

(51) International Patent Classification:
G06T 1/20 (2006.01)(21) International Application Number:
PCT/JP2015/001808(22) International Filing Date:
30 March 2015 (30.03.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **RENESAS ELECTRONICS CORPORATION** [JP/JP]; 2-24, Toyosu 3-chome, Koutou-ku, Tokyo, 1350061 (JP).(72) Inventor: **LIESKE, Hanno**; c/o Renesas Electronics Corporation, 2-24, Toyosu 3-chome, Koutou-ku, Tokyo, 1350061 (JP).(74) Agent: **IEIRI, Takeshi**; HIBIKI IP Law Firm, Asahi Bldg. 5th Floor, 3-33-8, Tsuruya-cho, Kanagawa-ku, Yokohama-shi, Kanagawa, 2210835 (JP).

AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

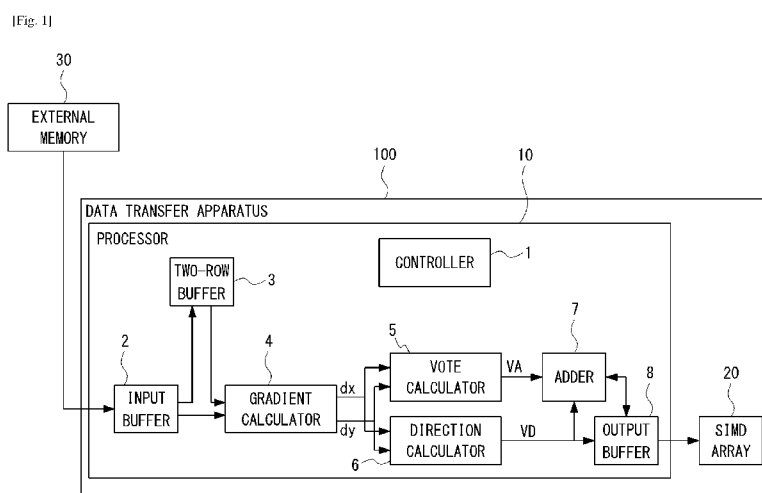
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

(54) Title: DATA TRANSFER APPARATUS AND MICROCOMPUTER



(57) **Abstract:** A two-row buffer (3) stores first and second rows. An input buffer (2) stores a third row. A gradient calculator (4) calculates first and second gradient values. A vote calculator (5) calculates a vote amount value. A direction calculator (6) calculates a vote direction value. An output buffer (8) stores accumulated vote amount values. An adder (7) adds the vote amount value to the received accumulated vote amount value and replaces the accumulated vote amount value in the output buffer (8) with the added value. The first gradient value is a difference between values of two pixels in the first and third row. The second gradient value is a difference between values of two pixels in the second row. The four pixels are immediately adjacent to a target pixel in the second row. The output buffer (8) outputs all of the accumulated vote amount values to an outside processor.

Description

Title of Invention: DATA TRANSFER APPARATUS AND MICROCOMPUTER

Technical Field

[0001] The present invention relates to a data transfer apparatus and a microcomputer.

Background Art

[0002] In the area of object detection, Histogram of Oriented Gradients (HOG) features are widely used as descriptors. In this algorithm, picture gradient orientation occurrences are counted in defined portions.

[0003] Therefore, cell histograms are generated where each pixel of a cell adds a weighted vote to an orientation-based histogram channel. The contribution of a pixel can be distinguished by using 1D centred filter in horizontal and vertical direction. Then the arctangent of the outputs, which are gradients in horizontal and vertical direction, specify the orientation while some function of the gradient magnitude is used as weighted vote.

[0004] Integral Histograms are used for fast construction of the HOG feature vectors. The advantage here is that after initial Integral Histogram generation, the bin values for each ROI (region of interest) area size can be accessed in constant time, so also the HOG feature vector can be calculated in constant time independent from the ROI size.

[0005] PTL1 and NPTL1 are showing two methods for detecting humans in images. Both are utilizing as one part of the processing chain the HOG algorithm. Differences between both HOG implementations are lying in e.g. the usage of fixed sized blocks (NPTL1) against variable sized blocks (PTL1) or usage of L2 normalization (NPTL1) against L1 normalization (PTL1).

[0006] The HOG algorithm can be implemented by utilizing the Integral Histogram as shown in NPTL2, which can significantly speed up the HOG vector calculation because of the constant processing time which becomes independent from the ROI size. NPTL2 shows for each pixel the required steps of the Integral Histogram algorithm. Assuming, the data is pixel-wise loaded from memory and the result is pixel-wise stored to memory, the generation of the Integral Histogram takes however a significant time.

Citation List

Patent Literature

[0007] PTL 1: US Patent Publication No. 2007/0237387

Non Patent Literature

[0008] NPTL 1: P. N. Dalal and B. Triggs, "Histograms of oriented gradients for human

detection", Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

NPTL 2: F. Porikli, "Integral histogram: A fast way to extract histograms in Cartesian spaces", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages I: 829-836, 2005.

Summary of Invention

Technical Problem

- [0009] PTL1 and NPTL1 are describing a whole chain of processing steps for human detection in images. The HOG algorithm is here only one part, which is described from an algorithmic view, where hardware details are not given. NPTL2 is pixel-wise describing the single tasks of the Integral Histogram algorithm. Also here, hardware details of an implementation are not given.
- [0010] Therefore, achieving a data transfer apparatus to specifically generate the Integral Histogram using data transferred from outside is required.

Solution to Problem

- [0011] An aspect of any one of embodiments is a data transfer apparatus, including: a two-row buffer that stores a first row consisted of consecutive pixels of the image and a second row consisted of consecutive pixels of the image, the first row being immediately adjacent to the second row; an input buffer that stores a third row which is consisted of consecutive pixels of the image and transferred from outside, the third row being immediately adjacent to the second row; a gradient calculator that calculates first and second gradient values; a vote calculator that calculates a vote amount value based on a sum of a square of the first gradient value and a square of the second gradient value; a direction calculator that calculates a vote direction value based on a ratio of the first and second gradient values; an output buffer that stores accumulated vote amount values, and outputs the accumulated vote amount values; and an adder that adds the vote amount value to the accumulated vote amount value, which is received from the output buffer according to the vote direction value, and replaces the accumulated vote amount value, with the newly calculated value. The first gradient value is a difference between values of a first pixel belonging to the third row and a second pixel belonging to the first row. The second gradient value is a difference between values of third and fourth pixels belonging to the second row. The first to fourth pixels are immediately adjacent to a target pixel that is a pixel in the second row. The output buffer outputs all of the accumulated vote amount values to an outside processor that performs a processing using the received accumulated vote amount values.
- [0012] An aspect of any one of embodiments is a microcomputer, including: a data transfer apparatus including first and second processors; and an external memory that is configured to transfer data to the data transfer apparatus. The first processor includes; a

two-row buffer that stores a first row consisted of consecutive pixels of the image and a second row consisted of consecutive pixels of the image, the first row being immediately adjacent to the second row; an input buffer that stores a third row which is consisted of consecutive pixels of the image and transferred from the external memory, the third row being immediately adjacent to the second row; a gradient calculator that calculates first and second gradient values; a vote calculator that calculates a vote amount value based on a sum of a square of the first gradient value and a square of the second gradient value; a direction calculator that calculates a vote direction value based on a ratio of the first and second gradient values; an output buffer that stores accumulated vote amount values, and outputs the accumulated vote amount values; and an adder that adds the vote amount value to the accumulated vote amount value, which is received from the output buffer according to the vote direction value, and replaces the accumulated vote amount value with the newly calculated value. The first gradient value is a difference between values of a first pixel belonging to the third row and a second pixel belonging to the first row. The second gradient value is a difference between values of third and fourth pixels belonging to the second row. The first to fourth pixels are immediately adjacent to a target pixel that is a pixel in the second row. The output buffer outputs all of the accumulated vote amount values to the second processor that performs a processing using the received accumulated vote amount values.

Advantageous Effects of Invention

- [0013] According to an aspect of any one of embodiments, it is possible to achieve a data transfer apparatus to specifically generate the Integral Histogram using data transferred from outside.

Brief Description of Drawings

- [0014] [fig.1]Fig. 1 is a block diagram schematically showing a configuration of a data transfer apparatus according to a first embodiment.
- [fig.2]Fig. 2 is a block diagram schematically showing data transfer for a gradient calculation in a processor according to the first embodiment.
- [fig.3]Fig. 3 is a block diagram schematically showing the data transfer for the gradient calculation in the processor according to the first embodiment.
- [fig.4]Fig. 4 is a block diagram schematically showing a row-update operation of an input buffer and a two-row buffer according to the first embodiment.
- [fig.5]Fig. 5 is a block diagram schematically showing the row-update operation of the input buffer and the two-row buffer according to the first embodiment.
- [fig.6]Fig. 6 is a block diagram showing an example of a configuration of an adder according to the first embodiment.

[fig.7]Fig. 7 is a block diagram showing an example of a configuration of an SIMD array according to the first embodiment.

[fig.8]Fig. 8 is a timing chart showing an operation of the data transfer apparatus according to the first embodiment.

[fig.9]Fig. 9 is a block diagram showing an initial state before the operation of the data transfer apparatus starts.

[fig.10]Fig. 10 is a block diagram showing a state of the data transfer apparatus after a first clock cycle.

[fig.11]Fig. 11 is a block diagram showing a state of the data transfer apparatus after a second clock cycle.

[fig.12]Fig. 12 is a block diagram showing a state of the data transfer apparatus after a third clock cycle.

[fig.13]Fig. 13 is a block diagram showing a state of the data transfer apparatus after a fourth clock cycle.

[fig.14]Fig. 14 is a block diagram schematically showing a configuration of a micro-computer according to a second embodiment.

Description of Embodiments

[0015] Hereinafter, embodiments shall be explained with reference to the drawings. The same components are denoted by the same reference numerals throughout the drawings, and repeated explanations shall be omitted as necessary.

[0016] First Embodiment

A data transfer apparatus according to a first embodiment shall be described. The data transfer apparatus is configured to be able to generate an Integral Histogram during Direct Memory Access (DMA) data transfer from an external memory to Single Instruction Multiple Data (SIMD) array.

[0017] Fig. 1 is a block diagram schematically showing a configuration of a data transfer apparatus 100 according to the first embodiment. As shown in Fig. 1, the data transfer apparatus 100 includes a processor 10 and a SIMD array 20. The processor 10 is a Direct Memory Access (DMA) processor that functions as an Integral Histogram Generator (IHG). The processor 10 can access to an external memory 30 and receive data from the external memory 30. The external memory 30 stores data which is the image data consisted of a plurality of rows. Each row is consisted of a plurality of pixels.

[0018] Here, a configuration of the processor 10 shall be described in detail. The processor 10 includes a controller 1, an input buffer 2, a two-row buffer 3, a gradient calculator 4, a vote calculator 5, a direction calculator 6, an adder 7, and an output buffer 8.

[0019] The controller 1 controls data transfer operations and calculation operations of the

input buffer 2, the two-row buffer 3, the gradient calculator 4, the vote calculator 5, the direction calculator 6, the adder 7, and the output buffer 8.

