

IMAGE PROCESSING DEVICE

CROSS-REFERENCE TO RELATED APPLICATIONS

5 [0001] This application claims the benefit and priority to Korean Patent Application No. 10-2018-0010183, filed on January 26, 2018, Korean Patent Application No. 10-2018-0015932, filed on February 8, 2018, and Korean Patent Application No. 10-2018-0041790, filed on April 10, 2018, in the Korean Intellectual Property Office, the disclosures of which are incorporated by reference in their entireties herein.

10 TECHNICAL FIELD

[0002] The present disclosure relates to an image processing device.

BACKGROUND

15 [0003] More and more applications demand high-resolution video images and high-frame rate images. Accordingly, the amount of data accesses to a memory (i.e., bandwidth) storing these images by various multimedia intellectual property (IP) blocks of image processing devices has greatly increased.

20 [0004] Each image processing device has limited processing capability. When the bandwidth increases, the processing capability of the image processing device may reach this limit. Accordingly, a user of the image processing device may experience a decrease in speed while recording or playing a video image.

SUMMARY

25 [0005] At least one embodiment of the present inventive concept provides an image processing device having an improved processing speed.

30 [0006] According to an exemplary embodiment of the present inventive concept, there is provided an image processing device configured to compress first image data. The image processing device includes an encoding circuit configured to compress the first image data into second image data comprising prediction data and residual data, compress the second image data into third image data by performing entropy encoding on the second image data, generate a header representing a compression ratio of the third image data, and store the third image data along with the header in a memory device as compressed first image data.

[0007] According to exemplary embodiment of the present inventive concept, there is provided an image processing device configured to compress first image data. The image processing device includes an encoding circuit having a mode selection circuit configured to determine whether a first mode is set to a lossless compression mode or a lossy compression mode based on a received signal, a first logic circuit configured to compress the first image data into second image data comprising prediction data and residual data, a second logic circuit configured to quantize the second image data using at least one quantization parameter (QP), and a third logic circuit configured to performing entropy encoding on one of i) an output of the first logic circuit when the first mode is set to the lossless compression mode and ii) an output of the second logic circuit when the first mode is set to the lossy compression mode.

[0008] According to exemplary embodiment of the present inventive concept, there is provided an image processing device configured to decompress first compressed image data. The image processing device includes a mode selection circuit configured to set a first mode to lossless decompression when the first compressed image data was compressed in a lossless manner and set the first mode to lossy decompression when the first compressed image data was compressed in a lossy manner, a first logic circuit performing an entropy decoding on the first compressed image data, a second logic circuit performing an inverse quantization on an output of the first logic circuit when the first mode is set to lossy decompression, and a third logic circuit configured to perform a decompression by adding residual data to prediction data received in an output of the first logic circuit when the first mode is set to the lossless decompression, and perform the decompression by adding residual data to prediction data received in an output of the second logic circuit when the first mode is set to lossy decompression.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present invention will become apparent by describing in detail exemplary embodiments thereof with reference to the accompanying drawings in which:

[0010] FIG. 1 is a block diagram of an image processing device according to an exemplary embodiment of the inventive concept;

[0011] FIG. 2 is a detailed block diagram of a frame buffer compressor (FBC) illustrated in FIG. 1;

[0012] FIG. 3 is a detailed block diagram of an encoder illustrated in FIG. 2;

[0013] FIG. 4 is a detailed block diagram of a decoder illustrated in FIG. 2;

[0014] FIG. 5 is a conceptual diagram for explaining three operation modes for a YUV 420 format data of the image processing device according to an exemplary embodiment of the inventive concept;

[0015] FIG. 6 is a conceptual diagram for explaining three operation modes for a YUV 422 format data of the image processing device according to an exemplary embodiment of the inventive concept;

[0016] FIG. 7 illustrates the structure of data losslessly compressed by an image processing device according to an exemplary embodiment of the present inventive concept;

[0017] FIG. 8 is a table for explaining a compression method of the losslessly compressed data of FIG. 7;

[0018] FIG. 9 illustrates the structure of data losslessly compressed by an image processing device according to an exemplary embodiment of the present inventive concept;

[0019] FIG. 10 is a table for explaining a compression method of the losslessly compressed data of FIG. 9;

[0020] FIG. 11 illustrates the structure of data losslessly compressed by an image processing device according to an exemplary embodiment of the present inventive concept;

[0021] FIG. 12 illustrates the structure of data lossy compressed by an image processing device according to an exemplary embodiment of the present inventive concept;

[0022] FIG. 13 illustrates the structure of data lossy compressed by an image processing device according to an exemplary embodiment of the present inventive concept;

[0023] FIG. 14 illustrates the structure of data lossy compressed by an image processing device according to an exemplary embodiment of the present inventive concept;

[0024] FIG. 15 illustrates the structure of data lossy compressed by an image processing device according to an exemplary embodiment of the present inventive concept;

[0025] FIG. 16 is a block diagram of an image processing device according to an exemplary embodiment of the inventive concept; and

[0026] FIG. 17 is a block diagram of an image processing device according to an exemplary embodiment of the inventive concept.

DETAILED DESCRIPTION

[0027] An image processing device according to an exemplary embodiment of the inventive concept will now be described with reference to FIGS. 1 through 14.

[0028] FIG. 1 is a block diagram of an image processing device according to at least one embodiment of the inventive concept.

[0029] Referring to FIG. 1, the image processing device according to at least one embodiment of the inventive concept includes a multimedia intellectual property (IP) 100 (e.g., an IP core, and IP block, a circuit, etc.), a frame buffer compressor (FBC) 200 (e.g., a circuit, a digital signal processor, etc.), a memory 300, and a system bus 400.

[0030] In an exemplary embodiment, the multimedia IP 100 is part of the image processing device that directly executes the image processing of the image processing device. The multimedia IP 100 may include a plurality of modules for performing image recording and reproduction such as camcording, playback, etc. of video images.

[0031] The multimedia IP 100 receives first data (e.g., image data) from an external device such as a camera and converts the first data into second data. For example, the first data may be moving raw image data or raw static image data. The second data may be data generated by the multimedia IP 100 and may also include data resulting from the multimedia IP 100 processing the first data. The multimedia IP 100 may repeatedly store the second data in the memory 300 and update the second data through various steps. The second data may include all of the data used in these steps. The second data may be stored in the memory 300 in the form of third data. Therefore, the second data may be data before being stored in the memory 300 or after being read from the memory 300.

