US 20080232462A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2008/0232462 A1**

(43) **Pub. Date: Sep. 25, 2008**

Van Doren et al.

(54) **METHOD VIDEO ENCODING AND DECODING PRESERVING CAHE LOCALITIES**

(75) Inventors: **Giel Van Doren**, Lommel (BE); **Abraham Karel Riemens**, Eersel (NL); **Frederik Jan De Bruijn**, Eindhoven (NL)

Correspondence Address:
NXP, B.V.
NXP INTELLECTUAL PROPERTY DEPARTMENT
M/S41-SJ, 1109 MCKAY DRIVE
SAN JOSE, CA 95131 (US)

(73) Assignee: **NXP B.V.**, Eindhoven (NL)

(21) Appl. No.: **12/063,855**

(22) PCT Filed: **Aug. 7, 2006**

(86) PCT No.: **PCT/IB06/52721**

§ 371 (c)(1),
(2), (4) Date: **Feb. 15, 2008**

(30) **Foreign Application Priority Data**

Aug. 17, 2005 (EP) 05107557.0
Sep. 29, 2005 (EP) 05108995.1

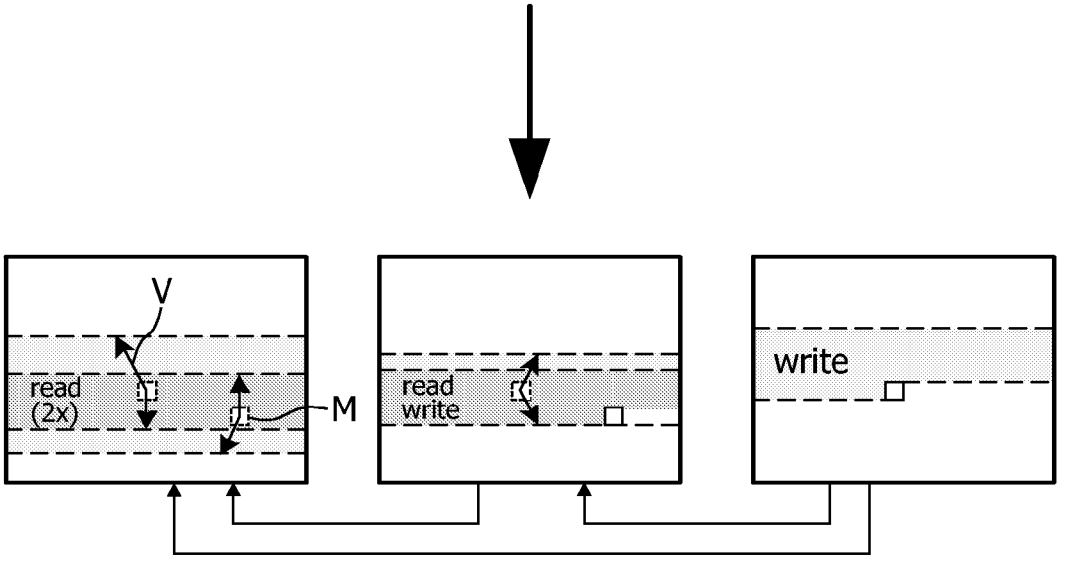
Publication Classification

(51) **Int. Cl.**
H04N 11/02 (2006.01)

(52) **U.S. Cl.** **375/240.01**; 375/E07.076

(57) **ABSTRACT**

The invention relates to a method for encoding/decoding a video stream including a plurality of images (A, B, C) in a video processing apparatus having a processing unit (**11**) coupled to a first memory (**12**), further comprising a second memory (**13**), comprising the steps: providing a subset of image data stored in the second memory (**13**) in the first memory (**12**), —simultaneous encoding/decoding of more than one image (B, C) of the video stream, by accessing said subset, wherein the simultaneously encoding/decoding is performed by access sharing to at least one image (A). For reducing the traffic or the number of access to second memory (**13**) an apparatus is proposed including: a processing unit (**11**) for performing an encoding/decoding process of video data; a first memory (**12**) coupled to the processing unit (**11**) for storing image data required for encoding/decoding of video data, —a second memory (**13**), wherein the video stream includes a plurality of images (A, B, C); the first memory (**11**) is adapted to store a subset of image data of the second memory (**13**) in the first memory (**12**) and the processing unit (**11**) is adapted to simultaneously encode/decode more than one image (B, C) of the video stream by accessing said subset of image data in the first memory (**12**), wherein the simultaneously encoding/decoding is performed by sharing access to at least one image.