[0020] The gradient calculator 4 calculates gradient values based on the pixel values in the rows stored in the input buffer 2 and the two-row buffer 3. Hereinafter, a gradient calculation will be described. When the gradient value of a certain pixel (hereinafter, this pixel is referred to as a target pixel.) in horizontal direction is calculated, values of an immediate left pixel and an immediate right pixel of the target pixel are needed. When the gradient value of the target pixel in vertical direction is calculated, values of an immediate upper pixel and an immediate lower pixel of the target pixel are needed. In sum, for the gradient calculation of each pixel, four pixel values, which are the values of the immediate left, right, upper and lower pixels, are required. In other words, the data from three consecutive rows are required to calculate the gradient values in the horizontal and vertical directions of each pixel.

[0021] In this embodiment, the input buffer 2 stores one row and the two-row buffer 3 stores two rows. Fig. 2 is a block diagram schematically showing data transfer for the gradient calculation in the processor according to the first embodiment. Rows R1 and R2 are stored in the row-slots RS1 and RS2 in two-row buffer 3 before the gradient calculation starts. Further, a Row R3 is stored in the input buffer 2 before the gradient calculation starts. In this case, the row R1 is an upper row, the row R2 is a middle row, the row R3 is a lower row, and the rows R1 to R3 are consecutive rows. In this case, the row R1 includes values of six pixels V11 to V16, the row R2 includes values of six pixels V21 to V26, and the row R3 includes values of six pixels V31 to V36. Thus, the immediate upper pixel belongs to the row R1, the immediate left and right pixels belong to the row R2, and the immediate lower pixel belongs to the row R3. In Fig. 2, a value of the target pixel is V22, a value of the upper pixel is V12, a value of the left pixel is V21, a value of the right pixel is V23, and a value of the lower pixel is V32. Therefore, a 1-D centered, point discrete derivative mask in horizontal and vertical direction, e.g., can be used.

[0022] The gradient calculator 4 calculates the gradient values in horizontal and vertical direction (also referred to as first and second gradient values) of each target pixel except for the end pixels (i.e., pixel values V21 and V26) in the middle row R2. Thus, the gradient calculator 4 calculates the gradient values in horizontal and vertical direction of each of the pixels (i.e., pixel values V22 to V25). For example, after the gradient values in horizontal and vertical direction are calculated, the target pixel is moved to the adjacent pixel (V22->V23) as shown in Fig. 3. Fig. 3 is a block diagram schematically showing data transfer for the gradient calculation in the processor according to the first embodiment. In Fig. 3, a value of the target pixel is V23, a value of the upper pixel is V13, a value of the left pixel is V22, a value of the right pixel is

V24, and a value of the lower pixel is V33.

[0023] Then, when the gradient calculator 4 finished calculating the gradient values for the pixels V22 to V25, the rows stored in the input buffer 2 and the two-row buffer 3 are updated. Figs. 4 and 5 are block diagrams schematically showing a row-update operation of the input buffer 2 and the two-row buffer 3 according to the first embodiment. As shown in Fig. 4, the row R3 is firstly transferred from the input buffer 2 to the row-slot RS1 and replaces the row R1. After that, as shown in Fig. 5, the row R2 in the row-slot RS2 and the row R3 in the row-slot RS1 are exchanged each other. Further, a row R4 (V41 to V46), which is an immediate lower row of the row R3, is transferred from the external memory 30 to the input buffer 2 and replaces the row R3. In sum, the middle row R2 before the uploading becomes the upper row after the uploading, the lower row R3 before the uploading becomes the middle row after the uploading, and the lower row R4 is newly loaded. As a result, as compared with the consecutive rows (R1, R2, and R3) before the uploading, the consecutive rows (R2, R3, R4) after the uploading are shifted in the vertical direction by one row.

[0024] The gradient calculator calculates the gradient value in the horizontal direction dx and the gradient value in the vertical direction dy according to the following expressions (1) and (2), respectively, for example.

$$dx = \text{right_pixel_value} - \text{left_pixel_value} \text{ --- (1)}$$

$$dy = \text{lower_pixel_value} - \text{upper_pixel_value} \text{ --- (2)}$$

The gradient value in the horizontal direction dx and the gradient value in the vertical direction dy are given to the vote calculator 5 and the direction calculator 6.

[0025] The vote calculator 5 calculates a vote amount value VA in each clock cycle based on the gradient value in the horizontal direction dx and the gradient value in the vertical direction dy calculated in the previous clock cycle. The vote amount value VA is expressed by a following expression (3).

$$VA = dx^2 + dy^2 \text{ --- (3)}$$

[0026] The direction calculator 6 calculates a vote direction value VD in each clock cycle based on the gradient value in the horizontal direction dx and the gradient value in the vertical direction dy calculated in the previous clock cycle. The vote amount value VA is expressed by a following expression (4), for example.

$$VD = \arctan(dy/dx) \text{ --- (4)}$$

Because an exact value is not needed but only the partition into one of the histogram channels, an approximation of this equation is possible by, e.g., using table look-ups. The vote amount value VA and vote direction value VD are then given to the adder 7.

[0027] The output buffer 8 includes the slots for storing accumulated vote amount value for each direction. In this case, the output buffer 8 includes eight slots corresponding to eight directions. The output buffer 8 sends out all accumulated vote values and a mul-

tiplexer is selecting one of the accumulated vote values based on the vote direction value VD.

[0028] The adder 7 adds the vote amount VA from the vote calculator 5 onto the selected accumulated vote amount value received from the output buffer 8. After that, the adder 7 sends out the accumulated value to the slot of the output buffer 8 which is selected based on the vote direction value VD. Then the added value is stored in the slot corresponding to the vote direction value VD as an updated accumulated vote amount value. Further, the accumulated vote amount values in all slots of the output buffer 8 are output to the SIMD array 20.

[0029] Here, an example of the configuration of the adder 7 shall be described. Fig. 6 is a block diagram showing the example of the configuration of the adder 7 according to the first embodiment. As shown in Fig. 6, the adder 7 includes a multiplexer 71, an adder unit 72, and a demultiplexer 73. The multiplexer 71 receives all of the accumulated vote amount values A0 to A7 in the slots SL0 to SL7 of the output buffer 8 and selects one based on the vote direction value VD. In this case, the multiplexer 71 selects the accumulated vote amount value A1 in the slots SL1 of the output buffer 8 based on the vote direction value VD. The adder unit 72 adds the vote amount value VA to the received accumulated vote amount value (A1) and outputs the added value (VA+A1) to the demultiplexer 73. Then, the demultiplexer 73 outputs the added value (VA+A1) to the same slot (A1) which was selected by the multiplexer 71 based on the vote direction value VD. As a result, the value in this slot is updated (A1 -> VA+A1).

[0030] The SIMD array 20 adds the accumulated vote values received from the output buffer 8 to the accumulated vote amount values which were generated by the operations for the previous row, for example. Fig. 7 is a block diagram showing an example of a configuration of the SIMD array 20 according to the first embodiment.

[0031] The SIMD array 20 includes a pipelined ring bus 21, a buffer array 22, an adder array 23, and a memory element array 24. The pipelined ring bus 21 receives the accumulated vote amount values AVV from the output buffer 8. The buffer array 22 holds the accumulated vote amount values AVV_PRE which were generated by the operations for the previous row. The adder array 23 adds each of the accumulated vote amount values AVV to the corresponding accumulated vote amount values AVV_PRE. The added accumulated vote amount values are stored in the memory element array 24. The buffer array 22 is always filled in time with data from the memory element array 24. The transferred data is the data from the last row for the same horizontal positions.

[0032] Hereinafter, operations of the data transfer apparatus 100 at various clock cycles shall be described. Fig. 8 is a timing chart showing the operation of the data transfer apparatus 100 according to the first embodiment. In the Fig. 8, numerical signs T0 to

T4 point out ends of clock cycles T0 to T4. In this case, the two-row buffer 3 is already filled with the upper two rows, and the input buffer 2 is already filled with the current row R3 elements, until the end of the clock cycle T0.

[0033] In the clock cycle T1 which is a clock cycle after the clock cycle T0, the gradient calculator 4 outputs the gradient value in the horizontal direction dx and the gradient value in the vertical direction dy of the target pixel. In the clock cycle T2 which is a clock cycle after the clock cycle T1, the vote calculator 5 calculates the vote amount value VA and the direction calculator 6 calculates the vote direction value VD. In the clock cycle T3 which is a clock cycle after the clock cycle T2, the operations in the adder 7 and the output buffer 8 are executed based on the vote amount value VA and the vote direction value VD calculated in the clock cycle T2. In the clock cycle T4 which is one or more clock cycle after the clock cycle T3, the SIMD array 20 executes the accumulated vote values addition as described above.

[0034] Fig. 9 is a block diagram showing an initial state before the operation of the data transfer apparatus 100 starts. Here, the two-row buffer 3 holds the upper and middle rows, while the input buffer 2 holds the lower row which is received from the external memory 30. Specifically, the row-slot RS1 holds a first row consisted of "1 2 3 4 5 6", the row-slot RS2 holds a second row consisted of "3 4 5 7 8 9", and the input buffer 2 holds a third row consisted of "1 3 5 7 9 2".

[0035] First clock cycle

< Operation for a first target pixel >

Fig. 10 is a block diagram showing a state of the data transfer apparatus 100 after the first clock cycle. In the first clock cycle, the gradient value in the horizontal direction dx and the gradient value in the vertical direction dy for the first target pixel "4" (surrounded by a square in Fig. 10) are calculated as expressed by the following expressions (5) and (6). In this case, the right value, left value, lower value, and upper value are "5", "3", "3", and "2", respectively.

$$dx = \text{right_pixel_value} - \text{left_pixel_value} = 5 - 3 = 2 \text{ --- (5)}$$

$$dy = \text{lower_pixel_value} - \text{upper_pixel_value} = 3 - 2 = 1 \text{ --- (6)}$$

[0036] Second clock cycle

< Operation for the first target pixel >

Fig. 11 is a block diagram showing a state of the data transfer apparatus 100 after the second clock cycle. In the second clock cycle, the vote amount value VA for the first target pixel is calculated in the vote calculation unit 5 as expressed by the following expressions (7) based on the gradient value in the horizontal direction dx and the gradient value in the vertical direction dy calculated in the first clock cycle.

$$VA = dx^2 + dy^2 = 2^2 + 1^2 = 5 \text{ --- (7)}$$

Further, the vote direction value VD for the first target pixel is calculated in the

arctangent calculation unit 6 as expressed by the following expressions (8) based on the dx and dy calculated in the first clock cycle, for example.

$VD = \text{floor}(\arctan(dy/dx) / (180 / \text{sup_dir}))$ for $dy/dx \geq 0$ and

$VD = \text{floor}((180 + \arctan(dy/dx)) / (180 / \text{sup_dir}))$ for $dy/dx < 0$ --- (8)

In this case, the arctangent calculation unit 6 decides the vote direction value VD from the eight directions, so that $\text{sup_dir} = 8$. As $dx=2$ and $dy=1$, the vote direction value VD is expressed by the following expressions (9).

