

(19)



(11)

**EP 1 917 812 B1**

(12)

**EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:  
**11.11.2009 Bulletin 2009/46**

(51) Int Cl.:  
**H04N 7/26 (2006.01) H04N 7/46 (2006.01)**

(21) Application number: **06795604.5**

(86) International application number:  
**PCT/IB2006/052721**

(22) Date of filing: **07.08.2006**

(87) International publication number:  
**WO 2007/020560 (22.02.2007 Gazette 2007/08)**

**(54) VIDEO ENCODING AND DECODING PRESERVING CACHE LOCALITIES**

VIDEOCODIERUNG UND -DECODIERUNG UNTER BERÜCKSICHTIGUNG VON POSITIONEN IM CACHE

CODAGE/DECODAGE VIDEO PERMETTANT DE PRESERVER DES EMPLACEMENTS D'ANTEMEMOIRE

(84) Designated Contracting States:  
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IS IT LI LT LU LV MC NL PL PT RO SE SI SK TR**  
Designated Extension States:  
**AL BA HR MK RS**

(74) Representative: **Schouten, Marcus Maria et al NXP B.V.**  
**IP & L Department**  
**High Tech Campus 32**  
**5656 AE Eindhoven (NL)**

(30) Priority: **17.08.2005 EP 05107557**  
**29.09.2005 EP 05108995**

(56) References cited:  
**EP-A- 1 542 474**

(43) Date of publication of application:  
**07.05.2008 Bulletin 2008/19**

- **GE S ET AL: "Efficient Multithreading Implementation of H.264 Encoder on Intel Hyper-Threading Architectures" PROCEEDINGS OF THE FOURTH INTERNATIONAL CONFERENCE ON INFORMATION, COMMUNICATIONS AND SIGNAL PROCESSING AND THE FOURTH PACIFIC-RIM CONFERENCE ON MULTIMEDIA (ICICS-PCM 2003), 15 December 2003 (2003-12-15), pages 469-473, XP010701105 IEEE, Piscataway, NJ, USA ISBN: 0-7803-8185-8**
- **TSUO H-K ET AL: "An Efficient Macroblock-Sync Decoding Architecture for MPEG-4 FGS Video" PROCEEDINGS OF THE 45TH MIDWEST SYMPOSIUM ON CIRCUITS AND SYSTEMS, vol. 1 of 3, 4 August 2002 (2002-08-04), pages 232-235, XP010635193 IEEE, US ISBN: 0-7803-7523-8**
- **CHEN Y-K ET AL: "Media Applications on Hyper-Threading Technology" INTEL TECHNOLOGY JOURNAL, US, February 2002 (2002-02), pages 1-11, XP002296402**

(73) Proprietor: **NXP B.V.**  
**5656 AG Eindhoven (NL)**

(72) Inventors:  
• **VAN DOREN, Giel**  
**c/o Philips Intellectual Property & Standards GmbH**  
**52066 Aachen (DE)**  
• **RIEMENS, Abraham Karel**  
**c/o Philips Intellectual Property & Standards GmbH**  
**52066 Aachen (DE)**  
• **DE BRUIJN, Frederik Jan**  
**c/o Philips Intellectual Property & Standards GmbH**  
**52066 Aachen (DE)**

**EP 1 917 812 B1**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

## Description

**[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 article, Ge. S. et al "Efficient Multithreading implementation of H.264 Encoder on Intel Hyper-Threading architecture", Proceedings of the fourth international conference on information, Dec, 2003, describes a test architecture comprising a CPU cache and an external main memory. It describes to generate a parallel code structure, a multithreaded code suitable to run on a hyper-threading architecture and to convert sequential programs into multithreaded programs and to thereby parallelize the code to exploit the Hyper-Threading Technology. This parallisation is done within the H.264 coding scheme via data domain decomposition on slice-level and on frame-level.