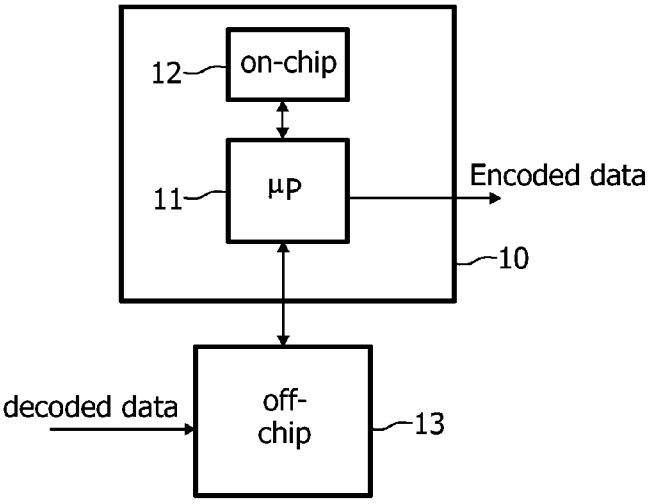

12
on-chip
11
µP
Encoded data
10
decoded data
off-
chip
13

FIG. 1

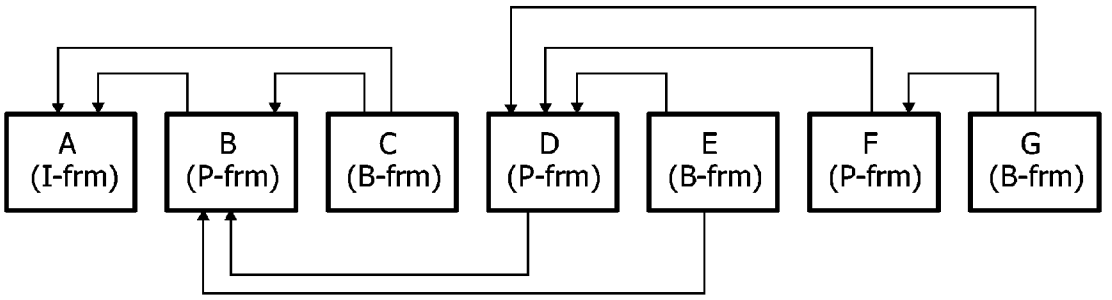

A
(I-frm)
B
(P-frm)
C
(B-frm)
D
(P-frm)
E
(B-frm)
F
(P-frm)
G
(B-frm)

FIG. 2

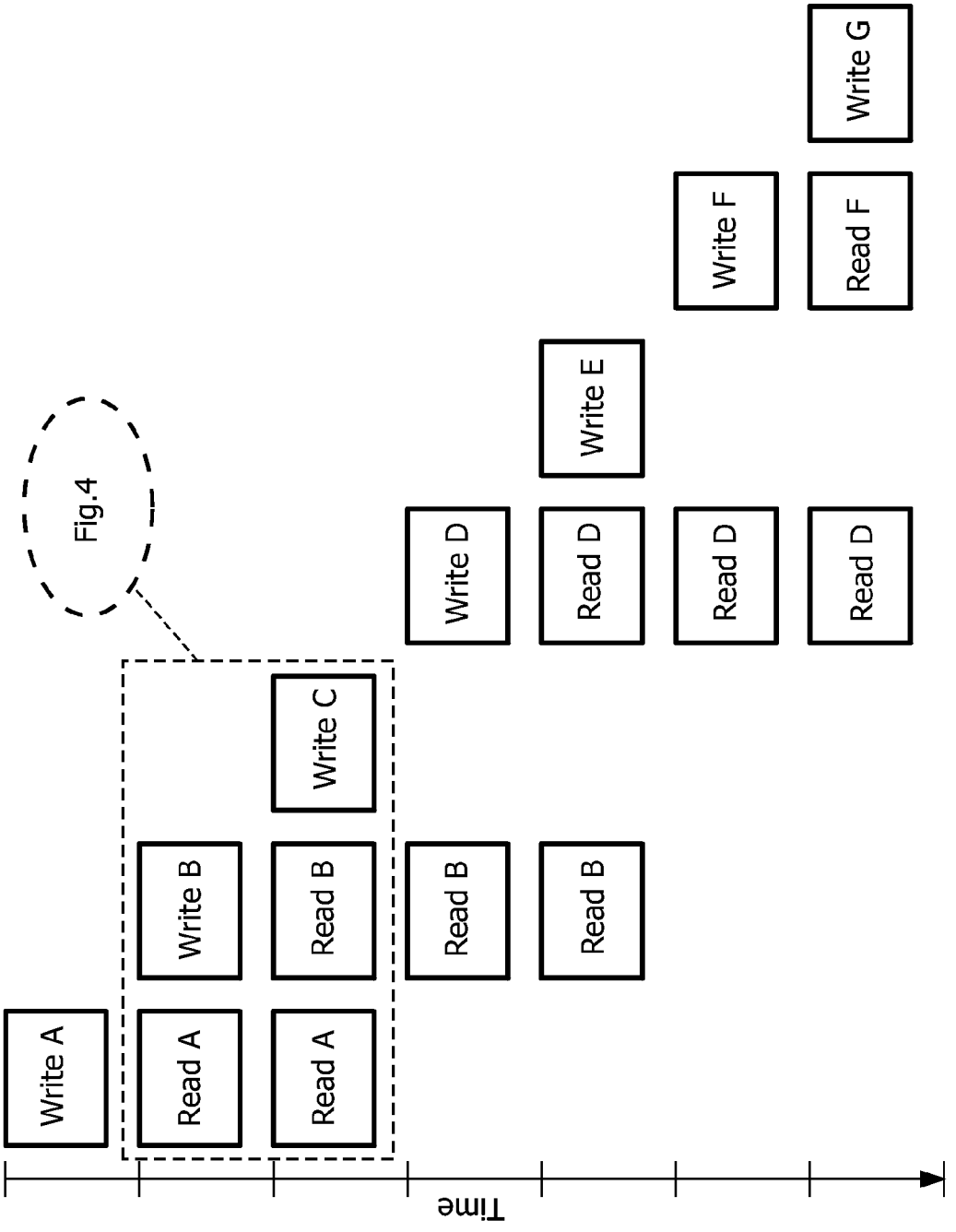

Write A
Read A
Read A
Write B
Read B
Write C
Read B
Read B
Write D
Read D
Write E
Read D
Read D
Write F
Read F
Write G
Time
Fig.4