$VD = \text{floor}(\arctan(1/2) / (180 / 8)) = 1$ --- (9)

[0037] < Operation for a second target pixel >

Furthermore, in the second clock cycle, the gradient value in the horizontal direction dx and the gradient value in the vertical direction dy for the second pixel "5" (surrounded by a square in Fig. 11) are calculated as expressed by the following expressions (10) and (11). In this case, the right value, left value, lower value, and upper value are "7", "4", "5", and "3", respectively.

$dx = \text{right_pixel_value} - \text{left_pixel_value} = 7 - 4 = 3$ --- (10)

$dy = \text{lower_pixel_value} - \text{upper_pixel_value} = 5 - 3 = 2$ --- (11)

[0038] Third clock cycle

< Operation for the first target pixel >

Fig. 12 is a block diagram showing a state of the data transfer apparatus 100 after the third clock cycle. In the third clock cycle, the adder 7 receives the value from any one of the elements SL0 to SL7 of the output buffer 8 corresponding to the vote direction value VD. Then, the adder 7 adds the newest vote amount VA to the received value from the output buffer 8. After that, the adder 7 sends out the added value to the slot of the output buffer 8 corresponding to the vote direction value VD. Further, all values stored in the slots SL0 to SL7 are transferred to the pipelined ring bus 21 in the SIMD array 20. In this case, as $VD=1$, the adder receives "0" from the first element SL1 of the output buffer 8. Then, as the newest vote amount VA is "5", the adder 7 adds "5" to "0" and sends out the added value "5" to the first element SL1 of the output buffer 8.

[0039] < Operation for the second target pixel >

Further, in the third clock cycle, the vote amount value VA is calculated in the vote calculator 5 as expressed by the following expressions (12). The vote direction value VD is expressed by the following expressions (13).

$VA = dx^2 + dy^2 = 3^2 + 2^2 = 13$ --- (12)

$VD = \text{floor}(\arctan(2/3) / (180/8)) = 1$ --- (13)

[0040] < Operation for a third target pixel >

Furthermore, in the third clock cycle, the gradient value in the horizontal direction dx and the gradient value in the vertical direction dy for the third pixel "7" (surrounded by a square in Fig. 12) are calculated as expressed by the following expressions (14) and

(15) as shown in Fig. 12. In this case, the right value, left value, lower value, and upper value are "8", "5", "7", and "4", respectively.

$$dx = \text{right_pixel_value} - \text{left_pixel_value} = 8 - 5 = 3 \quad \text{--- (14)}$$

$$dy = \text{lower_pixel_value} - \text{upper_pixel_value} = 7 - 4 = 3 \quad \text{--- (15)}$$

[0041] Fourth clock cycle

< Operation for the first target pixel >

In the SIMD array 20, the adder array 23 adds the accumulated vote amount values AVV "0 5 0 0 0 0 0" stored in the pipelined ring bus 21 to the accumulated vote amount values AVV_PRE "1 1 1 1 1 1 1" stored in the buffer array 22, respectively. Note that, the accumulated vote amount values AVV_PRE "1 1 1 1 1 1 1" are loaded in advance. Then added values "1 6 1 1 1 1 1" are stored in the memory element array 24.

[0042] < Operation for the second target pixel >

Fig. 13 is a block diagram showing a state of the data transfer apparatus 100 after the fourth clock cycle. In the fourth clock cycle, as VD=1, the adder 7 receives "5" from the slot SL1 of the output buffer 8. Then, as the newest vote amount value VA is "13", the adder 7 adds "13" to "5" and sends out the added value "18" to the slot SL1 of the output buffer 8. Further, in the pipelined ring bus 21, the accumulated vote values "0 5 0 0 0 0 0" stored in the third clock cycle are replaced with the accumulated vote values "0 18 0 0 0 0 0" transferred in the fourth clock cycle.

[0043] < Operation for the third target pixel >

Further, in the fourth clock cycle, the vote amount value VA is calculated in the vote calculator 5 as expressed by the following expressions (16). The vote direction value VD is expressed by the following expressions (17).

$$VA = dx^2 + dy^2 = 3^2 + 3^2 = 18 \quad \text{--- (16)}$$

$$VD = \text{floor}(\arctan(3/3) / (180/8)) = 2 \quad \text{--- (17)}$$

[0044] < Operation for a fourth target pixel >

Furthermore, in the fourth clock cycle, the gradient value in the horizontal direction dx and the gradient value in the vertical direction dy for the fourth pixel "8" (surrounded by a square in Fig. 13) are calculated as expressed by the following expressions (18) and (19) as shown in Fig. 13. In this case, the right value, left value, lower value, and upper value are "9", "7", "9", and "5", respectively.

$$dx = \text{right_pixel_value} - \text{left_pixel_value} = 9 - 7 = 2 \quad \text{--- (18)}$$

$$dy = \text{lower_pixel_value} - \text{upper_pixel_value} = 9 - 5 = 4 \quad \text{--- (19)}$$

[0045] As describe above, the four clock cycles are needed for one target pixel. Thus, when there are M (M is an integer lager than one) target pixels in a row, M+3 clock cycles are need to perform calculations for all target pixels of a row.

[0046] According to the configuration described above, it is possible to efficiently generate

the Integral Histogram while using the DMA data transfer for a data read transfer from external to internal memory. The data transfer apparatus receives the rows one by one and stores only three rows, so that size of the input buffer and two-row buffer can be suppressed as compared with the case in which the whole image or a partial area of the image is transferred from the external memory to the data transfer apparatus. Further, the output buffer stores only one accumulated vote amount value for one bin, so that size of the output buffer can be also suppressed.

[0047] Further, in each clock cycle, the data transfer apparatus according to the first embodiment updates only one of the accumulated vote amount values for one bin of a pixel in the output buffer and transfers the accumulated vote amount values from the output buffer to the SIMD array. Thus, the data transfer apparatus according to the first embodiment can achieve the rapid histogram generation.

[0048] Second Embodiment

Next, a microcomputer 200 according to a second embodiment shall be explained. The microcomputer 200 is an example of a microcomputer in which the data transfer apparatus 100 is incorporated. Fig. 14 is a block diagram schematically showing a configuration of the microcomputer 200 according to the second embodiment. As shown in Fig. 14, the microcomputer 200 includes the data transfer apparatus 100, the external memory 30, and a bus 40. Data can be transferred among the data transfer apparatus 100 and the external memory 30 via the bus 40.

[0049] As described above, the data transfer apparatus can be applied to a microcomputer. Therefore, according to the configuration, it is possible to suppress the size of the input buffer, two-row buffer and output buffer in the data transfer apparatus and achieve the rapid histogram generation as well as the first embodiment described above.

[0050] Other embodiments

The present invention is not limited to the embodiments described above, and can be modified as appropriate without departing from the scope of the invention. For example, in the above embodiments, it is described that the rows stored in the input buffer and the two-row buffer include the values of the six pixels. However, it is merely an example. Therefore, the number of pixels in the row may be three to five, or more than seven.

[0051] In the above embodiments, the input buffer stores the lower row. However, it is merely an example. Thus, the input buffer may store the upper row and the one slot of the two-row buffer may store the lower row. Further, in this case, the input buffer may replace the row stored therein with a more upper row received from the external memory.

[0052] In the above embodiments, the input buffer has the size of a whole row. However, this is merely an example. Thus the input buffer can also hold only a few or even only

one pixel. In such case, instead of replacing a whole row at the end of the row processing, the pixels of the input buffer are replacing the oldest pixel inside the two-row buffer, which are the pixel on the left side of the first row, during the processing of the pixel row. At the end of the one pixel row processing, both rows of the two-row buffer are then finally swapped.

[0053] In the above embodiments, the vote amount value is the sum of the square of the first gradient value and the square of the second gradient value as expressed by the expression (3). However, it is merely an example. For example, the vote amount value may be a square root of the sum of the square of the first gradient value and the square of the second gradient value.

[0054] In the above embodiments, it is described that the number of directions is eight. However, it is merely an example. Thus, the number of directions may be arbitrary plural number.

Reference Signs List

[0055] 1 CONTROLLER
 2 INPUT BUFFER
 3 TWO-ROW BUFFER
 4 GRADIENT CALCULATOR
 5 VOTE CALCULATOR
 6 DIRECTION CALCULATOR
 7 ADDER
 8 AN OUTPUT BUFFER
 10 PROCESSOR
 20 SIMD ARRAY
 21 PIPELINED RING BUS
 22 BUFFER ARRAY
 23 ADDER ARRAY
 24 MEMORY ELEMENT ARRAY
 30 EXTERNAL MEMORY
 71 MULTIPLEXER
 72 ADDER UNIT
 73 DEMULTIPLEXER
 100 DATA TRANSFER APPARATUS
 200 MICROCOMPUTER
 A0 to A7 ACCUMULATED VOTE VALUES
 SL0 to SL7 SLOTS
 VA VOTE AMOUNT VALUE

VD VOTE DIRECTION VALUE

Claims

[Claim 1]

A data transfer apparatus, comprising:
a two-row buffer that stores a first row consisted of consecutive pixels of the image and a second row consisted of consecutive pixels of the image, the first row being immediately adjacent to the second row;
an input buffer that stores a third row which is consisted of consecutive pixels of the image and transferred from outside, the third row being immediately adjacent to the second row;
a gradient calculator that calculates first and second gradient values;
a vote calculator that calculates a vote amount value based on a sum of a square of the first gradient value and a square of the second gradient value;
a direction calculator that calculates a vote direction value based on a ratio of the first and second gradient values;
an output buffer that stores accumulated vote amount values, and outputs the accumulated vote amount values; and
an adder that adds the vote amount value to the accumulated vote amount value, which is received from the output buffer according to the vote direction value, and replaces the accumulated vote amount value, with the newly calculated value, wherein
the first gradient value is a difference between values of a first pixel belonging to the third row and a second pixel belonging to the first row, the second gradient value is a difference between values of third and fourth pixels belonging to the second row, the first to fourth pixels are immediately adjacent to a target pixel that is a pixel in the second row, and
the output buffer outputs all of the accumulated vote amount values to an outside processor that performs a processing using the received accumulated vote amount values.