[0032] In an exemplary embodiment, the multimedia IP 100 includes an image signal processor (ISP) 110, a shake correction module (G2D) 120, a multi-format codec (MFC) 130, a graphics processing unit (GPU) 140, and a display 150. However, the present inventive concept is not limited to this case. That is, the multimedia IP 100 can include at least one of the ISP 110, the G2D 120, the MFC 130, the GPU 140, and the display 150 described above. In other words, the multimedia IP 100 may be implemented by a processing module that has to access the memory 300 in order to process data representing a moving image or a static image.

[0033] The ISP 110 receives the first data and converts the first data into the second data by pre-processing the first data. Here, the first data may be image source data in an RGB

format. For example, the ISP 110 may convert the first data in the RGB format into the second data in a YUV format.

[0034] The RGB format refers to a data format in which colors are represented based on three primary colors of light. That is, an image is represented using three kinds of colors, i.e., red, green, and blue. On the other hand, the YUV format refers to a data format in which brightness, i.e., a luma signal and a chroma signal are represented separately. That is, Y denotes a luma signal, and U(Cb) and V(Cr) denote chroma signals, respectively. U denotes a difference between the luma signal and a blue signal component, and V denotes a difference between the luma signal and a red signal component. Here, items of Y, U (Cb), and V (Cr) may be defined as planes. For example, data about a luma signal may be referred to as data of a Y plane, and data about a chroma signal may be referred to as data of a U(Cb) plane or data of a V (Cr) plane.

[0035] Data in the YUV format may be obtained by converting data in the RGB format using a conversion formula. For example, a conversion formula such as $Y=0.3R+0.59G+0.11B$, $U=(B-Y) \times 0.493$, $V=(R-Y) \times 0.877$ may be used to convert the data of the RGB format into the data of the YUV format.

[0036] Since the human eye is sensitive to luma signals but less sensitive to color signals, it may be easier to compress data in the YUV format than to compress data in the RGB format. Therefore, the ISP 110 may convert the first data in the RGB format into the second data in the YUV format.

[0037] After the ISP 110 converts the first data into the second data, it stores the second data in the memory 300.

[0038] The G2D 120 may perform shake correction of image data or moving image data. The G2D 120 may read the first data or the second data stored in the memory 300 to perform shake correction. Here, the shake correction refers to detecting shakes of a camera in moving image data and removing the shakes.

[0039] The G2D 120 may generate new second data or update the second data by correcting shakes in the first data or the second data and may store the generated or updated second data in the memory 300.

[0040] The MFC 130 may be a codec for compressing moving image data. Generally, moving image data is very large in size. Therefore, a compression module for reducing the size of the moving image data is needed. The moving image data may be compressed

based on the associative relationship between a plurality of frames, and this compression may be performed by the MFC 130. The MFC 130 may read and compress the first data or may read and compress the second data stored in the memory 300.

[0041] The MFC 130 may generate new second data or update the second data by compressing the first data or the second data and store the new second data or the updated second data in the memory 300.

[0042] The GPU 140 may perform an arithmetic process to calculate and generate two-dimensional or three-dimensional graphics. The GPU 140 may calculate the first data or calculate the second data stored in the memory 300. The GPU 140 may be specialized in processing graphics data and may process graphics data in parallel.

[0043] The GPU 140 may generate new second data or update the second data by compressing the first data or the second data and store the new second data or the updated second data in the memory 300.

[0044] The display 150 may display the second data stored in the memory 300 on a screen. The display 150 may display image data, i.e., the second data processed by other elements of the multimedia IP 100, that is, the ISP 110, the G2D 120, the MFC 130 and the GPU 140, on the screen. However, the present inventive concept is not limited to this case.

[0045] Each of the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100 may operate individually. That is, each of the ISP 110, the G2D 120, the MFC 130, the GPU 140, and the display 150 may individually access the memory 300 to write or read data.

[0046] In an embodiment, the FBC 200 converts the second data into the third data by compressing the second data before the elements of the multimedia IP 100 access the memory 300 individually. The FBC 200 may transmit the third data to the multimedia IP 100, and the multimedia IP 100 may transmit the third data to the memory 300.

[0047] Therefore, the third data generated by the FBC 200 may be stored in the memory 300. Conversely, the third data stored in the memory 300 may be loaded by the multimedia IP 100 and transmitted to the FBC 200. The FBC 200 may convert the third data into the second data by decompressing the third data. The FBC 200 may transmit the second data to the multimedia IP 100.

[0048] That is, whenever the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100 access the memory 300 individually, the FBC 200

may compress the second data into the third data and transmit the third data to the memory 300. For example, after one of the components of the multimedia IP 100 generates and stores the second data in the memory 300, the frame buffer compressor 200 can compress the stored data and store the compressed data into the memory 300. Conversely, whenever a data request is made from the memory 300 to the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100, the FBC 200 may decompress the third data into the second data and transmit the second data to each of the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100.

[0049] The memory 300 may store the third data generated by the FBC 200 and provide the stored third data to the FBC 200 so that the FBC 200 can decompress the third data.

[0050] In an embodiment, the system bus 400 is connected to each of the multimedia IP 100 and the memory 300. Specifically, the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100 may be individually connected to the system bus 400. The system bus 400 may serve as a path through which the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100 and the memory 300 exchange data with each other.

[0051] In an embodiment, the FBC 200 is not connected to the system bus 400 and converts the second data into the third data or the third data into the second data when each of the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100 accesses the memory 300.

[0052] FIG. 2 is a detailed block diagram of the FBC 200 illustrated in FIG. 1.

[0053] Referring to FIG. 2, the FBC 200 includes an encoder 210 (e.g., an encoding circuit) and a decoder 220 (e.g., a decoding circuit).

[0054] The encoder 210 may receive the second data from the multimedia IP 100 and generate the third data. Here, the second data may be transmitted from each of the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100. The third data may be transmitted to the memory 300 through the multimedia IP 100 and the system bus 400.

[0055] Conversely, the decoder 220 may decompress the third data stored in the memory 300 into the second data. The second data may be transmitted to the multimedia IP 100. Here, the second data may be transmitted to each of the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100.

[0056] FIG. 3 is a detailed block diagram of the encoder 210 illustrated in FIG. 2.

[0057] Referring to FIG. 3, the encoder 210 includes a first mode selector 219 (e.g., a mode selection circuit), a prediction module 211 (e.g., a logic circuit or processor), a quantization module 213 (e.g., a logic circuit or processor), an entropy encoding module 215 (e.g., a logic circuit or processor), and a padding module 217 (e.g., a logic circuit or processor).