FIG. 3

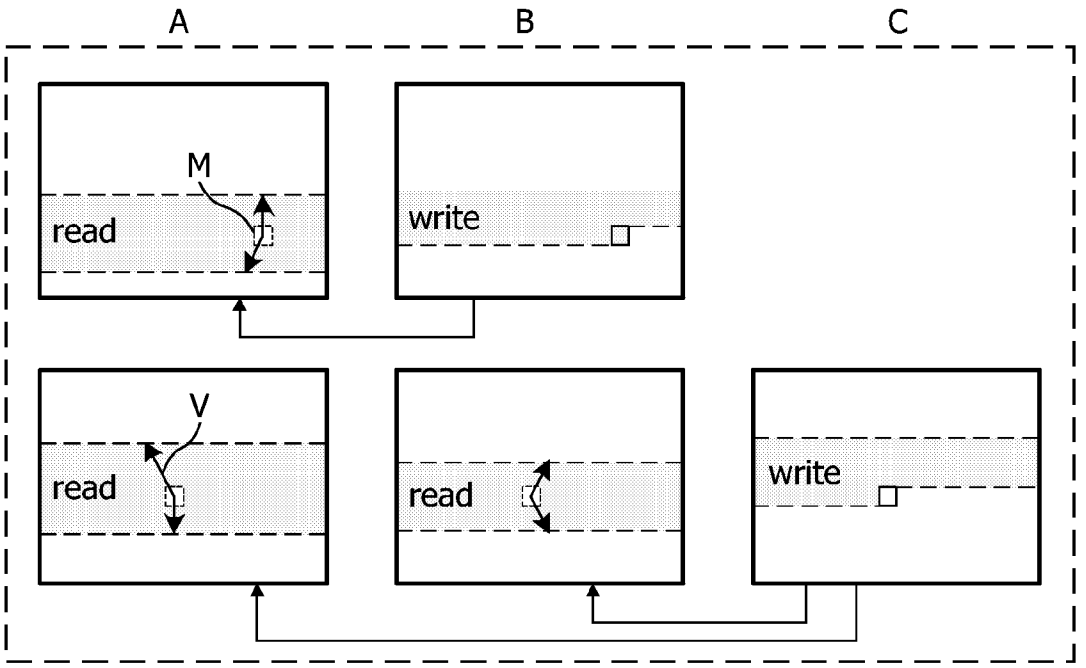


FIG. 4

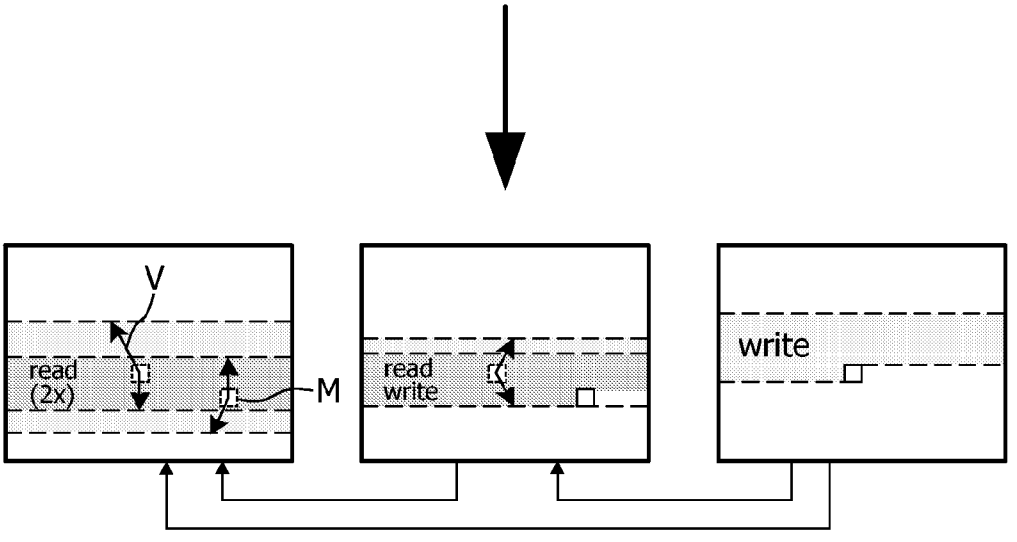


FIG. 5

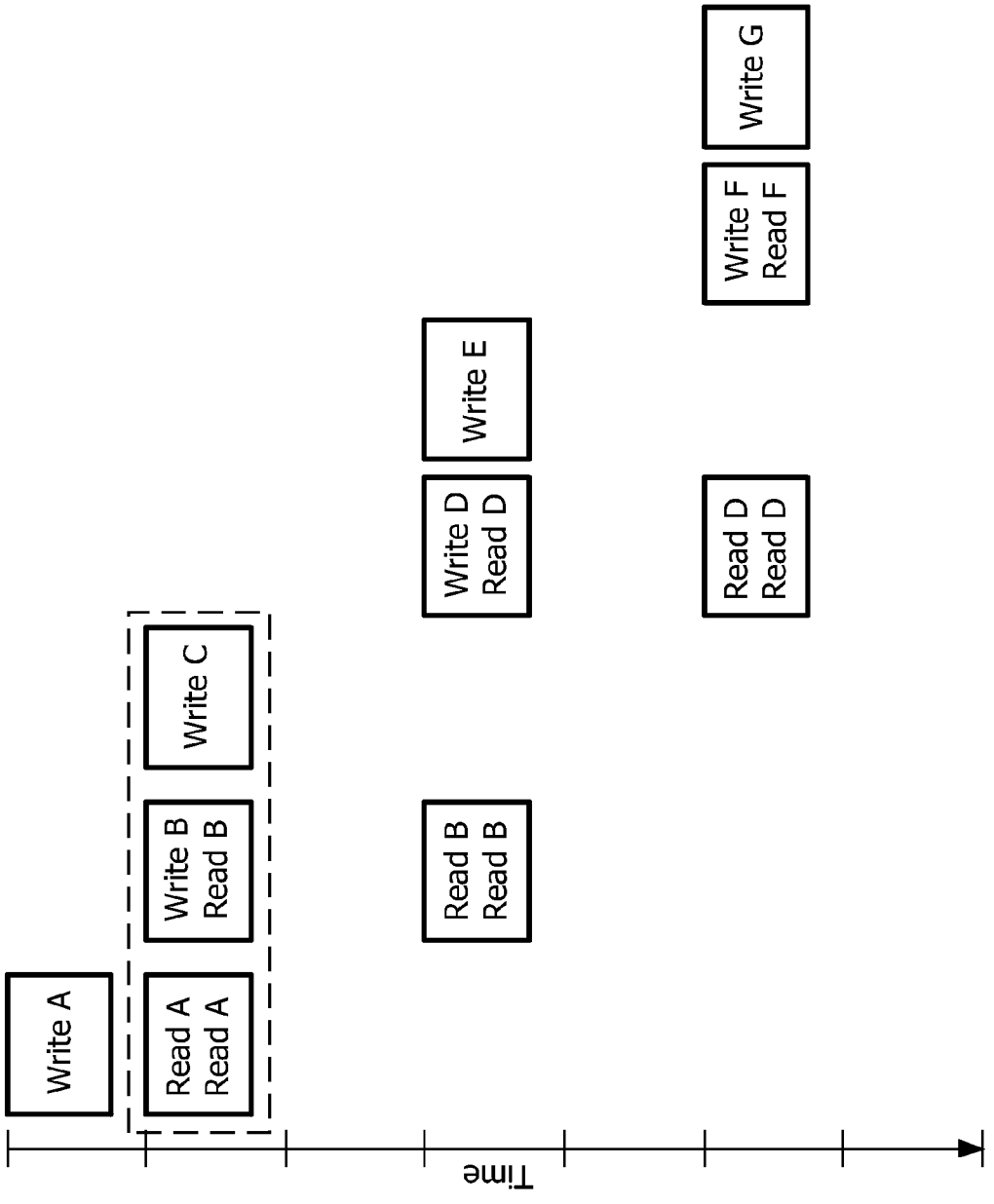

Write A
Read A
Read A
Write B
Read B
Write C
Read B
Read B
Write D
Read D
Write E
Read D
Read D
Write F
Read F
Write G
Time

FIG. 6

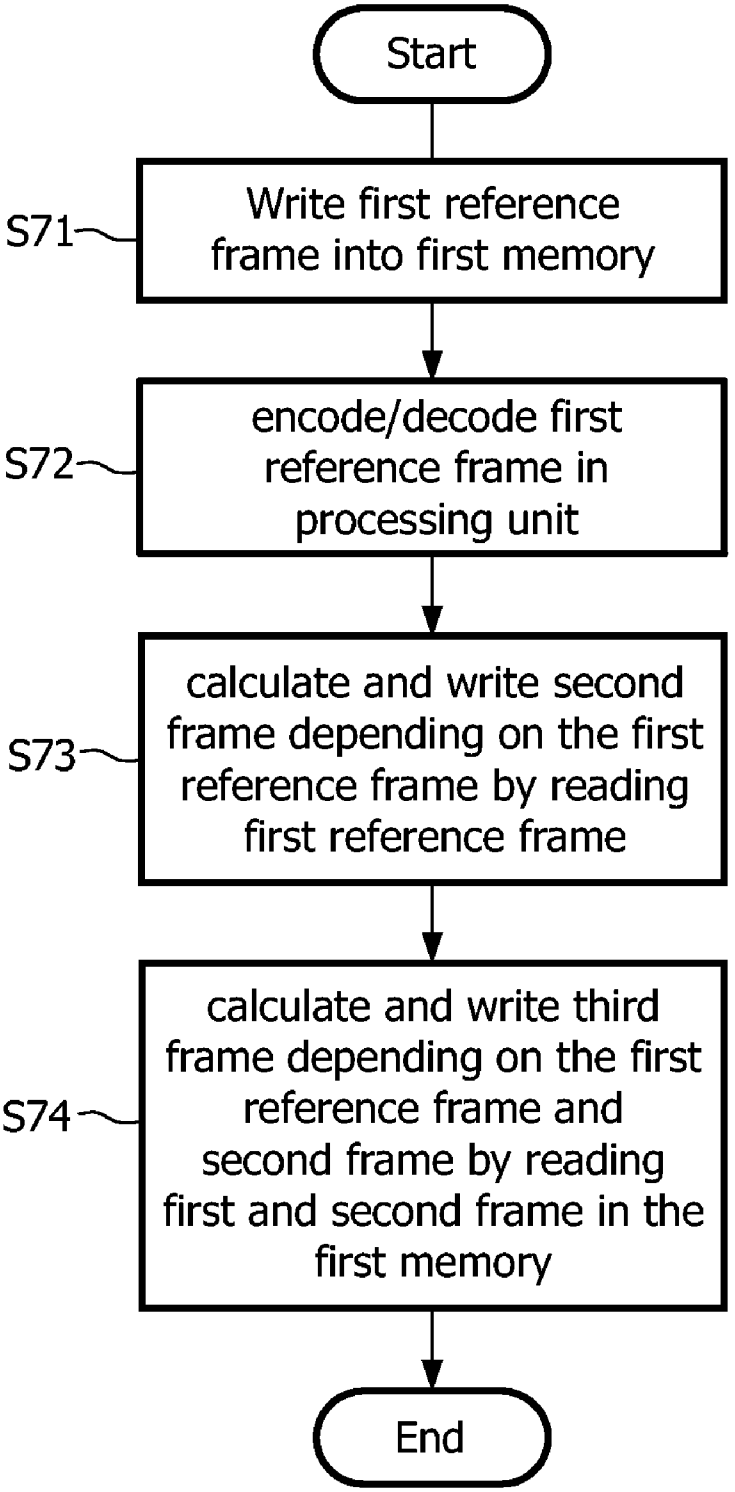

Start
S71
Write first reference frame into first memory
S72
encode/decode first reference frame in processing unit
S73
calculate and write second frame depending on the first reference frame by reading first reference frame
S74
calculate and write third frame depending on the first reference frame and second frame by reading first and second frame in the first memory
End

FIG. 7

METHOD VIDEO ENCODING AND DECODING PRESERVING CAHE LOCALITIES

[0001] The invention relates to method for encoding/decoding video data. Further it relates to an apparatus for encoding/decoding video data.

[0002] In video encoding/decoding reference frames are used. Typical examples of such coding are standards like MPEG2, H.264, etc. However, also in proprietary algorithms for e.g. web cams, reference frames are used.