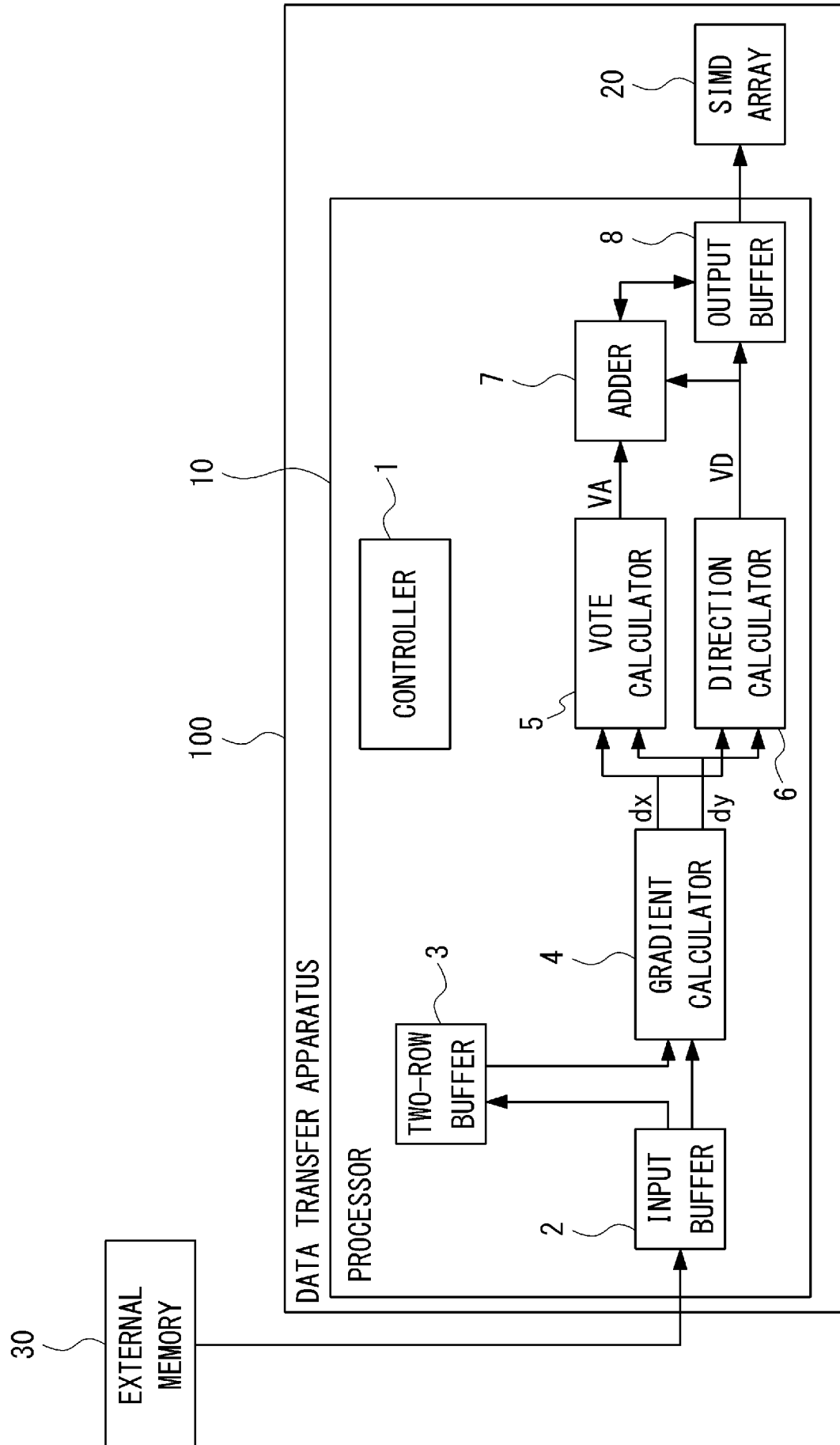
[Claim 2]

The data transfer apparatus according to Claim 1, wherein
when the gradient calculator finishes calculating the first gradient values and the second gradient values for a defined number of processed pixel larger zero, the processed pixel of the input buffer are replacing the oldest pixel of the two-row buffer, which are the pixel on the left side of the first row in the two-row buffer, and
when all pixels in one row have been processed, both rows of the two-row buffer are swapped.

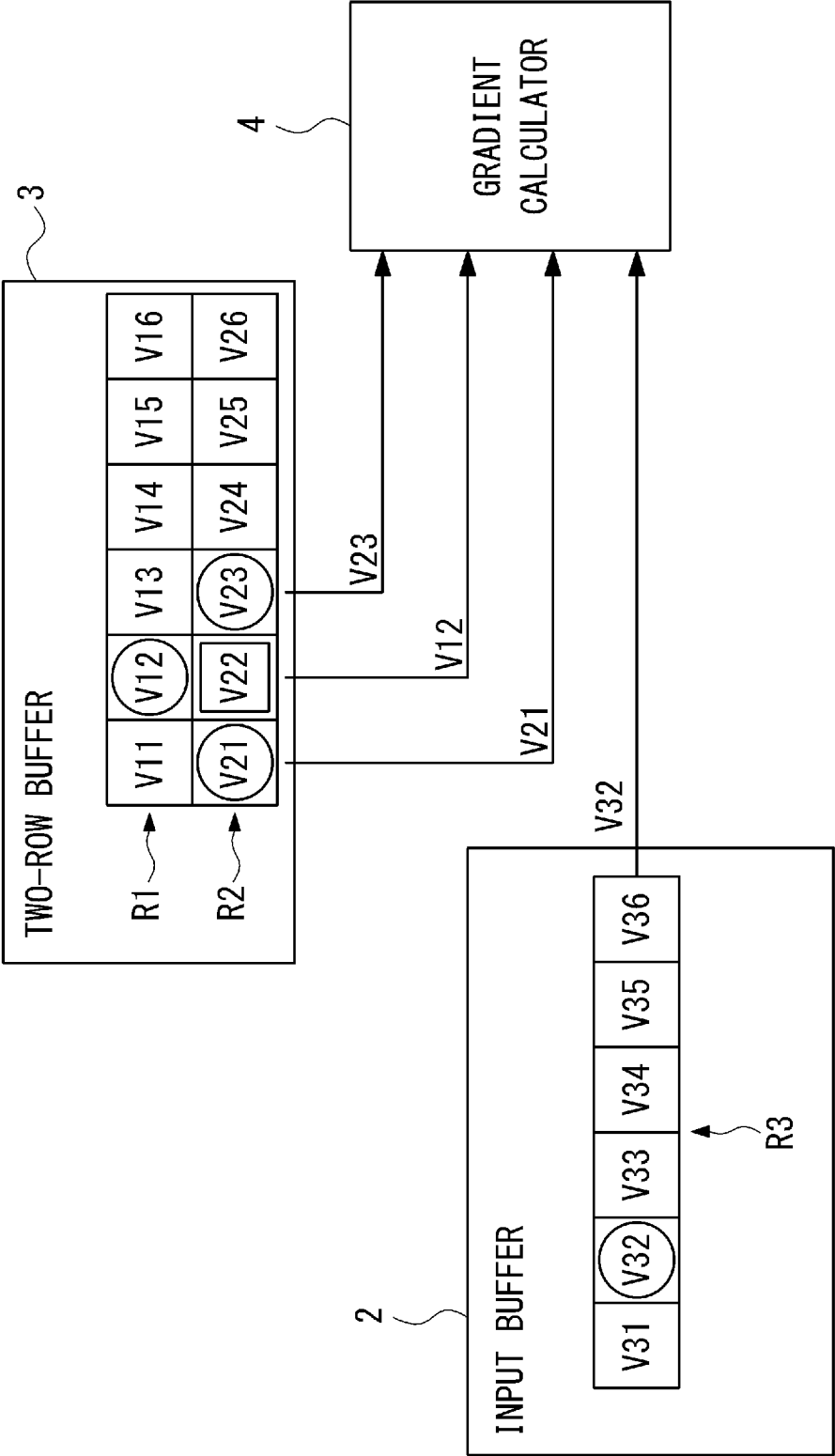
- [Claim 3] The data transfer apparatus according to Claim 1, wherein the direction calculator calculates the vote direction value based on an arctangent of the first and second gradient values.
- [Claim 4] The data transfer apparatus according to Claim 1, wherein the vote amount value is the sum of the square of the first gradient value and the square of the second gradient value.
- [Claim 5] The data transfer apparatus according to Claim 1, wherein the vote amount value is a square root of the sum of the square of the first gradient value and the square of the second gradient value.
- [Claim 6] The data transfer apparatus according to Claim 1, wherein the vote direction value indicates anyone of the N (N is an integer larger than two) directions based on the arctangent of the first and second gradient values.
- [Claim 7] The data transfer apparatus according to Claim 6, wherein the output buffer includes N slots corresponding to the N directions, respectively, and each of the N slots stores the accumulated vote amount value corresponding to the vote direction value.
- [Claim 8] The data transfer apparatus according to Claim 1, further comprising the outside processor, wherein the outside processor comprises:
a first buffer that stores the accumulated vote amount values received from the output buffer;
a second buffer that stores temporary from a local memory received accumulated vote amount values for pixels with same horizontal position of a row, the row being immediately adjacent to the current processed row; and
an adder array that adds each of the accumulated vote amount values in the first buffer to each of the accumulated vote amount values in the second buffer, and
the outside processor stores the output of the adder array to a local memory.
- [Claim 9] A microcomputer, comprising:
a data transfer apparatus including first and second processors; and
an external memory that is configured to transfer data to the data transfer apparatus, wherein
the first processor comprises;
a two-row buffer that stores a first row consisted of consecutive pixels

of the image and a second row consisted of consecutive pixels of the image, the first row being immediately adjacent to the second row;
an input buffer that stores a third row which is consisted of consecutive pixels of the image and transferred from the external memory, the third row being immediately adjacent to the second row;
a gradient calculator that calculates first and second gradient values;
a vote calculator that calculates a vote amount value based on a sum of a square of the first gradient value and a square of the second gradient value;
a direction calculator that calculates a vote direction value based on a ratio of the first and second gradient values;
an output buffer that stores accumulated vote amount values, and outputs the accumulated vote amount values; and
an adder that adds the vote amount value to the accumulated vote amount value, which is received from the output buffer according to the vote direction value, and replaces the accumulated vote amount value with the newly calculated value, wherein
the first gradient value is a difference between values of a first pixel belonging to the third row and a second pixel belonging to the first row,
the second gradient value is a difference between values of third and fourth pixels belonging to the second row,
the first to fourth pixels are immediately adjacent to a target pixel that is a pixel in the second row, and
the output buffer outputs all of the accumulated vote amount values to the second processor that performs a processing using the received accumulated vote amount values.

