indicated by P (pattern index in the x-direction and pattern index in the y-direction), a sum total of 20 interleaving patterns specified by

$$\begin{aligned}
 P(x,y) = & P(3,1), P(4,1), \\
 & P(1,2), P(2,2), \\
 & P(3,2), P(4,2), \\
 & P(1,3), P(2,3), \\
 & P(3,3), P(4,3), \\
 & P(5,3), P(2,4), \\
 & P(3,4), P(4,4), \\
 & P(5,4), P(3,5), \\
 & P(4,5), P(5,5), \\
 & P(4,6), P(5,6)
 \end{aligned}$$

are detected, as shown in Fig.5.

The control circuit 101 routes the pattern information specifying the 20 interleaving patterns, detected as described above, to the selector 102 on the interleaving pattern basis. When performing memory accessing on the address basis, the control circuit 101 routes the mask information corresponding to the shape of the triangle T_{ABC} to the selector 102.

Based on the pattern information, supplied from the control circuit 101 on the interleaving pattern basis, the selector 102 designates the address corresponding to the (4×4) interleaving pattern P to be accessed for the MUX/DMUX 103a, 103b, 103c and

103d.

If fed with the mask information from the control circuit 101, the selector 102 designates for MUX/DMUX 103a to 103d the accessing addresses obtained as a result of masking performed in the (4×4) interleaving patterns P, on the basis of the masking information, as shown in Fig.6. Thus, of the addresses A_0 to A_{15} in the interleaving pattern specified by $P(4,1)$ shown in Fig.6, the addresses to be accessed, obtained as the result of masking, are A_4 , A_5 , A_6 , A_8 , A_9 , A_{10} , A_{13} , A_{14} and A_{15} , shown shaded in Fig.7.

The MUX/DMUX 103a, 103b, 103c and 103d access the addressees A_0 to A_{15} on the memory bank [X] designated by the selector 102.

The pixel engines $33D_{x1}$, $33D_{x2}$, $33D_{x3}$ and $33D_{x4}$ output pixel data to the MUX/DMUX 103a, 103b, 103c and 103d, respectively, as explained previously.

Thus the MUX/DMUX 103a accesses the address designated by the selector 102 to write pixel data from the pixel engine X_a , via one of the input/output ports P_0 to P_{15} corresponding to the address designated by the selector 102, in an area on the memory bank [X] designated by the above address.

The MUX/DMUX 103a accesses the address designated by the selector 102 to read out data written in an area designated by the address on the memory bank [X] via one of the input/output ports P_0 to P_{15} corresponding to the above address. The MUX/DMUX 103a performs pre-set processing on the data read out from the memory bank [X].

Since the operation of the MUX/DMUX 103b to 103d is similar to the above-described operation of the MUX/DMUX 103a, detailed description therefor is omitted for clarity.

The method for changing over the interleaving patterns in the above-described control circuit 101 is now specifically explained.

First, the number of times of accessing the inside of a transversely elongated triangle T_{DEF} (shape of a second polygon) as the shape of a polygon drawn on the memory bank [X] shown in Fig.8 with a (4×4) interleaving pattern P is explained.

In this case, the interleaving patterns to be accessed are:

$$\begin{aligned}
 P(x,y) = & P(1,1), P(2,1), \\
 & P(3,1), P(4,1), \\
 & P(5,1), P(0,2), \\
 & P(1,2), P(2,2), \\
 & P(3,2), P(4,2), \\
 & P(5,2), P(6,2), \\
 & P(7,2), P(8,2), \\
 & P(7,3), P(8,3), \\
 & P(9,3),
 \end{aligned}$$

totaling at 17, as shown in Fig.9.

That is, for accessing the inside of the triangle T_{DEF} with the (4×4) interleaving pattern, the number of times of accessing for accessing the inside of the triangle T_{DEF} in its entirety is 17.

In case of accessing on the address basis, only the required memory addresses can be accessed by carrying out the masking in the

(4×4) interleaving pattern P , as in case of accessing the above-mentioned triangle T_{ABC} , as shown in Fig.10.

Then, if the inside of the triangle T_{DEF} is accessed with an (8×2) interleaving pattern, as shown in Fig.11, the interleaving patterns to be accessed are

$$\begin{aligned}
 P_1(x,y) = & P_1(1,2), P_1(2,2), \\
 & P_1(0,3), P_1(1,3), \\
 & P_1(2,3), P_1(0,4), \\
 & P_1(1,4), P_1(2,4), \\
 & P_1(3,4), P_1(1,5), \\
 & P_1(2,5), P_1(3,5), \\
 & P_1(4,5), P_1(3,6), \\
 & P_1(4,6),
 \end{aligned}$$

totaling at 15, as shown in Fig.12.

That is, if the inside of the triangle T_{DEF} is accessed with the (8×2) interleaving pattern, the number of times of accessing required for accessing the entire inside of the triangle T_{DEF} is 15.