[0003] A video processing apparatus includes in general a processor, e.g. a digital signal processor for performing the encoding/decoding calculation based on uncompressed/or encoded video data. There are different kinds of memories connected to that processor. Normally the video data are stored in memory having a large storage capacity, since there is a large amount of data to be processed for high quality video. Additionally there is smaller size memory coupled to the processor serving as temporary buffer. The larger size memory for storing the video is coupled via a connection to the processor having a limited bandwidth. The memory used as temporary buffer is connected via a connection having a higher bandwidth than the large memory connection. Mostly the large memory is arranged outside the chip including the processor; therefore it is designated as off-chip memory, wherein the temporary buffer is located on the same chip, thus called on-chip memory.

[0004] Reference images used for encoding/decoding are typically too large to fit in on-chip memory completely. For SD MPEG decoding it is required to store 1.2 Mbytes of reference image data. For HD MPEG decoding, it can take up to 6 Mbytes. For MPEG encoding, even more image memories are needed for image reordering. So for using such reference images an access to off-chip memory is required.

[0005] However also as result of long-term technology progress feature size decreases, so both memories may reside on a single chip. Also in this case, several levels of caching will be built. A first cache level will be "close" to processing units, wherein next cache levels will be at "larger distance" and have larger memory sizes. The reasoning still holds that bandwidth to a next level cache is significantly smaller than bandwidth to a first cache level. So the memory bottleneck will remain an important issue in future systems.

[0006] The trend to an increased picture resolution causes the reference frames to become larger in size, and thus requiring more memory to store. Hereby is it likely that in the future the reference frames will still be stored in off-chip memory. This results also in a higher off-chip memory bandwidth.

[0007] Further, the performance gap between processing or computational power and memory communication will increase with progress in silicon technology. Hence, both bandwidth considerations and memory access latency will become more dominant design parameters. Both endanger efficient utilization of the processing power on-chip due to lack of data.

[0008] More and more signal processing functions are designed for mobile applications with the obvious emphasis on power consumption. This poses an extra pressure to reduce off-chip communication, since high bandwidth off-chip communication requires substantial power.

[0009] Standard memory components usually comply to interface standards posing fixed bandwidth limits. Such bottleneck or limited bandwidth capacity imposes a very rigid limitation on the video processing system. For relaxing such limitation significant cost increasing is required. So it is possible to double the number of off-chip memory chips to double bus bandwidth. This increases system cost and Si-area and chip pin count.

[0010] As a result of these trends, the memory bottleneck will become even more prominent in the future.

[0011] The U.S. Pat. No. 6,263,112 B1 describes a motion vector searching apparatus and motion picture coding apparatus. It describes to reduce bandwidth requirements for video processing. Reference frames are used for encoding/decoding of video data. It is noted that B frames put the largest burden on bandwidth to reference frames, since B frame coding typically requires two reference frames, an I-frame and a P-frame. It discloses to use only a single reference frame for B frame coding, thereby taking advantage of previously calculated motion vectors. The motion vector is stored in a motion vector data storage memory. The reduction of bandwidth requirements results in a reduced image quality. Further, it is applicable only on an encoder. Additionally it increases the bit rate since not all features of the standard can be exploited.

[0012] Therefore it is object of the present invention to provide a method and an arrangement reducing the off-chip memory traffic for video encoding/decoding, without reduction of image quality or increasing of bit rate.

[0013] The object of the present invention is solved by the features of the independent claims.

[0014] The invention is based on the observation that a reference image is accessed multiple times for processing successive images. This holds for decoding as well as for encoding. In prior art, these accesses are relatively far apart in time and require separate accesses to the same image data in off-chip memory, or require that complete reference images are stored on-chip in the first memory.

[0015] The present invention proposes to utilize a single access to a subset of image data multiple times for processing multiple images, without having to store the complete reference images on chip. This is done by processing successive images simultaneously.

[0016] Most compression standards use motion compensation for the predictor. Causality constraints and differences in motion vectors prevent that exactly the same data is required at exactly the same moment in time for the images processed simultaneously. To overcome this discrepancy, a small on-chip buffer is still required. This on-chip buffer is designated in the following as first memory. The first memory contains a window or a subset of a reference image, in the order of the maximum vector range. The complete reference images are stored in the large second memory, typically located off-chip.

[0017] Since the first memory contains the relevant part of the reference image, it can also be used to compensate for the long latency of the second memory.

[0018] The idea is applicable for both video encoders and decoders independently. Examples are MPEG, and H.264. It is both useful in software and hardware encoding/decoding realizations.

[0019] In an advantageous embodiment at least one of the simultaneously encoded/decoded images is used as a reference image for encoding/decoding at least one of the other simultaneously encoded/decoded images. Thus the off-chip memory accesses are further reduced and reducing processing time and power consumption.

[0020] In another embodiment it is preferred that simultaneously encoded/decoded images share access to a common reference image. The subset stored in the first memory includes data of a common reference image used for encoding/decoding both simultaneously encoded/decoded images.

[0021] In a preferred embodiment the writing of data into the first memory and the reading of data from the first memory are synchronized, such that the part of the reference image required for decoding a dependent image or a part thereof is already decoded before starting decoding the dependent image. Thus it is assured that writing is ahead of reading between two simultaneously encoded/decoded images

[0022] It is further advantageous required to synchronize the accesses in the first memory for ensuring that the decoding/encoding of the first image of the simultaneously encoded/decoded images occurs at substantially equal image positions of the second image. Thus the amount of data stored in the first memory is kept small.

[0023] The synchronizing of the accesses in the first memory ensures that the accesses of the shared common reference image during simultaneously encoding/decoding first and second images occur at substantially equal image positions of the common reference image. This synchronizing assures that two reading accesses on a shared image are at about the same location.

[0024] In a further preferred embodiment the synchronization of the accesses is based on a vertical location in the respective images used as common reference image.