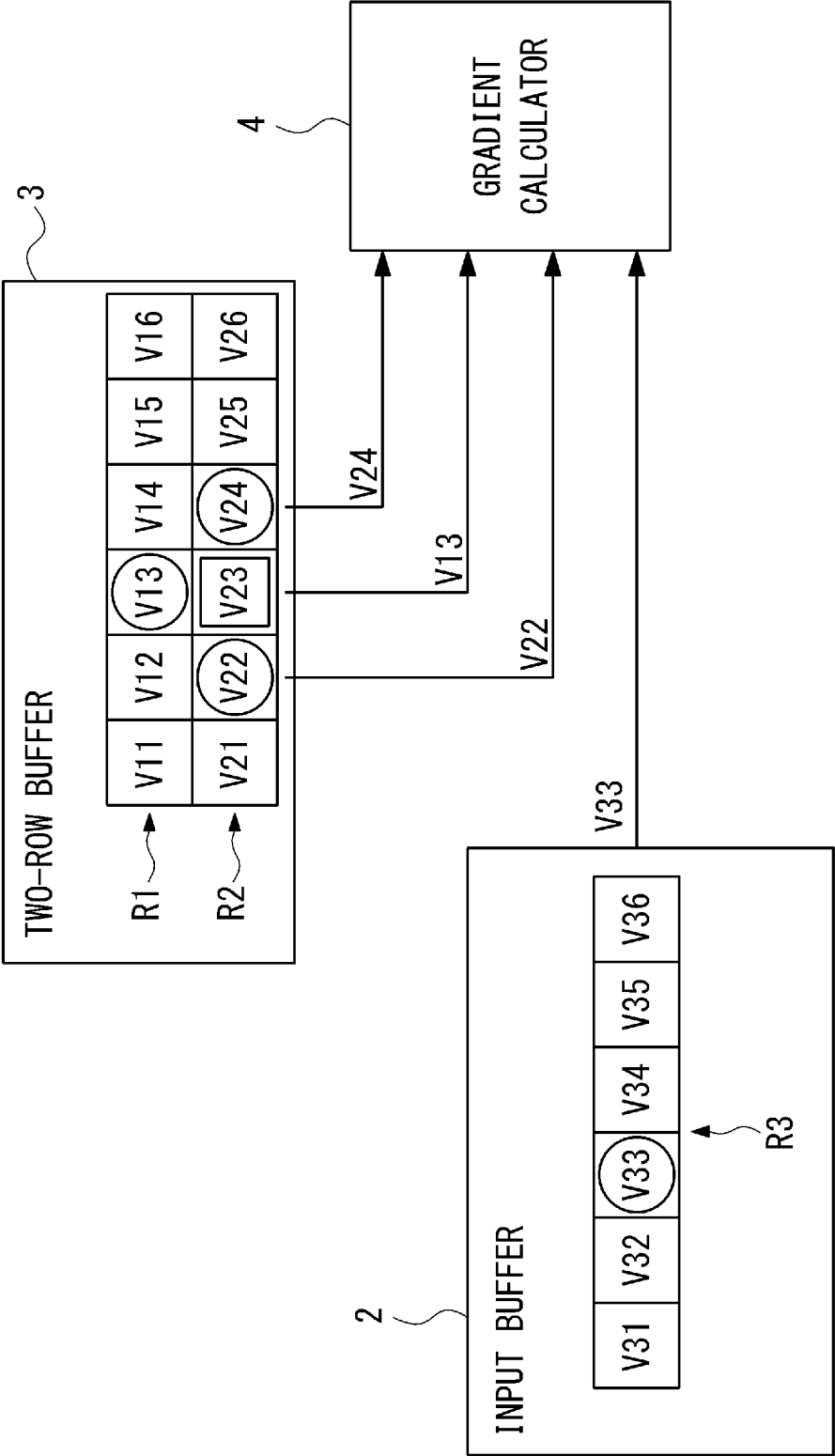
[Fig. 1]



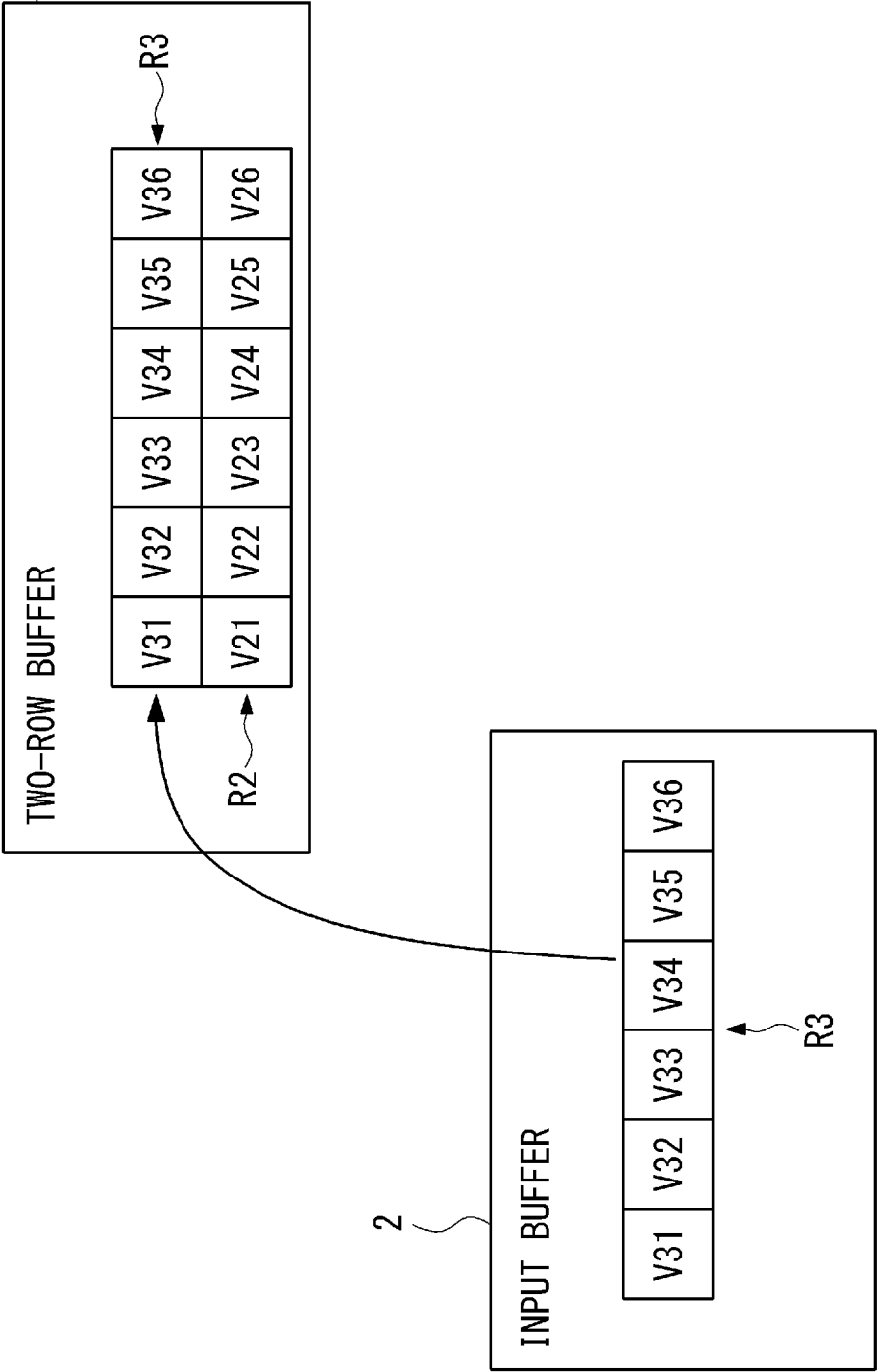
[Fig. 2]



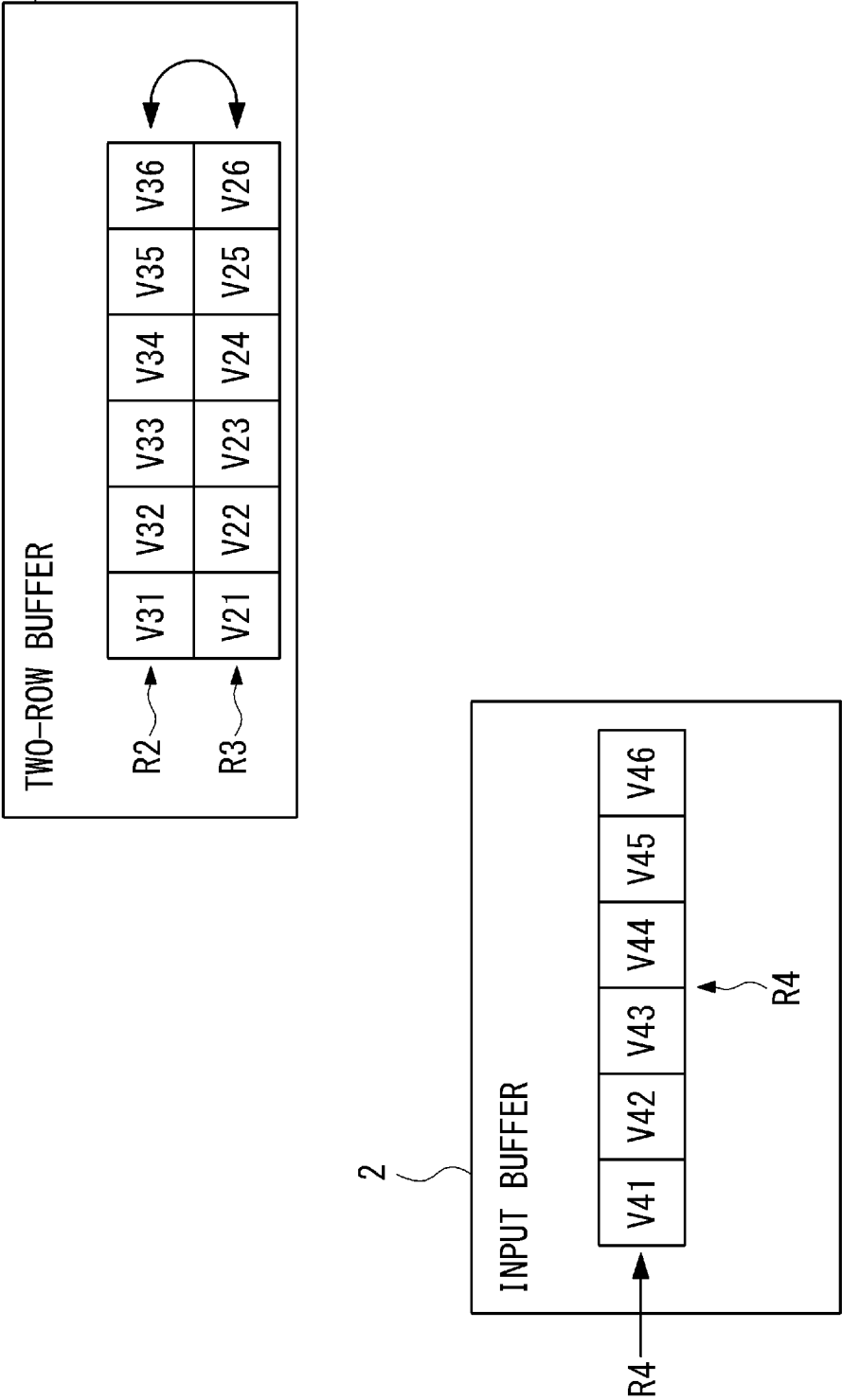
[Fig. 3]