In case of accessing on the address basis, masking is performed within the (8×2) interleaving pattern P_1 as in case of accessing the triangle T_{ABC} as described above, as shown in Fig.13, for accessing only the needed memory address.

Then, if the inside of the triangle T_{DEF} is accessed with a (16×1) interleaving pattern, as shown in Fig.14, the interleaving patterns to be accessed are

$$\begin{aligned}
P_2(x,y) = & P_2(0,5), P_2(1,5), \\
& P_2(0,6), P_2(1,6), \\
& P_2(0,7), P_2(1,7), \\
& P_2(0,8), P_2(1,8), \\
& P_2(0,9), P_2(1,9), \\
& P_2(0,10), P_2(1,10), \\
& P_2(2,10), P_2(1,11), \\
& P_2(2,11), P_2(1,12), \\
& P_2(2,12), P_2(2,13)'
\end{aligned}$$

totaling at 18, as shown in Fig.15.

That is, if the inside of the triangle T_{DEF} is accessed with the (16×1) interleaving pattern, the number of times of accessing required for accessing the entire inside of the triangle T_{DEF} is 18.

In case of accessing on the address basis, masking is performed within the (8×2) interleaving pattern P_2 as in case of accessing the triangle T_{ABC} as described above, as shown in Fig.16, for accessing only the needed memory address.

As described above, the number of times of accessing the inside of the triangle T_{DEF} with the (4×4) interleaving pattern P is 17, while that of accessing the inside of the triangle T_{DEF} with the (8×2) interleaving pattern P_1 is 15 and that of accessing the inside of the triangle T_{DEF} with the (16×1) interleaving pattern P_2 is 18. Thus, the number of times of accessing the inside of the triangle T_{DEF} with the (8×2) interleaving pattern P_1 represents the

minimum number of times of accessing. Thus it is seen that a proper interleaving pattern for the triangle T_{DEF} is the (8×2) interleaving pattern P_1 .

Thus, for switching the interleaving pattern used for accessing the memory bank [X] to a proper interleaving pattern in meeting with the shape of the polygon to be accessed, the control circuit 101 executes the following processing operations.

If, for example, the shape of a polygon drawn on the memory bank [X] is a triangle T_{HIJ} , the control information for pixel interleaving is supplied from the pre-processor 32 to the control circuit 101, as described above. For example, the control information for pixel interleaving is such information as xy coordinates H (X_h, Y_h), I (X_i, Y_i) or J (X_j, Y_j) of three apex points of the triangle T_{HIJ} .

The control circuit 101 then finds, using the control information for pixel interleaving from the pre-processor 32, the aspect ratio R of the triangle T_{HIJ} , by calculations:

$$R = dy/dx$$

$$= (MAXx - MINx)/(MAXy - MINy)$$

where MAXx and MINx are maximum and minimum values in the X-direction, respectively, and MAXy and MINy are maximum and minimum values in the Y-direction, respectively, as shown in Fig.17.

In the triangle T_{HIJ} ,

$$MAXx = X_j$$

$$MINx = X_i$$

$$\text{MAXy} = \text{Yh}$$

$$\text{MINy} = \text{Yi}.$$

Depending on the aspect ratio R , thus found, the control circuit 101 selects one of five interleaving patterns Pa to Pe of (1×16) , (2×8) , (4×4) , (8×2) and (16×1) , as shown in Fig.18, and switches the interleaving pattern used in accessing the inside of the triangle T_{HIJ} to the selected interleaving pattern.

The control circuit 101 has a table for the aspect ratio R and interleaving patterns (Table 1). In this table are pre-set proper interleaving patterns associated with various values of the aspect ratio R , that is such an interleaving pattern as minimizes the number of times of accessing. Thus the control circuit 101 selects, using the above table, a proper interleaving pattern associated with the aspect ratio R found as described above.

TABLE 1

aspect ratio R	interleaving pattern
~ 0.1	$\text{Pa} (16 \times 1)$
$0.1 \sim 0.5$	$\text{Pb} (8 \times 2)$
$0.5 \sim 2.0$	$\text{Pc} (4 \times 4)$
$2.0 \sim 8.0$	$\text{Pd} (2 \times 8)$
$8.0 \sim$	$\text{Pe} (1 \times 16)$

In the second bus switcher 33E, a proper one of five interleaving patterns Pa to Pe shown in Fig.18 is selected responsive to the shape of the polygon to be drawn on the memory

bank [X], which is then accessed with the selected interleaving pattern, so that the polygon can be drawn on the memory bank [X] with the minimum number of accessing operations. Thus it is possible for the second bus switcher 33E to effect memory accessing efficiently.

The GPU 15 accesses the frame buffer 18 by the second bus switcher 33E aimed at raising the memory accessing efficiency, for performing various data processing operations, as described above, for realizing efficient data processing operations.