[0025] Preferably synchronization offsets between the accesses are based on maximum vertical component of the motion vector. The synchronization offset refers to the vertical difference between position in the image where the first image is written and position where the second image is being processed. For decoding the second image, data from the first image is required. That data needs to be written first. The precise position of the read access in the first image depends on a motion vector. This motion vector is obtained from the bit stream of the second image. According to the prior art, this would require a check and synchronization at every read operation in the first image. This results in many checks and synchronization actions, causing significant extra operations and complexity. However, there is usually a maximum value for the motion vectors. By using this maximum value as synchronization criterion, all possible vector values can be accessed and synchronization does not depend on specific vector values anymore.

[0026] Furthermore, by using only the vertical component of the maximum vector, the check and synchronization only needs to occur when vertically advancing over the image.

[0027] In a further preferred embodiment a search process is performed searching within the compressed data for the start of next image. Thus it is possible to simultaneously decode multiple images.

[0028] In a further preferred embodiment for encoding process the encoded stream is encoded in parallel. The encoded different images are concatenated to produce an image sequential bitstream according to the standard.

[0029] During encoding it is further possible to limit the size of the motion vector. Thereby it is possible to consider the size of the first memory, so that the working set of data does not exceed the first memory size.

[0030] The object is also solved by a video processing apparatus including a processing unit for performing an encoding/decoding process of video data; a first memory coupled to the processing unit for storing image data required for encoding/decoding of video data and a second memory, wherein the video stream includes a plurality of images; the first memory is adapted to store a subset of image data of the second memory in the first memory and the processing unit is adapted to simultaneously encode/decode more than one image of the video stream by accessing said subset of image data in the first memory, wherein the simultaneously encoding/decoding is performed by sharing access to at least one image.

[0031] In a further preferred embodiment, multiple processing units are arranged, each operating at the same time and processing a single image. This way, task level parallelism is effectively utilized to encode/decode a single video stream. The multiple processors will access a single first memory.

[0032] In yet another embodiment, a plurality of first memories is available, each located closely to their respective processing units. In this case, additional communication means are required between the first memories to duplicate the image data over the first memories. An example of such system is a multi-processor system where each processor is equipped with its own level 0 cache. Cache coherency hardware takes care of duplicating the image data over the various caches.

[0033] Other objects and advantages of the present invention will become apparent during the following discussion in conjunction with the accompanying drawings. Preferred embodiments of the invention are described in detail below, by way of example only, with reference to the following schematic drawings.

[0034] FIG. **1** a schematic structure of a video processing apparatus according to the invention;

[0035] FIG. **2** an illustration of dependencies of images (GOP);

[0036] FIG. **3** an illustration of read and write accesses to images for decoding the sequence of images as shown in FIG. **2** according to the prior art;

[0037] FIG. **4** an illustration showing that read and write accesses of the section in FIG. **3** according to the prior art which are limited to an image window;

[0038] FIG. **5** an illustration showing overlapping image windows for simultaneous calculation/decoding of two images according to the present invention;

[0039] FIG. **6** an illustration showing the reduction of reading accesses according to the present invention when decoding two images in parallel;

[0040] FIG. **7** presents a multiple processor configuration having multiple first memories;

[0041] The drawings are provided for illustrative purpose only and do not necessarily represent practical examples of the present invention to scale.

[0042] In the following the various exemplary embodiments of the invention are described.

[0043] Although the present invention is applicable in a broad variety of applications it will be described with the focus put on MPEG encoding/decoding applications. A further field for applying the invention might be the H.264/AVC algorithm or any other encoding/decoding algorithm using reference images.

[0044] FIG. **1** illustrates a structure of a video processing apparatus as used for the present invention. There is a processing unit **11**, e.g. a digital signal processor or a fixed-function block, performing the encoding/decoding process. A

first memory **12** is coupled to the processing unit **11** having a large bandwidth connection. The memory management can be directly controlled by the processing unit **11**. A not illustrated further control unit may also perform this, using e.g. direct memory access (DMA) techniques. As a further alternative, the first memory **12** may behave as a data cache of the second memory **13**, and generic caching strategies determine which data is available in the first memory **12**.

[0045] The second memory **13** may be arranged outside the chip **10**. The second memory **13** has a higher storage capacity than the first memory **11**. The connection from the second memory **13** to the processing unit **11** is limited in its bandwidth. In case of using a further memory control unit it is possible to write data direct from second memory **13** to the first memory **11** without using the processing unit **11**. FIG. **1** illustrates the decoding direction, wherein the same apparatus is used for the encoding direction, by only exchanging input and output of data.

[0046] FIG. **2** illustrates the dependencies of images in case of MPEG2 coding. There are I-images, B-images and P-images. The I-image A is decoded without using a reference image. The P-image B is decoded by use of the I-image A as reference image. For decoding the B-image C the decoded P-image B and the decoded I-image A are used as reference images. Such sequence of images is called a GOP (group of pictures). The IPBPBPB . . . sequence is known as GOP structure. Different GOP structures can be used; e.g. IPBBPBBPBB . . . to which the present invention can be applied. A P-image depends on the predecessor I-image, and a B-image depends on the predecessor I-image as well as on the predecessor P-image.

[0047] Starting from the well known dependencies of images, which are also called frames, the writing/reading access for decoding the sequences of frames shown FIG. **2** is explained referring to FIG. **3**. A video stream is encoded as a sequence of images. Normally, the images are decoded in the order of receiving the data. When a reference image A is used for decoding an image B, the reference image A that has been decoded before is reused. Typically, this reference image A has to be reread from off-chip or second memory **13** since the on-chip or first memory **12** is not large enough to keep the reference image A in the on-chip or first memory **12**. As depicted in FIG. **3**, a reference image A is written once, and two times reread for the purpose of being a reference image for decoding image B and image C. This requires a lot of off-chip or second memory **13** accesses.

[0048] FIG. **4** illustrates an enlarged section of FIG. **3**. FIG. **4** illustrates the calculation of a macro block M. For calculating such macro block a region or part of the reference image A is required. The size of this access region or subset depends on the maximum size of the motion vectors V for the macro blocks. For each referenced image, such "search area" can be recognized. When the macro block position proceeds downwards in the image, the search area slides also downwards. In state of the art systems, the data from these search area windows is usually available in local buffers or caches, similar to the first memory of the present invention. FIG. **4** illustrates that the data from image A is read once to produce image B and once again to produce image C. Furthermore, at approximately the same output image location, the contents of the respective access regions show considerable overlap. This property is exploited in the present invention as illustrated in FIG. **5**.