**[0012]** The article Tsuo, H.K. et al "An efficient macroblock sync decoding architecture for MPEG-4 FGS Video", proceedings of the 45th midwest symposium, on circuits and systems, 4. Aug. 2002, describes a decoding architecture for a MPEG-4 decoder eliminating unnecessary buffer requirements while optimizing memory accessing and cache utilization.

**[0013]** The article Chen Y-K et al, "Media applications on Hyper-Threading Technology" describes performance improvements that are possible in multimedia applications using Hyper-Threading Technology architectures. It uses cache localities and the sharing of caches.

**[0014]** The US 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.

**[0015]** 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.

**[0016]** The object of the present invention is solved by the features of the independent claims.

**[0017]** 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.

**[0018]** 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.

**[0019]** 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.

**[0020]** 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.

**[0021]** 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.

**[0022]** 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.

**[0023]** 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.

**[0024]** 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

**[0025]** Further it is required to synchronize the accesses in the first memory for ensuring that the decoding/encoding of the first image of the simultaneously encoded/

ed/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.

**[0026]** 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.

**[0027]** Further the synchronization of the accesses is based on a vertical location in the respective images used as common reference image.

**[0028]** 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.

**[0029]** 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.

**[0030]** 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.

**[0031]** 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.

**[0032]** 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.

**[0033]** The object is also solved by a video processing apparatus as claimed in claim 7.

**[0034]** 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.

**[0035]** In yet another embodiment, a plurality of first memories is available, each located closely to their re-

spective 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.

**[0036]** 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.

- Fig. 1 a schematic structure of a video processing apparatus according to the invention;
- Fig. 2 an illustration of dependencies of images (GOP);
- 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;
- 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;
- Fig. 5 an illustration showing overlapping image windows for simultaneous calculation/decoding of two images according to the present invention;
- Fig. 6 an illustration showing the reduction of reading accesses according to the present invention when decoding two images in parallel;
- Fig. 7 presents a multiple processor configuration having multiple first memories;

**[0037]** The drawings are provided for illustrative purpose only and do not necessarily represent practical examples of the present invention to scale.

**[0038]** In the following the various exemplary embodiments of the invention are described.

**[0039]** 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.

**[0040]** Figure 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.

**[0041]** The second memory 13 may be arranged out-

side 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. Figure 1 illustrates the decoding direction, wherein the same apparatus is used for the encoding direction, by only exchanging input and output of data.

**[0042]** Figure 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 IPBPB-PB... 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.

**[0043]** Starting from the well known dependencies of images, which are also called frames, the writing / reading access for decoding the sequences of frames shown figure 2 is explained referring to figure 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 Figure 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.

**[0044]** Figure 4 illustrates an enlarged section of figure 3. Figure 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. Figure 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 Figure 5.

**[0045]** Figure 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. Figure 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.

**[0046]** 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.

**[0047]** 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.

**[0048]** The process of simultaneously decoding two images B and C is applied on the example of Figure 3. The result according to present invention is shown in Figure 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.

**[0049]** In the standard case as described in figure 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 figure 6 there are only three full accesses to the second memory 13 required (Read A, Write B, Write C). Thus a reduction by  $2/5 = 40\%$  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.

**[0050]** 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 (2x 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.

**[0051]** 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.

**[0052]** 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.

**[0053]** In the following it is described which data are in the first memory 12:

1. Motion vectors of all images B and C that are decoded concurrently. Since motion vectors are available for every macro block, this requires only a fraction of an image store capacity.
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)
3. Storing the parts of the reference images as depicted in Figure 5. The required amount of first memory 12 depends on the maximum vector sizes.

**[0054]** 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  $128 \times 720 \times 1.5$  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.

**[0055]** 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.

**[0056]** 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:

**[0057]** One may choose to limit motion vectors such that the working set of image data does not exceed the on-chip buffer size.