[0058] In an embodiment, the first mode selector 219 determines whether the encoder 210 will operate in a lossless mode (e.g., lossless compression) or a lossy mode (e.g., lossy compression). When the encoder 210 operates in the lossless mode based on the determination result of the first mode selector 219, the second data may be compressed along a lossless path of FIG. 3. When the encoder 210 operates in the lossy mode, the second data may be compressed along a lossy path.

[0059] The first mode selector 219 may receive a signal from the multimedia IP 100 used to determine whether lossless compression or lossy compression will be performed. Here, the lossless compression denotes compression without loss of data and has a compression ratio that varies depending on data. On the other hand, the lossy compression denotes compression in which data is partially lost. The lossy compression has a higher compression ratio than the lossless compression and has a preset fixed compression ratio.

[0060] In the case of the lossless mode, the first mode selector 219 enables the second data to flow to the prediction module 211, the entropy encoding module 215 and the padding module 217 along the lossless path. Conversely, in the case of the lossy mode, the first mode selector 219 enables the second data to flow to the prediction module 211, the quantization module 213 and the entropy encoding module 215 along the lossy path.

[0061] The prediction module 211 converts the second data into predicted image data. The predicted image data is a compressed representation of the second data as a combination of prediction data and residual data. In an embodiment, the prediction data is image data of one pixel of the image data and the residual data is created from the differences between the prediction data and the image data of the pixels of the image data that are adjacent the one pixel. For example, if the image data of the one pixel has a value of 0 to 255, 8 bits may be needed to represent the value. When the adjacent pixels have similar values to that of the one pixel, the residual data of each of the adjacent pixels is much smaller than the prediction data. For example, if an adjacent pixel has a similar value, only a difference (i.e., a residual) from the value of the adjacent pixel can be represented

without loss of data, and the number of bits of data needed to represent the difference may be much smaller than 8 bits. For example, when pixels having values of 253, 254 and 255 are arranged successively, if prediction data is 253, a residual data representation of (253(prediction), 1(residual), 2(residual)) may be sufficient, and the number of bits per pixel
5 needed for this residual data representation may be 2 bits which is much smaller than 8 bits. For example, 24bits of data of 253, 254, and 255 can be reduced to 12bits due to 8bit prediction data of 253 (11111101), 2 bit residual data of $254-253 = 1$ (01), and 2 bit residual data of $255-253=2$ (10).

[0062] Therefore, the prediction module 211 may compress the overall size of the second data by dividing the second data into prediction data and residual data. Various methods
10 can be used to determine the prediction data.

[0063] The prediction module 211 may perform prediction on a pixel-by-pixel basis or on a block-by-block basis. Here, a block may be an area formed by a plurality of adjacent pixels. For example, prediction on a pixel basis could mean that all the residual data is
15 created from one of the pixels, and prediction on the block basis could mean that residual data is created for each block from a pixel of the corresponding block.

[0064] The quantization module 213 may further compress the second data compressed by the prediction module 211. The quantization module 213 may remove lower bits of the second data using a preset quantization parameter (QP) (e.g., quantizing the second data
20 using the QP). For example, if the prediction data is 253 (11111101), the prediction data can be reduced from 8 bits to 6 bits by removing the lower 2 bits, which results in prediction data of 252 (111111). Specifically, a representative value may be selected by multiplying the data by a QP, where numbers below a decimal point are discarded, thus causing a loss. If pixel data has a value of 0 to 2^8-1 (= 255), the QP may be defined as $1/(2^n-1)$ (where n is
25 an integer of 8 or less). However, the current embodiments are not limited to this case.

[0065] Here, since the removed lower bits are not restored later, they are lost. Therefore, the quantization module 213 is utilized only in the lossy mode. The lossy mode may have a relatively higher compression ratio than the lossless mode and may have a preset fixed compression ratio. Therefore, information about the compression ratio is not needed later.

[0066] The entropy encoding module 215 may compress, through entropy coding, the
30 second data compressed by the quantization module 213 in the lossy mode or the second

data compressed by the prediction module 211 in the lossless mode. In the entropy coding, the number of bits may be allocated according to frequency.

[0067] In an embodiment, the entropy encoding module 215 compresses the second data using Huffman coding. In an alternative embodiment, the entropy encoding module 215 compresses the predicted image data through exponential Golomb coding or Golomb rice coding. In an embodiment, the entropy encoding module 215 generates a table using k values, it compresses the predicted image data using the generated table. The k values may be entropy encoding values using in the entropy encoding.

[0068] The padding module 217 may perform padding on the second data compressed by the entropy encoding module 215 in the lossless mode. Here, the padding may refer to adding meaningless data in order to fit a specific size. This will be described in more detail later.

[0069] The padding module 217 may be activated not only in the lossless mode but also in the lossy mode. In the lossy mode, the second data may be compressed by the quantization module 213 more than an intended compression ratio. In this case, the second data may be passed through the padding module 217 even in the lossy mode, converted into the third data and then transmitted to the memory 300. In an exemplary embodiment, the padding module 217 is omitted so that no padding is performed.

[0070] A compress manager 218 determines a combination of a QP table and an entropy table used for quantization and entropy coding, respectively, and controls the compression of the second data according to the determined combination of the QP table and the entropy table.

[0071] In this case, the first mode selector 219 determines that the encoder 210 will operate in the lossy mode. Accordingly, the second data is compressed along the lossy path of FIG. 3. That is, the compress manager 218 determines that a combination of a QP table and an entropy table is needed and compresses the second data according to the determined combination of the QP table and the entropy table based on the premise that the FBC 200 compresses the second data using a lossy compression algorithm.

[0072] Specifically, the QP table may include one or more entries, and each of the entries may include a QP used to quantize the second data.

[0073] In an embodiment, the entropy table refers to a plurality of code tables identified by k values to perform an entropy coding algorithm. The entropy table that can be used in

some embodiments may include at least one of an exponential Golomb code and a Golomb rice code.

[0074] The compress manager 218 determines a QP table including a predetermined number of entries, and the FBC 200 quantizes predicted the second data using the determined QP table. In addition, the compress manager 218 determines an entropy table using a predetermined number of k values, and the FBC 200 performs entropy coding on the quantized second data using the determined entropy table. That is, the FBC 200 generates the third data based on a combination of a QP table and an entropy table determined by the compress manager 218.