[0049] FIG. **5** illustrates overlapping image windows for simultaneous decoding of the two images B and C according to the present invention. Both images B and C require data from the image A. When decoding them simultaneously, this subset of image data is only read once from second memory **13** into first memory **12**. The decoding process then uses this subset of image data from the first memory **12** twice. Since the decoding positions in the respective images may differ somewhat, the combined access region is slightly enlarged compared to the access region for only a single decoded image. FIG. **5** shows two overlapping access regions, indicated by the gray shading. The darker gray shading indicates the overlap of the two areas. The clear consequence of this approach is a reduction to 50% of the second memory traffic for image A data.

[0050] Both images B and C are decoded simultaneously. Note however, that image B is also a reference image for image C. In the present invention, the calculated image data of image B is stored in the first memory **12** as well as copied to the second memory **13**. Thus an access region of image B for decoding image C is immediately available in first memory **12**.

[0051] A conventional decoder needs to write image B to second memory **13** and later, when decoding image C it is read again from second memory **13**. The current invention avoids the reading from second memory **13** yielding a reduction of 50% of second memory traffic for image B data.

[0052] The process of simultaneously decoding two images B and C is applied on the example of FIG. **3**. The result according to present invention is shown in FIG. **6**. In this image, the horizontal width of a block is illustrative for the bandwidth to an image in second memory **13**. This height of a block is illustrative for the duration of the operation. The number of accesses to the second memory **13** is reduced significantly.

[0053] In the standard case as described in FIG. **3** five full accesses to the second memory are required (Read A, Write B, Read A, Read B, Write C) to decode image B and C. In the situation according to this invention described in FIG. **6** there are only three full accesses to the second memory **13** required (Read A, Write B, Write C). Thus a reduction by $\frac{2}{5}$=40% of off-chip bandwidth consumption is achieved. In this particular example, a choice is made to process two images B, C simultaneously. Other choices are possible, depending on GOP structure and availability of on-chip buffer memory or first memory **12**. Thus two B images may be processed simultaneously. It is also possible to process an I-image and a P-image simultaneously. Further, it is possible to process more than two images simultaneously.

[0054] It should be noted, that the reduction of bandwidth occurs at the bandwidth peaks. When decoding a B-image C, three accesses are required to decode a single image (2× read, 1 write). For a P-frame B, two access are required, for an I-frame A only a single access. So as an additional advantage, this invention smoothes the bandwidth consumption over time. This relaxes the design target for off-chip bandwidth, since the system must support worst-case bandwidth consumption. Further it allows better scheduling and bus utilization due to the more constant bandwidth consumption of the video encoder/decoder.

[0055] The realization of this idea uses the standard decoding algorithm. Compared to a normal realization it requires only an additional synchronization process to ensure that the decoding of the image B is ahead of the calculation of the

image C that depends on that reference image B. Further, that the decoding of the reference image B is not too much ahead, to ensure that the data of the reference image B is still in the first memory **12** when decoding the image C that depends on the reference image B.

[0056] Further it is required to scan efficiently through the compressed video stream to find the beginning of the next image, without having to decode the current image first. Both additions in the realization are relatively simple to implement.

[0057] In the following it is described which data are in the first memory **12**:

[0058] 1. Motion vectors of all images B and C that are decoded concurrently.

[0059] Since motion vectors are available for every macro block, this requires only a fraction of a image store capacity.

[0060] 2. Compressed video data for the images B, C that are concurrently decoded. (since this is data in the compressed domain, this also requires only a fraction of an image store)

[0061] 3. Storing the parts of the reference images as depicted in FIG. **5**. The required amount of first memory **12** depends on the maximum vector sizes.

[0062] Lets take SD MPEG as example: Lets calculate a B-image at the same time as the P-image used as reference. Assume that the motion vectors of the P-image are at most 64 pixels. To keep the required I-image data on-chip requires 128*720*1.5 equals 135 Kbytes. To calculate the B-image, we need the P-image as reference. Lets assume we also need 135 Kbytes to host the P-image reference data, and an 65 Kbytes for additional data on the I-image since the decoding of the B-image is typically behind the P-image. In total 335 Kbytes are required. Having more than one B-image does not increase the amount of first memory needed. Having more than two reference images (e.g. H.264) will increase the required amount of first memory.

[0063] In the following the encoding process is described. For video encoding, the same principle as for decoding can be used. One can encode the reference images and the images that depend on it at the same time. There are a number of advantages compared to decoding: When to encode a normal image or reference image can be decided freely.

[0064] The encoded stream is created in parallel. The encoded streams of different images have to be concatenated in the right order. This is simpler than the required parsing for decoding and requires no substantial computation power. During encoding, one may choose not to use all possible features of the standard. This allows for optimizations that take the properties of the hardware architecture into account.

Some Examples:

[0065] One may choose to limit motion vectors such that the working set of image data does not exceed the on-chip buffer size.

[0066] Further one may choose to encode a B image with references to only a single image (so this B image uses references similar to a P image). Since a B image is only written, and a P image is also read, this reduces access on the local memory. It saves local buffer bandwidth.

[0067] In some cases, extra memory is required (trading bandwidth for memory consumption, which is still advantageous from cost point of view).

[0068] Some extra latency is added to the decoder; in many applications this is no problem at all (e.g. DVD decoding, broadcast, etc.)

[0069] In yet an other embodiment, a plurality of processing units **11***x*, **11***y*, **11***z* and first memories **12***x*, **12***y*, **12***z* is available, each located closely to their respective processing units **11***x*, **11***y*, **11***z*. In this case, additional communication means **14** between the first memories **12***x*, **12***y*, **12***z* take care to duplicate the image data over the first memories. An example of such system is a multi-processor system where each processing unit **11***x*, **11***y*, **11***z* is equipped with its own level 0 cache. Cache coherency hardware takes care of duplicating the image data over the various caches.