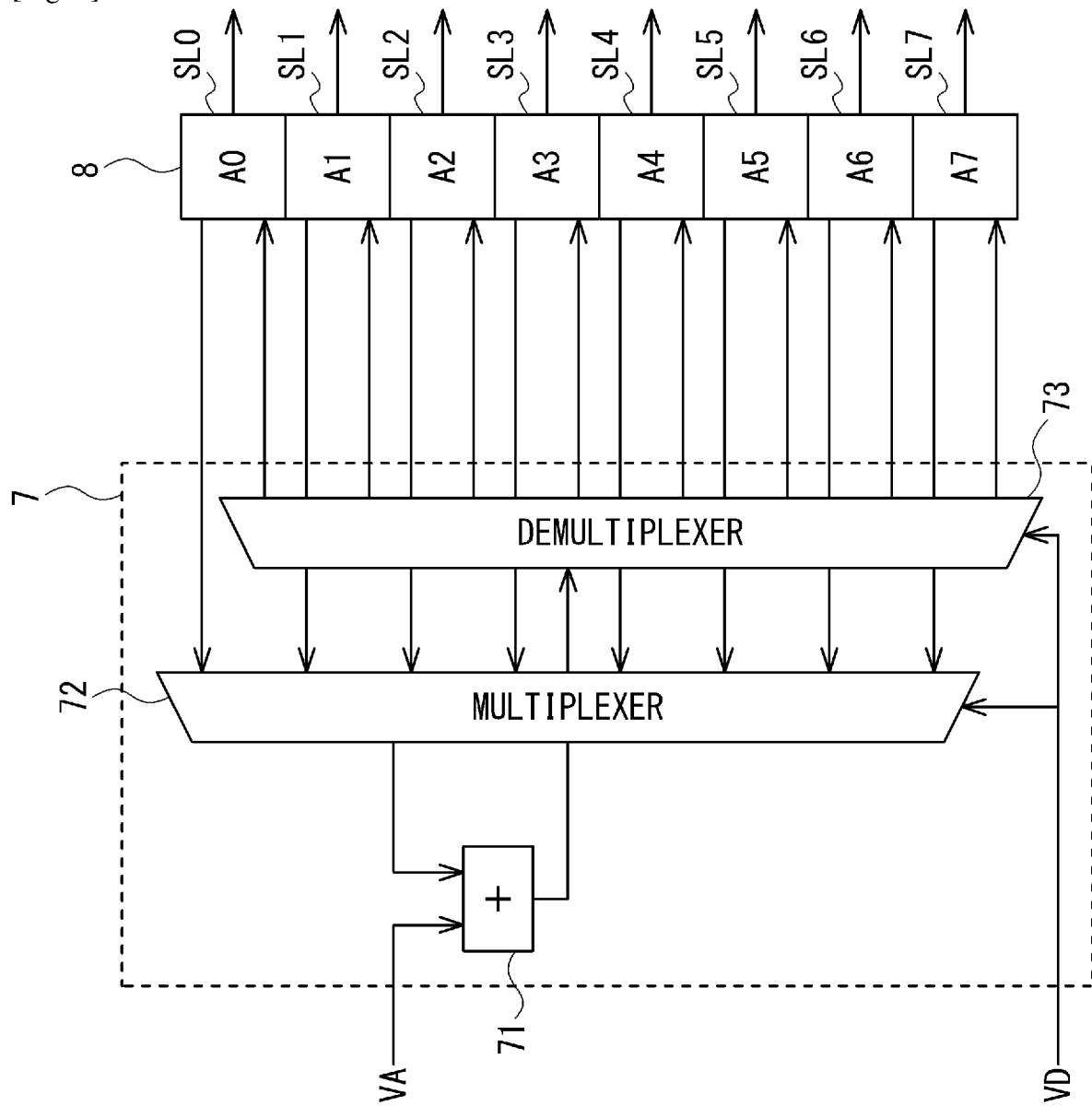
[Fig. 4]



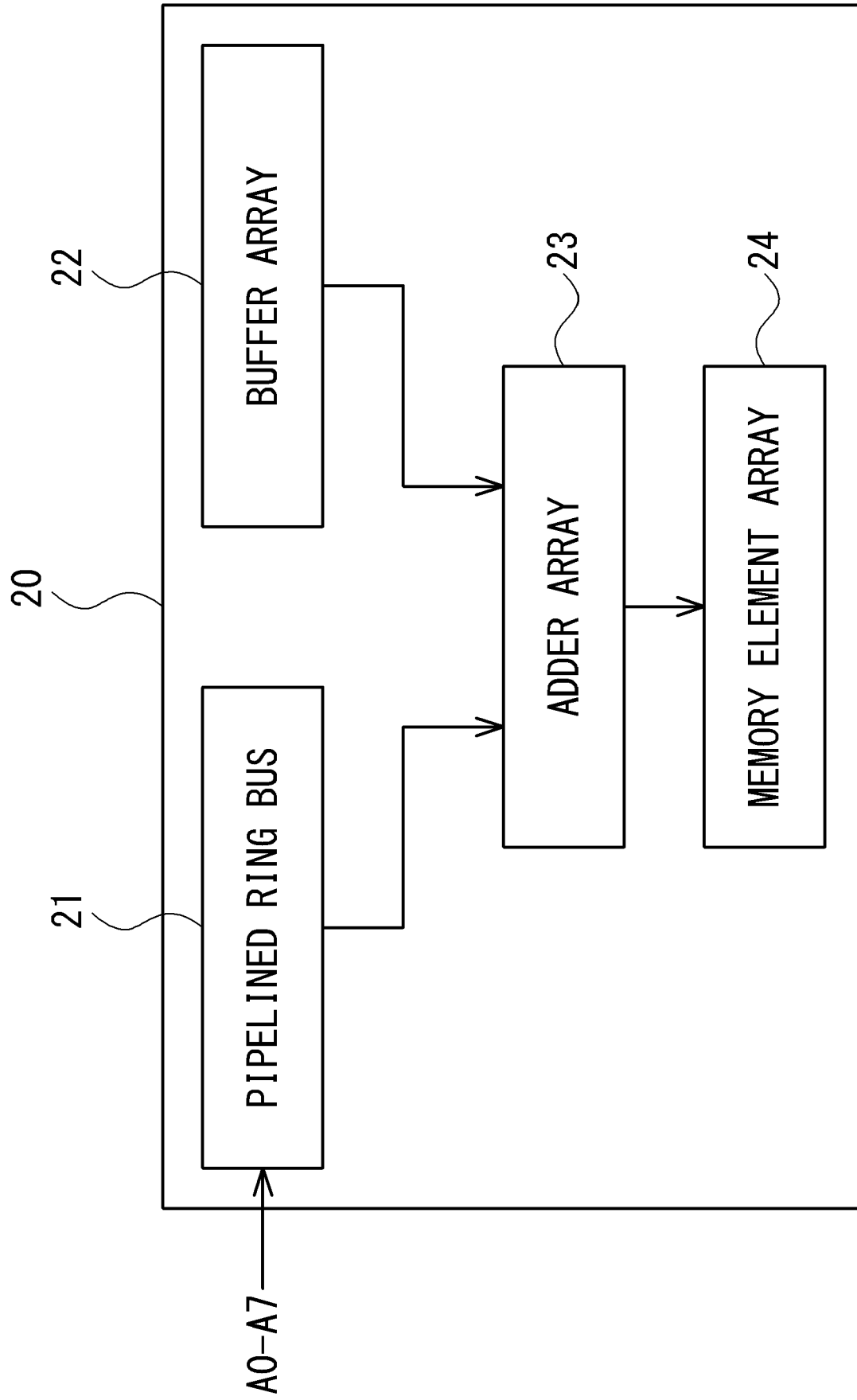
[Fig. 5]



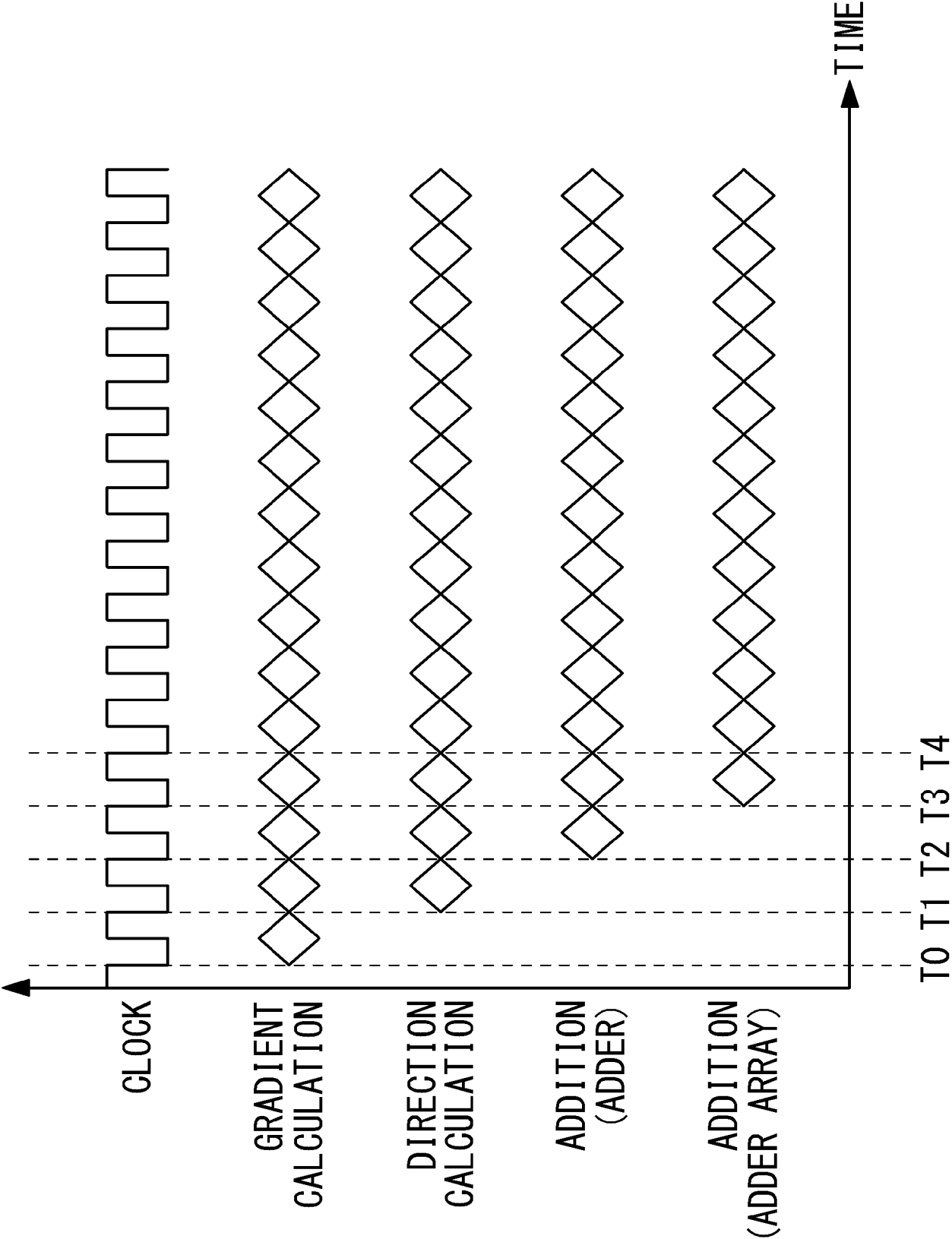
[Fig. 6]