[0075] Then, the FBC 200 may write the generated third data into the memory 300. In addition, the FBC 200 may read the third data from the memory 300, decompress the read third data, and provide the decompressed third data to the multimedia IP 100.

[0076] FIG. 4 is a detailed block diagram of the decoder 220 illustrated in FIG. 2.

[0077] Referring to FIGS. 3 and 4, the decoder 220 includes a second mode selector 229 (e.g., a mode selection circuit), an unpadding module 227 (e.g., a logic circuit or processor), an entropy decoding module 225 (e.g., a logic circuit), an inverse quantization module 223 (e.g., a logic circuit or processor), and a prediction compensation module 221 (e.g., a logic circuit or processor).

[0078] The second mode selector 229 determines whether the third data stored in the memory 300 has been generated through lossless compression or lossy compression of the second data. In an exemplary embodiment, the second mode selector 229 determines whether the third data has been generated by compressing the second data in the lossless mode or the lossy mode based on the presence or absence of a header. In an embodiment, the third data was compressed in a lossy mode when the third data includes at least one quantization parameter, and was compressed in a lossless mode when the third data includes no quantization parameters. In another embodiment, the compressed data is stored with a flag indicating whether the data was compressed in a lossy or lossless manner.

[0079] If the third data has been generated by compressing the second data in the lossless mode, the second mode selector 229 enables the third data to flow to the unpadding module 227, the entropy decoding module 225 and the prediction compensation module 221 along a lossless path. Conversely, if the third data has been generated by compressing the second data in the lossy mode, the second mode selector 229 enables the third data to flow to the

entropy decoding module 225, the inverse quantization module 223 and the prediction compensation module 221 along a lossy path.

[0080] The unpadding module 227 may remove a portion of data which has been padded by the padding module 217 of the encoder 210. The unpadding module 227 may be omitted when the padding module 217 is omitted.

[0081] The entropy decoding module 225 may decompress data which has been compressed by the entropy encoding module 215. The entropy decoding module 225 may perform decompression using Huffman coding, exponential Golomb coding, or Golomb rice coding. Since the third data includes k values, the entropy decoding module 225 may perform decoding using the k values.

[0082] The inverse quantization module 223 may decompress data which has been compressed by the quantization module 213. The inverse quantization module 223 may restore the second data compressed by the quantization module 213 using a predetermined quantization parameter (QP). For example, the inverse quantization module 223 may perform an inverse quantization operation on the output of the entropy decoding module 225. However, the inverse quantization module 223 cannot completely recover the data lost in the compression process. Therefore, the inverse quantization module 223 is only in the lossy mode.

[0083] The prediction compensation module 221 may perform prediction compensation to recover data represented as prediction data and residual data by the prediction module 211. For example, the prediction compensation module 221 may convert a residual data representation of (253(prediction), 1(residual), 2(residual)) into (253, 254, 255). For example, the prediction compensation module 221 may restore the data by adding the residual data to the prediction data.

[0084] The prediction compensation module 221 may restore data predicted on a pixel-by-pixel basis or a block-by-block basis by the prediction module 211. Accordingly, the second data may be restored or the third data may be decompressed and then transmitted to the multimedia IP 100.

[0085] A decompress manager 228 may perform an operation to appropriately reflect a combination of a QP table and an entropy table, which has been determined by the compress manager 218 to compress the second data as described above with reference to FIG. 3, in the decompression of the third data.

[0086] In an exemplary embodiment, the second data of the image processing device is data in the YUV format. Here, the data in the YUV format may have a YUV 420 format or a YUV 422 format.

[0087] FIG. 5 is a conceptual diagram for explaining three operation modes for the YUV 420 data format processed by an image processing device according to an exemplary embodiment of the inventive concept.

[0088] Referring to FIGS. 1 through 5, the encoder 210 and the decoder 220 of the FBC 200 may have three operation modes. In FIG. 5, the second data in the YUV 420 format has a 16×16 luma signal block Y, an 8×8 first chroma signal block Cb or U, and an 8×8 second chroma signal block Cr or V. Here, the size of each block denotes how many rows and columns of pixels are arranged, and the 16×16 size denotes the size of a block composed of 16 rows and 16 columns of pixels.

[0089] The FBC 200 may include three operation modes: a concatenation mode ①, a partial concatenation mode ②, and a separation mode ③. These three modes are about compression formats of data and may be operation modes determined separately from the lossy mode and the lossless mode. In an exemplary embodiment, only two of the three operation modes are available. In a further exemplary embodiment, only one of three operation modes are available.

[0090] The concatenation mode ① is an operation mode for compressing and decompressing the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr together. That is, in the concatenation mode ①, a unit block of compression is a block in which the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr are combined as illustrated in FIG. 5. Accordingly, the unit block of compression may have a size of 16×24. For example, a single compression operation may be used to compress a block of data including the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr.

[0091] In the partial concatenation mode ②, the first chroma signal block Cb and the second chroma signal block Cr are combined with each other and compressed and decompressed together, and the luma signal block Y is separately compressed and decompressed. Accordingly, the luma signal block Y may have its original size of 16×16, and a block in which the first chroma signal block Cb and the second chroma signal block Cr are combined may have a size of 16×8. For example, a first compression operation may

be used to compress the luma signal block Y and a second compression operation may be used to compress a block of data including the first chroma signal block Cb and the second chroma signal block Cr.

[0092] The separation mode ③ is an operation mode for separately compressing and decompressing the luma signal block Y, the first chroma signal block Cb, and the second chroma signal block Cr. Here, in order to make unit blocks of compression and decompression equal in size, the luma signal block Y is maintained at its original size of 16x16, and the first chroma signal block Cb and the second chroma signal block Cr are each enlarged to a size of 16x16. For example, a first compression operation may be used to compress the luma signal block Y, a second compression operation may be used to compress the first chroma signal block Cb, and a third compression operation may be used to compress the second chroma signal block Cr.

[0093] Accordingly, if the number of the luma signal blocks Y is N, the number of the first chroma signal blocks Cb and the number of the second chroma signal blocks Cr may each be reduced to N/4.

[0094] When the FBC 200 of the image processing device according to an exemplary embodiment of the inventive concept operates in the concatenation mode ①, all the necessary data can be read through a single access request to the memory 300. In particular, when the multimedia IP 100 needs data in the RGB format, not in the YUV format, it is advantageous for the FBC 200 to operate in the concatenation mode ①. This is because all of the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr can be obtained at a time in the concatenation mode ① and because all of the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr are needed to obtain RGB data.