[0070] The FIG. **7** shows a plurality of processing units **11***x*, **11***y*, **11***z*, each connected to a first on-chip memory **12***x*, **12***y*, **12***z*. The first memories are connected to a communication network **14**, which is also connected to an off-chip second memory **13**. The communication network **14** supports both data transfers from a first memories **12***x*, **12***y*, **12***z* to the second memory **13**, and between first memories **12***x*, **12***y*, **12***z*. The first memories **12***x*, **12***y*, **12***z* may reside close a CPU **11***x*, **11***y*, **11***z* as data cache. In such system, multiple images can be processed simultaneously such that a single processing unit **11***x*, **11***y*, **11***z*, processes each of the images. E.g. decoding of image B can be performed by processor **11***x* while processor **11***y* is decoding image C. Cache coherency needs to be handled in this system. In a pure hierarchical memory system, a cache miss in a first memory always causes an access in the second memory. However, with cache coherency, first a check is done whether the data is already available in any of the other on-chip caches. If that is the case, the other on-chip cache is used instead of the off-chip memory. This way, the data access to the bandwidth-limited off-chip memory can still be reduced. E.g. when image C is processed on processor **11***y*, data from image A is used as reference image and loaded into first memory **12***y*. Meanwhile, processor **11***x* is processing image B, which also reads image data from image A. The cache coherency functionality takes care that this data is taken from first memory **12***y* and loaded into first memory **12***x* to be available for processor **11***x*. This way, the image data of image A is still read only once from the off-chip second memory **13**, while it is being used for simultaneous decoding of both image B and image C.

[0071] By use of the present invention following advantages are provided: A bandwidth reduction of about 40% on typical MPEG encoding/decoding (IBPBPBP) is achieved since only 3 full access to the second memory are required still in comparison to the conventional method in which 5 accesses to the second memory were required.

[0072] In case that more B-images are included in the group of pictures (E.g. a GOP structure of IBBPBBPBBP) a higher reduction is possible by increasing the number of images processed simultaneously.

[0073] Due to the reduced amount of accesses to the second memory the power consumption for encoding/decoding is decreased, which is very advantageously for mobile encoding.

[0074] The peaks in bandwidth are reduced so a more continuous bandwidth use is achieved.

[0075] The present invention may be applied on standards like MPEG, H.264. However it may be applied for a decoder independent of the encoder that has encoded the data, and vice versa. Further, it is applicable for SD to HD video.

[0076] The decoding algorithm is not impacted. There are no or minimal impacts on encoding algorithm. To implement the invention a very low impact on realisation of an encoder or decoder is required: it is required to add a synchronization process between the reference image producer and consumer. Further, it is required to add a search to start of next image in compressed data to start simultaneous decoding of multiple images. When encoding, the bit stream fragments that are produced simultaneously need to be concatenated in the right order. All these operations are relatively straightforward.

[0077] It is apparent that, in this invention, a wide range of different encoding or decoding methods may be used without deviating from the scope of the invention. This invention is not restricted by its specific encoding/decoding methods except being limited by the appended claims.

[0078] The numbers or letters between brackets "(" and ")" in the appended claims refer to elements of the figures. They are meant for illustrative and explanatory purposes only and must not be interpreted as limiting the scope of the claims.

[0079] The method described in this invention can be applied in an apparatus consisting of dedicated hardware units, or a programmable processing unit equipped with software code for the required functionality, or by programmable hardware configured for the required functionality, or any combination thereof.

1. Method for encoding/decoding a video stream including a plurality of images in a video processing apparatus having a processing unit coupled to a first memory, further comprising a second memory, comprising:

providing a subset of image data stored in the second memory in the first memory;

simultaneous encoding/decoding of more than one image of the video stream, by accessing said subset, wherein the simultaneously encoding/decoding is performed by access sharing to at least one image.

2. Method as claimed in claim **1**, further comprising:

synchronizing the accesses in the first memory for ensuring that the accesses occur at substantially equal image positions in the shared image.

3. Method as claimed in claim **1**, wherein at least one of the simultaneously encoded/decoded images is used as a reference image for encoding/decoding at least one of the other simultaneously encoded/decoded images.

4. Method as claimed in claim **1**, wherein the simultaneously encoded/decoded images share access to a common reference image.

5. Method as claimed in claim **1**, further comprising:

synchronizing the accesses in the first memory for ensuring that the decoding/encoding of the first image of the simultaneously encoded/decoded images is ahead of the reading position of the second image that depends on the first image.

6. Method as claimed in claim **2**, wherein the synchronization is based on a vertical location in the respective.

7. Method as claimed in claim **6**, wherein synchronization offsets are based on maximum vertical component of the motion vector.

8. Method as claimed in claim **1** for decoding a video stream, further comprising searching of the video stream for finding the beginning of a next image, without having decoded the image yet.

9. Method as claimed in claim **1** for encoding a video stream, wherein the data of simultaneously encoded images G are concatenated to produce an image sequential bit stream.

10. A video processing apparatus comprising:

a processing unit for performing an encoding/decoding process of video data;

a first memory coupled to the processing unit for storing image data required for encoding/decoding of video data;

a second memory wherein the video stream includes a plurality of images; the first memory is adapted to store a subset of image data of the second memory in the first memory and the processing unit is adapted to simultaneously encode/decode more than one image of the video stream by accessing said subset of image data in the first memory, wherein the simultaneously encoding/decoding is performed by sharing access to at least one image.

11. Video processing apparatus as claimed in claim **10** having multiple processing units each operating at the same time and processing a single image, further comprising a plurality of first memories, each associated to a respective processing unit, wherein a communication unit is provided between the first memories to provide data access to a first memory not directly associated to the processing unit that requires the data.

* * * * *