A video game machine embodying the present invention is configured as shown for example in a plan view of Fig.19, a front view of Fig.20 and in a side view of Fig.21.

That is, the present video game machine 201 is basically made up of a main body member 202 and an operating device 217 connected to the main body member 202 via a cable 227, as shown in Fig.19. At a mid portion on the upper surface of the main body member 202 is mounted a disc loading unit 203 within which is loaded a CD-ROM 251 shown in Fig.22. On the left side of the disc loading unit 203 are mounted a power source switch 205 operated for turning on or off the device and a reset switch 204 operated for transiently resetting the game. On the right side of the disc loading unit 203 is mounted a disc actuating switch 206 operated for mounting or dismounting the CD-ROM 251 with respect to the disc loading unit 203.

On the front side of the main body member 202 are mounted

connecting portions 207A, 207B, as shown in Fig.20. These connecting portions 207A, 207B are provided with a connection terminal 226 provided on the foremost part of the cable 227 derived from the operating device 217 and a connection terminal inserting portion 212 for connecting a recording device 228, such as a memory card, and a recording/inserting portion 208. That is, two each of the operating devices 217 and recording devices 228 can be connected to the main body member 202.

The front view of Fig.20 shows the state in which the connection terminal portion 226 and the recording device 228 are loaded on the right side connecting portion 207B with none of the connecting terminal portion 226 nor the recording device 228 being loaded on the left side connecting portion 207A. The recording inserting portion 208 for loading the recording device 228 thereon is provided with the shutter 209. When loading the recording device 228 on the main body member 202, the shutter 209 is thrust by the foremost part of the recording device 228 to effect loading.

The connecting terminal portion 226 has a knurled gripping part 231A for anti-slip effect. Similarly, the recording device 228 has a knurled gripping part 242A for anti-slip effect. Referring to a side view of Fig.21, the connecting terminal portion 226 has a length L substantially equal to a length of the recording device 228.

The operating device 27 has supports 220, 221 that can be

gripped with left and right hands. The foremost parts of the supports 220, 221 are provided with operating parts 218, 219, respectively. The operating parts 224, 225 are operated with index fingers of both hands, while the operating parts 218, 219 are operated with thumbs of both hands.

Between the operating parts 218 and 219 are provided a select switch 222 that is actuated for selection during the game and a start switch 223 operated for starting the game.

With the present video game machine 201, the CD-ROM 251 loaded on the disc loading unit 203 is reproduced by the CD-ROM driver 30. The operating device 217 is equivalent to the above-mentioned input device 28, while the recording device 228 is equivalent to the auxiliary storage device 27.

According to the present invention, as described above, the addresses for accessing plural addresses on the memory are switched depending on the shape on the memory of data desired to be accessed, for thereby decreasing the number of times of memory accessing operations even if the shape represented by plural simultaneously accessible address is not coincident with the shape on the memory of data desired to be accessed. Thus, the memory can be accessed with a smaller number of accessing operations, thus raising the memory accessing efficiency.

Also, according to the present invention, if the above data is picture data, the number of times of accessing the memory can be decreased even in case wherein the shape represented by plural

simultaneously accessible addresses is not coincident with the picture drawn on a picture memory.

Also, according to the memory accessing method of the present invention, the addresses to be accessed can be switched depending on the aspect ratio of the above shape for selecting a proper address for accessing in meeting with the data desired to be accesses. The result is that the memory can be accessed with the minimum number of times of accessing operations.



THIS SHEET HAS BEEN
REPLACED

27

Claims

1. A memory accessing method for simultaneously accessing a plurality of addresses of a memory characterized in that addresses to be accessed simultaneously are determined depending on the shape on the memory of data desired to be accessed.
2. The memory accessing method as claimed in claim 1 characterized in that the data is picture data.
3. The memory accessing method as claimed in claim 1 characterized in that the addresses to be accessed are determined depending on the aspect ratio of the shape.
4. The memory accessing method as claimed in claim 3 characterized in that the addresses to be accessed are selected from a plurality of patterns.
5. A data processing apparatus for performing pre-set data processing by simultaneously accessing a plurality of addresses on a memory comprising:

 accessing means for simultaneously accessing a plurality of addresses on the memory; and

 control means for determining the addresses simultaneously accessed by said accessing means depending on the shape of input data.
6. The memory accessing apparatus as claimed in claim 5 characterized in that the data is picture data.
7. The memory accessing apparatus as claimed in claim 5 characterized in that the control means determines the

This Sheet Has Been Replaced





THIS SHEET HAS BEEN
REPLACED

28

simultaneously accessed addresses depending on the aspect ratio of the shape of the input data.

8. The memory accessing apparatus as claimed in claim 7 characterized in that the control means selects the accessed addresses from plural patterns.

This Sheet Has Been Replaced