[0095] The separation mode ③ may require less hardware resources when the unit block of compression is smaller in the concatenation mode ①. Therefore, the separation mode ③ may be more advantageous when the multimedia IP 100 needs data in the YUV format, not in the RGB format.

[0096] Finally, the partial concatenation mode ② is a compromise between the concatenation mode ① and the separation mode ③. The partial concatenation mode ② requires less hardware resources than the concatenation mode ① and makes less access

requests (two access requests) to the memory 300 than the separation mode ③ even when RGB data is needed.

[0097] The first mode selector 219 determines in which of the three modes, i.e., the concatenation mode ①, the partial concatenation mode ② and the separation mode ③ the second data will be compressed. The first mode selector 219 may receive from the multimedia IP 100 a signal indicating which of the concatenation mode ①, the partial concatenation mode ② and the separation mode ③ will be executed. For example, the signal may be set to a first voltage level to indicate the concatenation mode, a second voltage level to indicate the partial concatenation mode, and a third voltage level to indicate the separate mode, where the first through third voltage levels differ from one another.

[0098] The second mode selector 229 may decompress the third data according to which of the concatenation mode ①, the partial concatenation mode ② and the separation mode ③ the second data has been compressed into the third data. For example, if the second data was compressed using the concatenation mode, the second mode selector 229 can make a single access to retrieve compressed data including the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr. For example, if the second data was compressed using the partial concatenation mode, the second mode selector 229 can make two accesses to retrieve first compressed data including the luma signal block Y and second compressed data including the first chroma signal block Cb and the second chroma signal block Cr. For example, if the second data was compressed using the separation mode, the second mode selector 229 can make three accesses to retrieve first compressed data including the luma signal block Y, second compressed data including the first chroma signal block Cb and third compressed data including the second chroma signal block Cr.

[0099] FIG. 6 is a conceptual diagram for explaining three operation modes for the YUV 422 data format of the image processing device according to an exemplary embodiment of the inventive concept.

[00100] Referring to FIGS. 1 through 4 and 6, the encoder 210 and the decoder 220 of the FBC 200 may also have three operation modes for the YUV 422 format. In FIG. 6, the second data in the YUV 422 format has a 16x16 luma signal block Y and a 16x8 first chroma signal block Cb or U and a 16x8 second chroma signal block Cr or V.

[00101] In a concatenation mode ①, a unit block of compression is a block in which the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr are combined. Accordingly, the unit block of compression may have a size of 16x32.

[00102] In a partial concatenation mode ②, the first chroma signal block Cb and the second chroma signal block Cr are combined with each other and compressed and decompressed together, and the luma signal block Y is separately compressed and decompressed. Accordingly, the luma signal block Y may have its original size of 16x16, and a block in which the first chroma signal block Cb and the second chroma signal block Cr are combined may have a size of 16x16. Therefore, the size of the luma signal block Y may be equal to the size of the block in which the first chroma signal block Cb and the second chroma signal block Cr are combined.

[00103] A separation mode ③ is an operation mode for separately compressing and decompressing the luma signal block Y, the first chroma signal block Cb, and the second chroma signal block Cr. Here, in order to make unit blocks of compression and decompression equal in size, the luma signal block Y is maintained at its original size of 16x16, and the first chroma signal block Cb and the second chroma signal block Cr are each enlarged to a size of 16x16.

[00104] Accordingly, if the number of the luma signal blocks Y is N, the number of the first chroma signal blocks Cb and the number of the second chroma signal blocks Cr are each reduced to N/2.

[00105] FIG. 7 illustrates the structure of data losslessly compressed by an image processing device according to an exemplary embodiment of the present inventive concept. FIG. 8 is a table for explaining a compression method of the losslessly compressed data of FIG. 7.

[00106] Referring to FIGS. 1 through 8, the third data (i.e., the compressed data) includes a payload and a header.

[00107] The header indicates a compression ratio, and the payload includes actual compressed data and values necessary for decompression.

[00108] In FIG. 8, a table for explaining lossless compression of a 16x16 block is illustrated by way of example. Since the data format is YUV 420 and the operation mode is the separation mode ③, the luma signal block Y, the first chroma signal block Cb or the second chroma signal block Cr of FIG. 5 may correspond to this table. A pixel depth

denotes a bit value of a value represented in one pixel. For example, a pixel depth of 8 bits is required to represent a value of 0 to 255. Thus, in the example of FIG. 8, a value represented in each pixel may be 0 to 255.

[00109] In the memory 300, the size of data that can be accessed at a time hardware-wise is predetermined. The size of a data access unit of the memory 300 may denote the size of data that can be accessed in the memory 300. In FIG. 8, it will be assumed for convenience that the size of the data access unit of the memory 300 is 32 bytes.

[00110] One pixel may have data of 8 bits, i.e., 1 byte, and a 16x16 block may have data of a total of 256 bytes. That is, the size of the second data (i.e., the uncompressed data) may be 256 bytes.

[00111] In the case of lossless compression, the size of compressed data may be different each time. In order to read the compressed data from the memory 300, the size of the compressed data must be recorded separately. However, if the size of the compressed data is recorded as it is, the compression efficiency may be reduced by the recorded size. Therefore, the compression ratio may be standardized to improve the compression efficiency.

[00112] Specifically, in FIG. 8, the range of the compression ratio is defined based on 32 bytes which is the size of the data access unit of the memory 300. That is, if the size of the compressed data is 0 to 32 bytes, the compression ratio is 100 to 87.5%. In this case, an operation of adjusting the compression ratio to 87.5% (i.e., an operation of adjusting the size of the compressed data to 32 bytes) may be performed, and 0 may be recorded in the header. Likewise, if the size of the compressed data is 161 to 192 bytes, the compression ratio is 37.5 to 25%. In this case, an operation of adjusting the compression ratio to 25% (i.e., an operation of adjusting the size of the compressed data to 192 bytes) may be performed, and 5 may be recorded in the header.

[00113] The padding module 217 of FIG. 3 may perform the operation of adjusting the size of the compressed data to a maximum size of a corresponding range. That is, if the size of the compressed data is 170 bytes, since 170 bytes is between 161 bytes and 192 bytes, a padding operation of adding "0" of 22 bytes may be performed to adjust the size of the compressed data to 192 bytes.

[00114] The compressed data whose size has been adjusted to a maximum size of a corresponding range by the padding module 217 may become the payload of the third data.