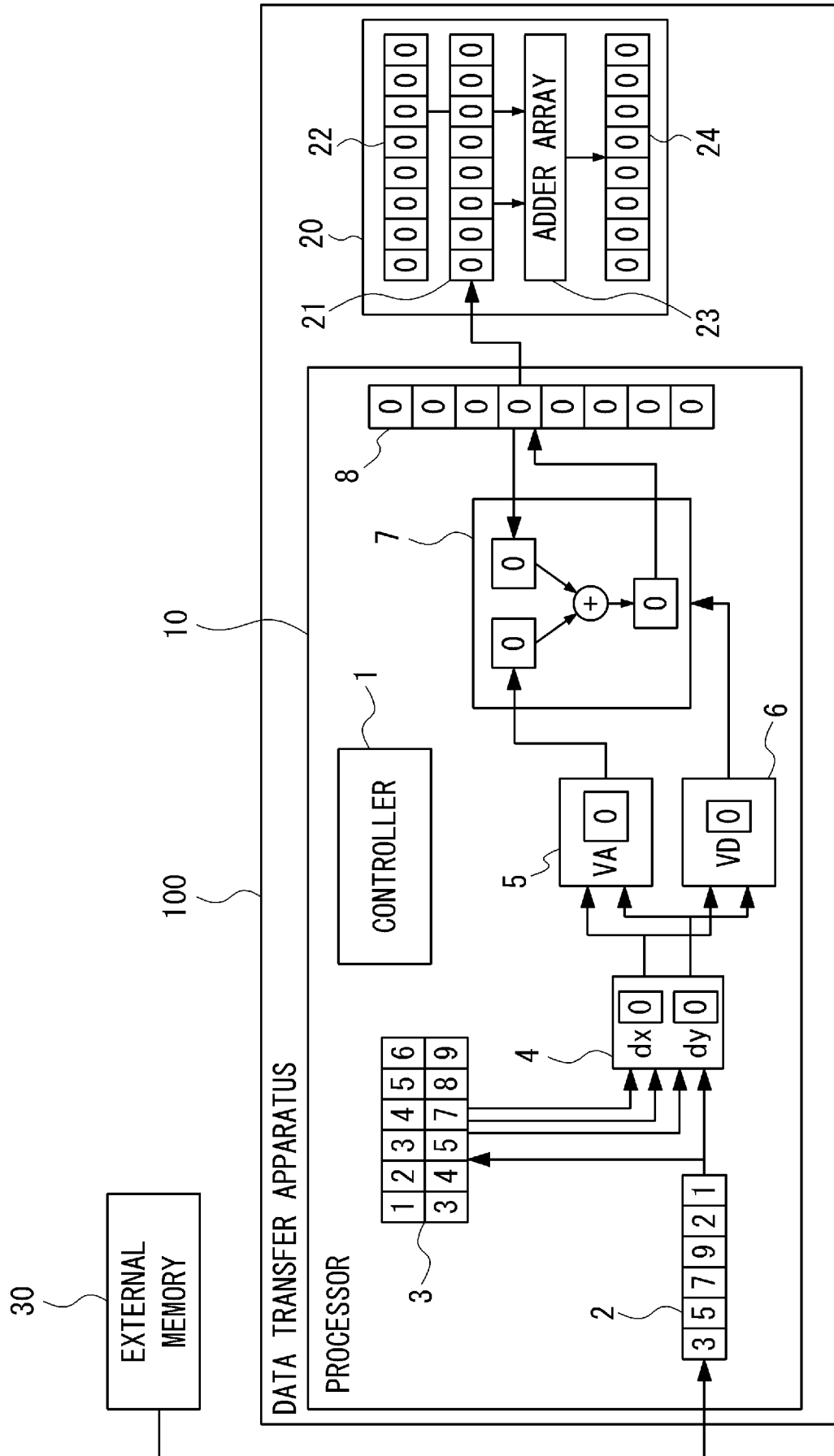
[Fig. 7]



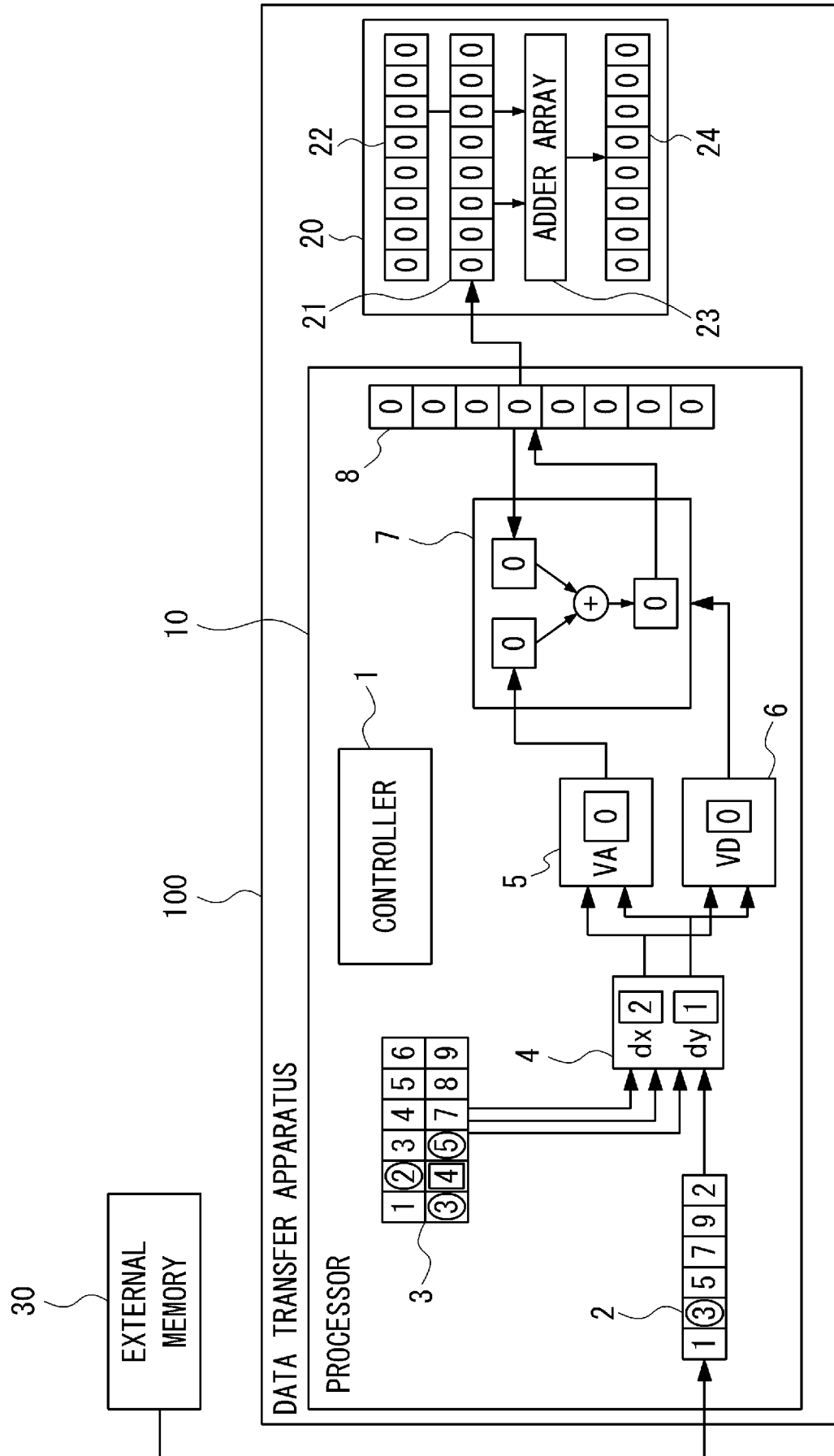
[Fig. 8]



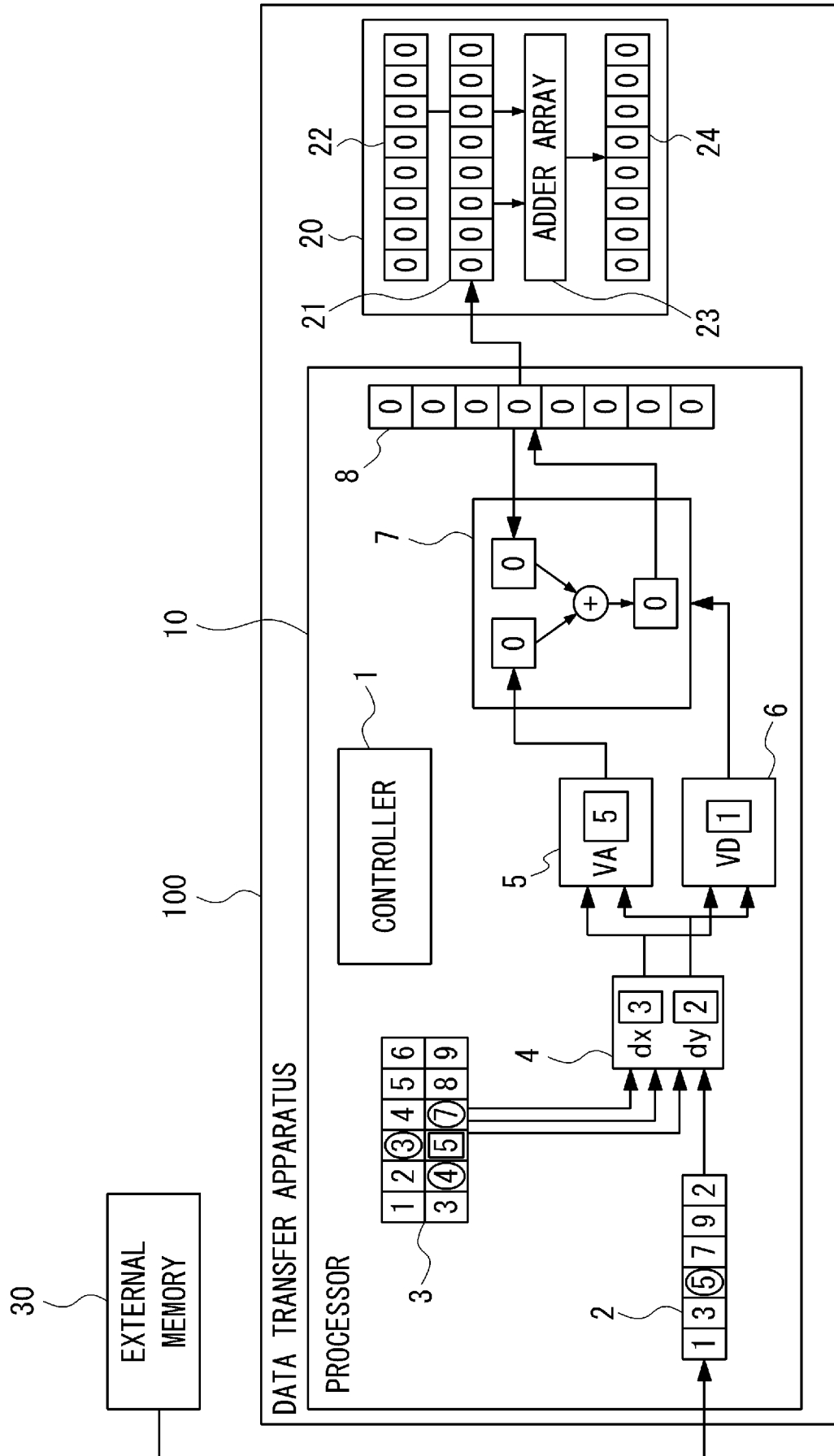
[Fig. 9]