**[0058]** 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.

**[0059]** In some cases, extra memory is required (trading bandwidth for memory consumption, which is still advantageous from cost point of view).

**[0060]** Some extra latency is added to the decoder; in many applications this is no problem at all (e.g. DVD decoding, broadcast, etc.)

**[0061]** In yet another embodiment, a plurality of processing units 11x, 11y, 11z and first memories 12x, 12y, 12z is available, each located closely to their respective processing units 11x, 11y, 11z. In this case, additional communication means 14 between the first memories 12x, 12y, 12z 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 11x, 11y, 11z is equipped with its own level 0 cache. Cache coherency hardware takes care of duplicating the image data over the various caches.

**[0062]** The figure 7 shows a plurality of processing units 11x, 11y, 11z, each connected to a first on-chip memory 12x, 12y, 12z. 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 12x, 12y, 12z to the second memory 13, and between first memories 12x, 12y, 12z. The first memories 12x, 12y, 12z may reside close a CPU 11x, 11y, 11z as data cache. In such system, multiple images can be processed simultaneously such that a single processing unit 11x, 11y, 11z, processes each of the images. E.g. decoding of image B can be performed by processor 11x while processor 11y 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 11y, data from image A is used as reference image and loaded into first memory 12y. Meanwhile, processor

11x 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 12y and loaded into first memory 12x to be available for processor 11x. 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.

**[0063]** By use of the present invention following advantages are provided: A bandwidth reduction of about 40% on typical MPEG encoding/decoding (IBBPBPBP) 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.

**[0064]** 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.

**[0065]** 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.

**[0066]** The peaks in bandwidth are reduced so a more continuous bandwidth use is achieved.

**[0067]** 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.

**[0068]** 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.

**[0069]** 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.

**[0070]** 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.

**[0071]** 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.

## Claims

1. 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); **characterised by**
- synchronizing the accesses in the first memory (12) for ensuring that the accesses occur at substantially equal image positions in the shared image (A), wherein the synchronization is based on a vertical location in the respective images (A, B) and the synchronization offsets are based on maximum vertical component of the motion vector (M).

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

3. Method as claimed in claim 1, wherein the simultaneously encoded/decoded images (B, C) share access to a common reference image (A).

4. Method as claimed in one of the claims 1 to 3, further comprising:

- synchronizing the access in the first memory (12) for ensuring that the decoding/encoding of the first image (B) of the simultaneously encoded/decoded images (B, C) is ahead of the reading position of the second image (C) that depends on the first image (B).

5. Method as claimed in one of the claims 1 to 4 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.

6. Method as claimed in one of the claims 1 to 4 for encoding a video stream, wherein the data of simultaneously encoded images (A, B, C) are concatenated to produce an image sequential bit stream.

7. A video apparatus 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);

wherein the first memory (12) 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), **characterised in that** the simultaneously encoding/decoding is performed by sharing access to at least one image, wherein the first memory (12) is further adapted to allow synchronized access for ensuring that the accesses in the first memory (12) occur at substantially equal image positions in the shared image (A), wherein the synchronization is based on a vertical location in the respective images (A, B) and the synchronization offsets are based on maximum vertical component of the motion vector (M).

8. Video processing apparatus as claimed in claim 7, having multiple processing units (11x, 11y, 11z), each operating at the same time and processing a single image (A, B, C), further comprising a plurality of first memories (12x, 12y, 12z), each associated to a respective processing unit (11x, 11y, 11z), wherein a communication unit (14) is provided between the first memories (12x, 12y, 12z) to provide data access to a first memory not directly associated to the processing unit (11x, 11y, 11z) that requires the data.