Accordingly, the size (n1 bits) of the payload may be an integer multiple of the size of the data access unit of the memory 300.

[00115] The header may include a header index of FIG. 8. The size of the header may vary depending on the size of the compressed data, but may be 3 bits in the case of FIG. 8 because the header represents only 0 to 7.

[00116] In an exemplary embodiment, the header and the payload are stored in different areas of the memory 300. That is, the header may be stored adjacent to another header, and the payload may be stored adjacent to another payload.

[00117] In an embodiment, the payload includes a binary code and a k value code. The binary code may represent the compressed second data. The k value code may represent a k value determined by the entropy encoding module 215.

[00118] The binary code may include data values of the entire block. Therefore, the binary code may successively include data of each pixel included in the block.

[00119] Since the current mode is the separation mode ③, the k value code is a k value for any one of the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr.

[00120] FIG. 9 illustrates the structure of data losslessly compressed by an image processing device according to an exemplary embodiment of the present inventive concept. FIG. 10 is a table for explaining a compression method of the losslessly compressed data of FIG. 9.

[00121] In FIG. 10, a table for explaining lossless compression of a 16x8 block is illustrated by way of example. Since the data format is YUV 420 and the operation mode is the partial concatenation mode ②, a block in which the first chroma signal block Cb and the second chroma signal block Cr of FIG. 5 are combined may correspond to this table. In FIG. 10, it will be assumed for convenience that the size of the data access unit of the memory 300 is 32 bytes.

[00122] The 16x8 block may have data of a total of 128 bytes. That is, the size of the second data may be 128 bytes.

[00123] The range of the compression ratio is defined based on 32 bytes which is the size of the data access unit of the memory 300. That is, if the size of the compressed data is 0 to 32 bytes, the compression ratio is 100 to 75%. In this case, an operation of adjusting the compression ratio to 75% (i.e., an operation of adjusting the size of the compressed data to

32 bytes) may be performed, and 0 may be recorded in the header. Likewise, if the size of the compressed data is 97 to 128 bytes, the compression ratio is 25 to 0%. In this case, an operation of adjusting the compression ratio to 0% (i.e., an operation of adjusting the size of the compressed data to 128 bytes) may be performed, and 3 may be recorded in the header.

5 This operation may be performed by the padding module 217 of FIG. 3.

[00124] Referring to FIG. 9, the compressed data whose size has been adjusted to a maximum size of a corresponding range by the padding module 217 may become the payload of the third data. Accordingly, the size (n2 bits) of the payload may be an integer multiple of the size of the data access unit of the memory 300.

10 [00125] In an embodiment, the payload includes a binary code and a k value code. Since the current mode is the partial concatenation mode ② and the 16x8 block is a block in which the first chroma signal block Cb and the second chroma signal block Cr are combined, the k value code includes a k value code of the first chroma signal block Cb and a k value code of the second chroma signal block Cr. The arrangement order of the k value code of
15 the first chroma signal block Cb and the k value code of the second chroma signal block Cr can vary.

[00126] A payload of an image processing device according to an exemplary embodiment includes only one k value code without having the k value code of the first chroma signal block Cb and the k value code of the second chroma signal block Cr separately. In this
20 case, the k value code is the same in the first chroma signal block Cb and the second chroma signal block Cr.

[00127] Since FIG. 9 illustrates the data structure of the block in which the first chroma signal block Cb and the second chroma signal block Cr are combined in the partial concatenation mode ②, the binary code may include both data about the first chroma signal
25 block Cb and data about the second chroma signal block Cr.

[00128] In the binary code, the data about the first chroma signal block Cb may be placed first for all pixels, and then the data about the second chroma signal block Cr may be placed for all pixels. That is, the data of the first chroma signal block Cb and the data of the second chroma signal block Cr may be arranged separately. The arrangement order of the
30 data of the first chroma signal block Cb and the data of the second chroma signal block Cr can vary.

[00129] Alternatively, a binary code of an image processing device according to an exemplary embodiment has an interleaving structure in which the data of the first chroma signal block Cb and the data of the second chroma signal block Cr are arranged successively for one pixel and the data of the first chroma signal block Cb and the data of the second chroma signal block Cr are successively arranged for another pixel.

[00130] This structure does not need to be considered in the separation mode ③ because only one plane is included in the separation mode ③. However, since a plurality of planes are included in the concatenation mode ① and the partial concatenation modes ②, the interleaving structure can be used.

[00131] Therefore, in the concatenation mode ①, the data of the luma signal block Y for all pixels, the data of the first chroma signal block Cb for all pixels, and the data of the second chroma signal block Cr may also be arranged separately in the binary code. The arrangement order of the data of the luma signal block Y, the data of the first chroma signal block Cb, and the data of the second chroma signal block Cr can vary.

[00132] Alternatively, in the binary code of the concatenation mode ①, the data of the luma signal block Y, the data of the first chroma signal block Cb, and the data of the second chroma signal block Cr may be arranged for one pixel. Then, the data of the luma signal block Y, the data of the first chroma signal block Cb, and the data of the second chroma signal block Cr may be arranged for another pixel.

[00133] That is, in the binary code, data may be arranged for each plane or may be arranged for each pixel (the interleaving structure).

[00134] FIG. 11 illustrates the structure of data losslessly compressed by an image processing device according to an exemplary embodiment of the present inventive concept.

[00135] Referring to FIG. 11, if the current mode is the concatenation mode ①, a block in which the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr are combined is compressed. Therefore, a k value code may include a k value code of the luma signal block Y, a k value code of the first chroma signal block Cb, and a k value code of the second chroma signal block Cr. Here, the arrangement order of the k value code of the luma signal block Y, the k value code of the first chroma signal block Cb, and the k value code of the second chroma signal block Cr can vary.

[00136] A payload of an image processing device according to an embodiment includes only one k value code without having the k value code of the luma signal block Y, the k

value code of the first chroma signal block Cb and the k value code of the second chroma signal block Cr separately. In this case, the k value code is the same in the luma signal block Y, the first chroma signal block Cb, and the second chroma signal block Cr.

[00137] Alternatively, if only two of the three planes share the same k value code, a total of two k value codes may be included in the payload.

[00138] In an embodiment, the size ($n3$ bits) of the payload including the binary code and the k value code is an integer multiple of the size of the data access unit of the memory 300.

[00139] FIG. 12 illustrates the structure of data lossy compressed by an image processing device according to an exemplary embodiment of the present inventive concept.