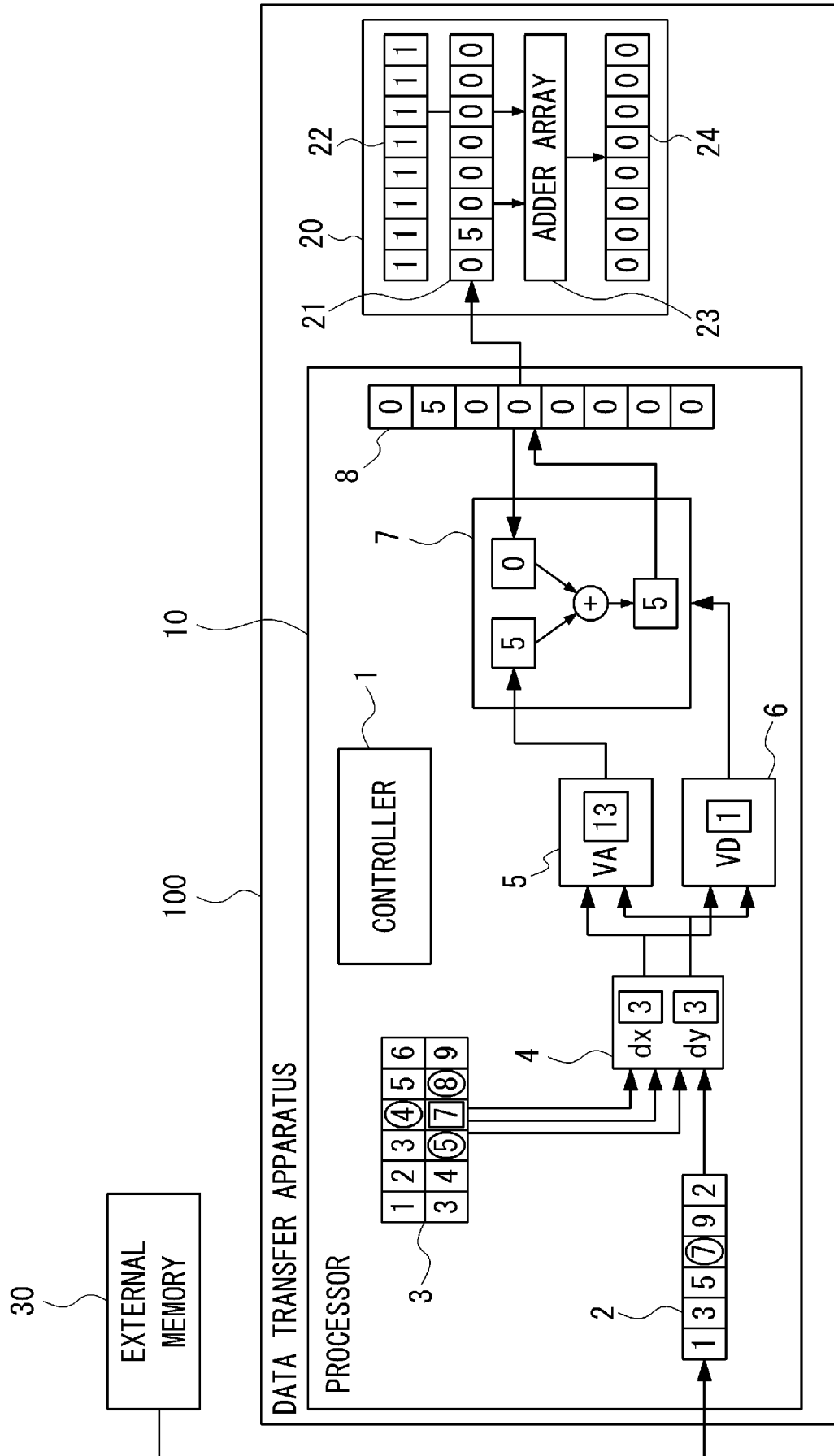
[Fig. 10]



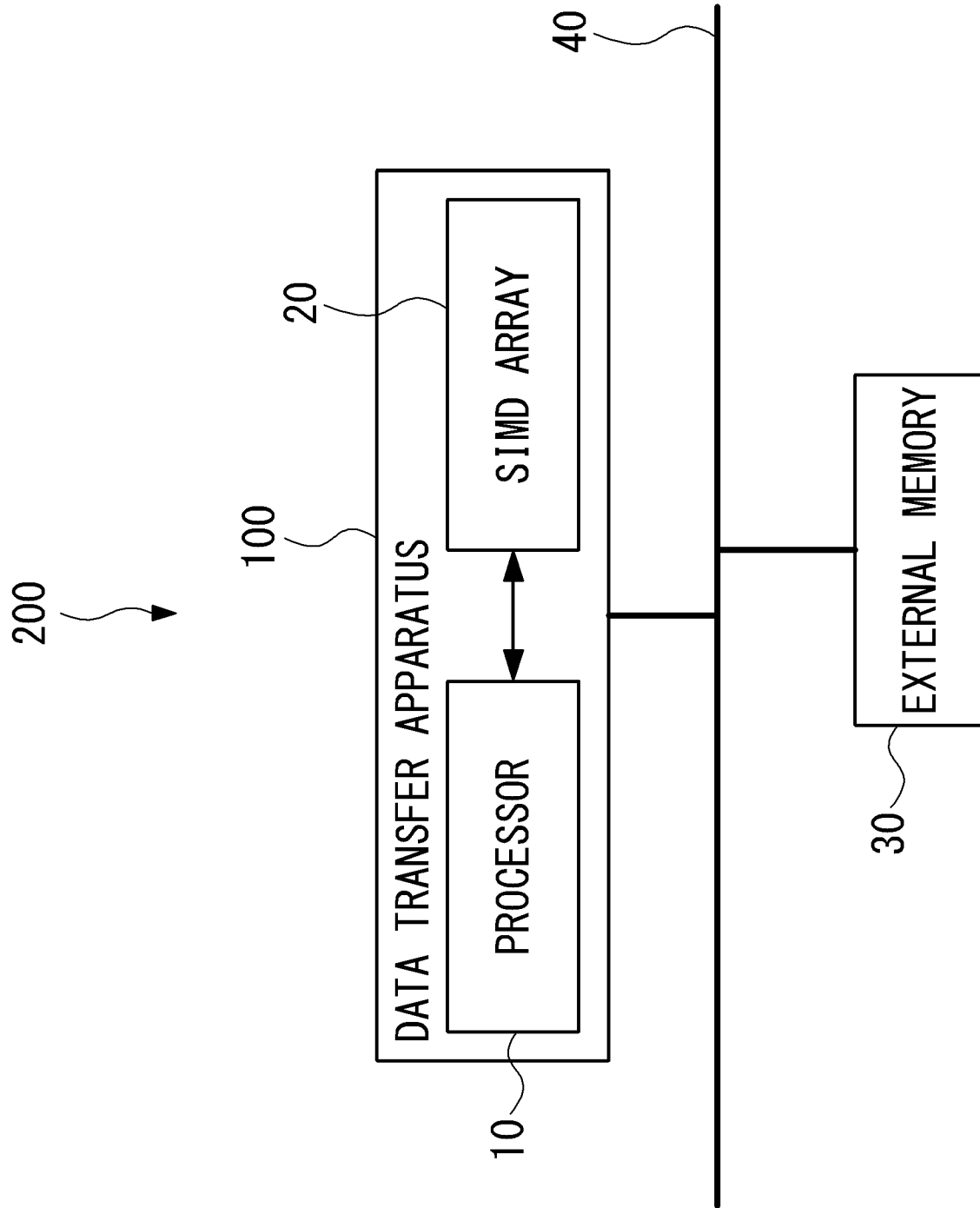
[Fig. 11]



[Fig. 12]



[Fig. 14]



INTERNATIONAL SEARCH REPORT

International application No
PCT/JP2015/001808

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06T1/20
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	LEE ET AL.: "Accelerating Histograms of Oriented Gradients descriptor extraction for pedestrian recognition", COMPUTERS AND ELECTRICAL ENGINEERING, vol. 39, 2013, pages 1043-1048, XP002751306, page 1044, paragraph 3 - page 1046, paragraph 2; figures 1, 3 ----- -/--	1-9



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

2 December 2015

Date of mailing of the international search report

15/12/2015

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Pierfederici, A

INTERNATIONAL SEARCH REPORT

International application No
PCT/JP2015/001808

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	RAMAKRISHNA ET AL.: "Hardware Acceleration of Histogram Equalization and Image Sharpening Filter on NIOS-II Processor based SOC on FPGA", INTERNATIONAL JOURNAL OF COMPUTER APPLICATIONS, vol. 31, no. 1, October 2011 (2011-10), pages 48-54, XP002751307, page 49, right-hand column, last paragraph - page 50, left-hand column, paragraph 1; figure 1 -----	1-9
A	BELLENS PIETER ET AL: "Parallel Implementation of the Integral Histogram", 22 August 2011 (2011-08-22), GRID AND COOPERATIVE COMPUTING - GCC 2004 : THIRD INTERNATIONAL CONFERENCE, WUHAN, CHINA, OCTOBER 21 - 24, 2004. IN: LECTURE NOTES IN COMPUTER SCIENCE, SPRINGER VERLAG, XP047266354, ISSN: 0302-9743, page 589, paragraph 3 - page 592, paragraph 3 -----	1-9
A	MAHDIEH POOSTCHI ET AL: "Efficient GPU Implementation of the Integral Histogram", 5 November 2012 (2012-11-05), COMPUTER VISION - ACCV 2012 WORKSHOPS, SPRINGER, BERLIN, PAGE(S) 266 - 278, XP047027043, page 269, paragraph 2 - page 270, paragraph 1 -----	1-9
A	SHAHBAHRAMI ET AL.: "FPGA implementation of parallel histogram computation", PROCEEDINGS 2ND HIPEAC WORKSHOP ON RECONFIGURABLE COMPUTING, 2008, pages 63-72, XP002751308, page 66, last paragraph - page 68, paragraph 2 -----	1-9
A	NALPANTIDIS LAZAROSA ET AL: "Review of Stereo Vision Algorithms: From Software to Hardware", INTERNATIONAL JOURNAL OF OPTOMECHATRONICS, TAYLOR & FRANCIS INC, US, vol. 2, no. 4, 1 January 2008 (2008-01-01), pages 435-462, XP009187347, page 437, paragraph 3, page 454, paragraph 1 - paragraph 4 -----	1-9