[00140] Referring to FIG. 12, the third data generated by compressing the second data in the lossy mode includes only a payload without a header.

[00141] In an embodiment, the payload includes a binary code, a k value code, and a QP code. In an embodiment, the QP code includes a QP used by the quantization module 213 in FIG. 3. In an embodiment, the inverse quantization module 223 of the decoder 220 decompresses data compressed by the quantization module 213 using the QP code.

[00142] The third data of FIG. 12 may be the structure of the third data corresponding to the separation mode ③. Therefore, the k value code and the QP code may be present only for one of the luma signal block Y, the first chroma signal block Cb, and the second chroma signal block Cr.

[00143] Since the compression ratio is fixed in the lossy mode, the size of the third data, that is, the size ($m1$ bits) of the payload may be fixed.

[00144] FIG. 13 illustrates the structure of data lossy compressed by an image processing device according to an exemplary embodiment of the present inventive concept.

[00145] Referring to FIG. 13, a payload includes a binary code, a k value code, and a QP code. The third data of FIG. 13 may be the structure of the third data of a block in which the first chroma signal block Cb and the second chroma signal block Cr are combined in the partial concatenation mode ②. Therefore, there are two k value codes and two QP codes. Specifically, the payload includes a k value code of the first chroma signal block Cb, a k value code of the second chroma signal block Cr, a QP code of the first chroma signal block Cb and a QP code of the second chroma signal block Cr.

[00146] The order of the k value code of the first chroma signal block Cb, the k value code of the second chroma signal block Cr, the QP code of the first chroma signal block Cb, the

QP code of the second chroma signal block Cr, and the binary code (i.e., the compressed data) can vary.

[00147] In an embodiment, the two k value codes are shared. Therefore, when the same k value code is shared, there is only one k value code.

5 [00148] Likewise, the two QP codes may be shared. Therefore, when the same QP code is shared, there is only one QP code. FIG. 14 illustrates the structure of data lossy compressed by an image processing device according to an exemplary embodiment of the present inventive concept.

10 [00149] Referring to FIG. 14, a payload includes a binary code, a k value code, and a QP code. The third data of FIG. 14 may be the structure of the third data of a block in which the luma signal block Y, the first chroma signal block Cb and the second chroma signal block Cr are combined in the concatenation mode ①. Therefore, there are three k value codes and three QP codes. Specifically, the payload includes a k value code of the luma signal block Y, a k value code of the first chroma signal block Cb, a k value code of the
15 second chroma signal block Cr, a QP code of the luma signal block Y, a QP code of the first chroma signal block Cb, and a QP code of the second chroma signal block Cr.

[00150] The order of the k value code of the luma signal block Y, the k value code of the first chroma signal block Cb, the k value code of the second chroma signal block Cr, the QP code of the luma signal block Y, the QP code of the first chroma signal block Cb, the QP
20 code of the second chroma signal block Cr, and the binary code can vary.

[00151] In an embodiment, the three k value codes are shared. Therefore, when the same k value is shared, there is only one k value code. When two of the three planes share a k value code, there is only two k value codes.

25 [00152] Likewise, the three QP codes may be shared. Therefore, when the same QP code is shared, there is only one QP code. When two of the three planes share a QP code, there is only two QP codes.

[00153] An image processing device according to an exemplary embodiment of the inventive concept will now be described with reference to FIGS. 1 through 6 and 15.

30 [00154] FIG. 15 illustrates the structure of data lossy compressed by an image processing device according to an exemplary embodiment of the present inventive concept.

[00155] Referring to FIGS. 1 through 6 and 15, in the lossy mode, a payload may include only a binary code and a maximum QP code without a k value code. Here, a QP stored in

the maximum QP code may be a maximum value from among possible QPs. If pixel data has a value of 0 to 2^8-1 ($= 255$), the QP may be defined as $1/(2^n-1)$ (where n is an integer of 8 or less). However, the current embodiment is not limited to this case.

[00156] In an embodiment, an entropy encoding module 215 performs entropy coding by allocating bits according to the frequency of similar data. If the QP is the maximum value, the frequency of similar data is not high. Therefore, performing entropy encoding can increase data size.

[00157] Hence, the second data of the image processing device according to an embodiment of the inventive concept is directly converted into the third data by a quantization module 213 of an encoder 210 and does not pass through the entropy encoding module 215.

[00158] Therefore, when the QP of the quantization module 213 has the maximum value, the encoder 210 of the image processing device according to an embodiment does not include the k value code in the payload and includes only the binary code and the maximum QP code in the payload.

[00159] An image processing device according to an exemplary embodiment will now be described with reference to FIG. 16.

[00160] FIG. 16 is a block diagram of an image processing device according to an exemplary embodiment of the inventive concept.

[00161] Referring to FIG. 16, an FBC 200 of the image processing device according to an exemplary embodiment of the inventive concept is directly connected to a system bus 400.

[00162] The FBC 200 is not directly connected to a multimedia IP 100 but is connected to the multimedia IP 100 through the system bus 400. Specifically, each of an ISP 110, a G2D 120, an MFC 130, a GPU 140 and a display 150 of the multimedia IP 100 may exchange data with the FBC 200 through the system bus 400 and transmit data to a memory 300 through the system bus 400.

[00163] That is, in a compression operation, each of the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100 may transmit the second data (i.e., the uncompressed data) to the FBC 200 through the system bus 400. Then, the FBC 200 may compress the second data into the third data and transmit the third data (i.e., the compressed data) to the memory 300 through the system bus 400.

[00164] Similarly, in a decompression operation, the FBC 200 may receive the third data stored in the memory 300 through the system bus 400 and decompress the third data into the second data. Then, the FBC 200 may transmit the second data to each of the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100 through the system bus 400.

[00165] In the embodiment of FIG. 16, although the FBC 200 is not directly connected to the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100, it can indirectly connected to the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100 through the system bus 400. Therefore, hardware configuration can be simplified, and operation speed can be improved.

[00166] An image processing device according to an exemplary embodiment of the inventive concept will now be described with reference to FIG. 17.

[00167] FIG. 17 is a block diagram of an image processing device according to an exemplary embodiment of the inventive concept.

[00168] Referring to FIG. 17, the image processing device according to the embodiments is configured such that a memory 300 and a system bus 400 are connected to each other through an FBC 200.

[00169] That is, the memory 300 is not directly connected to the system bus 400 and is connected to the system bus 400 only through the FBC 200. In addition, an ISP 110, a G2D 120, an MFC 130, a GPU 140 and a display 150 of a multimedia IP 100 may be directly connected to the system bus 400. Therefore, the ISP 110, the G2D 120, the MFC 130, the GPU 140 and the display 150 of the multimedia IP 100 can access the memory 300 only through the FBC 200.

[00170] Since the FBC 200 is involved in all accesses to the memory 300 in the current embodiment, the FBC 200 may be directly connected to the system bus 400, and the memory 300 may be connected to the system bus 400 through the FBC 200. This can reduce errors in data transmission and improve operation speed.

WHAT IS CLAIMED IS:

1. An image processing device configured to compress first image data, the image processing device comprising:

an encoding circuit configured to compress the first image data into second image data comprising prediction data and residual data, compress the second image data into third image data by performing entropy encoding on the second image data, generate a header representing a compression ratio of the third image data, and store the third image data along with the header in a memory device as compressed first image data.

2. The image processing device of claim 1, further comprising:
an intellectual property (IP) core directly connected to the encoding circuit; and
a data bus directly connected to the IP core and the memory device,
wherein the encoding circuit receives the first image data directly from the IP core, the encoding circuit transmits the compressed first image data directly to the IP core, and the IP core transmits the compressed first image data across the data bus for storage in the memory device.

3. The image processing device of claim 1, further comprising:
an intellectual property (IP) core; and
a data bus directly connected to the IP core, the encoding circuit, and the memory device,
wherein the encoding circuit receives the first image data indirectly from the IP core through the data bus and the encoding circuit transmits the compressed first image data indirectly to the memory device through the data bus.

4. The image processing device of claim 1, further comprising:
an intellectual property (IP) core; and
a data bus directly connected to the IP core and the encoding circuit,
wherein the encoding circuit receives the first image data indirectly from the IP core through the data bus and the encoding circuit transmits the compressed first image data directly to the memory device.

5. The image processing device of any one of claims 1 to 4, wherein the header includes an index corresponding to one compression ratio range among a plurality of available compression ratio ranges, and the compression ratio of the third image data fits within the one compression range.

5

6. The image processing device of any one of claims 1 to 5, wherein the encoding circuit adjusts a size of the first compressed data to a size of a data access unit of the memory device.

10

7. The image processing device of any one of claims 1 to 6, wherein the compressed first image data further includes at least one entropy encoding value used in the entropy encoding.

15

8. The image processing device of claim 7, wherein the entropy encoding comprises quantizing the second image data using at least one quantization parameter (QP) and performing entropy encoding on a result of the quantizing.

20

9. The image processing device of claim 8, wherein the compressed first image data further includes the at least one QP.

25

10. The image processing device of any one of claims 1 to 9, wherein the encoding circuit comprises a mode selection circuit configured to receive information indicating that a compression mode is one of i) a concatenation mode, ii) a partial concatenation mode, and iii) a separation mode,

wherein during the concatenation mode, the encoding circuit compresses the first image data by combining luma data, first chroma data, and second chroma data into first combined data and compressing the first combined data,

30

wherein during the partial concatenation mode, the encoding circuit compresses the first image data by compressing the luma data, combining the first chroma data and the second chroma data into second combined data and compressing the second combined data, and

wherein during the separation mode, the encoding circuit compresses the first image data by separately compressing the luma data, the first chroma data, and the second chroma data.

5 11. An image processing device configured to compress first image data, the image processing device comprising:

an encoding circuit comprising:

10 a mode selection circuit configured to determine whether a first mode is set to a lossless compression mode or a lossy compression mode based on a received signal;

 a first logic circuit configured to compress the first image data into second image data comprising prediction data and residual data;

 a second logic circuit configured to quantize the second image data using at least one quantization parameter (QP);

15 a third logic circuit configured to performing entropy encoding on one of i) an output of the first logic circuit when the first mode is set to the lossless compression mode and ii) an output of the second logic circuit when the first mode is set to the lossy compression mode.

20 12. The image processing device of claim 11, further comprising:

an intellectual property (IP) core directly connected to the encoding circuit; and

a data bus directly connected to the IP core and the memory device,

25 wherein the encoding circuit receives the first image data directly from the IP core, the encoding circuit transmits compressed first image data generated based on an output of the third logic circuit directly to the IP core, and the IP core transmits the compressed first image data across the data bus for storage in the memory device.

13. The image processing device of claim 11, further comprising:

an intellectual property (IP) core; and

30 a data bus directly connected to the IP core, the encoding circuit, and the memory device,

wherein the encoding circuit receives the first image data indirectly from the IP core through the data bus and the encoding circuit transmits compressed first image data generated based on an output of the third logic circuit indirectly to the memory device through the data bus.

5

14. The image processing device of claim 11, further comprising:
an intellectual property (IP) core; and
a data bus directly connected to the IP core and the encoding circuit,

10 wherein the encoding circuit receives the first image data indirectly from the IP core through the data bus and the encoding circuit transmits compressed first image data generated based on an output of the third logic circuit directly to the memory device.

15 15. The image processing device of any one of claims 11 to 14, wherein the encoding circuit determines a compression ratio of data output by the third logic circuit, generates a header representing the compression ratio, and stores the output of the third logic circuit along with the header in a memory device as compressed first image data.

20 16. The image processing device of claim 15, wherein the header includes an index corresponding to one compression ratio range among a plurality of available compression ratio ranges, and the compression ratio fits within the one compression range.

25 17. The image processing device of any one of claims 11 to 16, wherein encoding circuit adjusts a size of the first compressed data to a size of a data access unit of the memory device.

18. The image processing device of any one of claims 15 to 17, wherein the compressed first image data further includes at least one entropy encoding value using the entropy encoding.

30 19. The image processing device of any one of claims 15 to 18, wherein the compressed first image data further includes the at least one QP when the first mode is set to the lossy compression mode.

20. The image processing device of any one of claims 11 to 19, wherein the mode selection circuit is configured to determine whether a second mode is set to one of i) a concatenation mode, ii) a partial concatenation mode, and iii) a separation mode,

5 wherein during the concatenation mode, the first logic circuit compresses the first image data by combining luma data, first chroma data, and second chroma data into first combined data and compressing the first combined data,

10 wherein during the partial concatenation mode, the first logic circuit compresses the first image data by compressing the luma data, combining the first chroma data and the second chroma data into second combined data and compressing the second combined data, and

15 wherein during the separation mode, the first logic circuit compresses the first image data by separately compressing the luma data, the first chroma data, and the second chroma data.