## Patentansprüche

1. Verfahren zum Codieren/Decodieren eines Videodatenstroms, der eine Vielzahl von Bildern (A; B, C) enthält, in einer Videoverarbeitungsvorrichtung, die eine Verarbeitungseinheit (11) aufweist, die mit einem ersten Speicher (12) gekoppelt ist, und die ferner einen zweiten Speicher (13) umfasst, die Schritte umfassend:

- Bereitstellen einer Teilmenge der Bilddaten, die im zweiten Speicher (13) gespeichert sind, im ersten Speicher (12);
- gleichzeitiges Codieren/Decodieren von mehr als einem Bild (B, C) des Videodatenstroms durch Zugreifen auf die Teilmenge, wobei das gleichzeitige Codieren/Decodieren durch gemeinsamen Zugriff auf wenigstens ein Bild (A) durchgeführt wird;
- gekennzeichnet durch**
- Synchronisieren der Zugriffe im ersten Speicher

- (12), um sicherzustellen, dass die Zugriffe an im Wesentlichen gleichen Bildpositionen im gemeinsam genutzten Bild (A) stattfinden, wobei die Synchronisierung auf einer vertikalen Position in den entsprechenden Bildern (A, B) beruht und die Synchronisierungsversätze auf einer maximalen vertikalen Komponente eines Bewegungssektors (M) beruhen.
2. Verfahren nach Anspruch 1, wobei wenigstens eines der gleichzeitig codierten/decodierten Bilder (B, C) als Referenzbild (B) zum Codieren/Decodieren wenigstens eines der anderen gleichzeitig codierten/decodierten Bilder (C) verwendet wird.
  3. Verfahren nach Anspruch 1, wobei die gleichzeitig codierten/decodierten Bilder (B, C) sich den Zugriff auf ein gemeinsames Referenzbild (A) teilen.
  4. Verfahren nach irgendeinem der Ansprüche 1 bis 3, ferner umfassend:
 

Synchronisieren des Zugriffs im ersten Speicher (12), um sicherzustellen, dass das Decodieren/Codieren des ersten Bildes (B) der gleichzeitig codierten/decodierten Bilder (B, C) vor der Leseposition des zweiten Bildes (C) liegt, das vom ersten Bild (B) abhängt.
  5. Verfahren nach irgendeinem der Ansprüche 1 bis 4, zum Decodieren eines Videodatenstroms, das ferner das Durchsuchen des Videodatenstroms umfasst, um den Anfang eines nächsten Bildes zu finden, ohne das Bild bereits decodiert zu haben.
  6. Verfahren nach irgendeinem der Ansprüche 1 bis 4 zum Codieren eines Videodatenstroms, wobei die Daten der gleichzeitig codierten Bilder (A, B, C) verkettet sind, um einen bildsequenziellen Bitstrom zu erzeugen.
  7. Vorrichtung, enthaltend:
 

eine Verarbeitungseinheit (11) zum Ausführen eines Codierungs/Decodierungsprozesses von Videodaten;

einen ersten Speicher (12), der mit der Verarbeitungseinheit (11) gekoppelt ist, um die für das Codieren/Decodieren der Videodaten erforderlichen Bilddaten zu speichern;

einen zweiten Speicher (13), wobei der Videodatenstrom eine Vielzahl von Bildern (A, B, C) enthält;

wobei der erste Speicher (12) so ausgeführt ist, dass er eine Teilmenge der Bilddaten des zweiten Speichers (13) im ersten Speicher (12) speichert, und die Verarbeitungseinheit (12) so ausgeführt ist, dass sie gleichzeitig mehr als ein Bild

(B, C) des Videodatenstroms codiert/decodiert, indem sie auf die Teilmenge der Bilddaten im ersten Speicher (12) zugreift,

**dadurch gekennzeichnet, dass** das gleichzeitige Codieren/Decodieren ausgeführt wird, indem gemeinsam auf wenigstens ein Bild zugegriffen wird, wobei der erste Speicher (12) ferner so ausgelegt ist, dass er einen synchronisierten Zugriff erlaubt, um sicherzustellen, dass die Zugriffe im ersten Speicher (12) an im Wesentlichen gleichen Bildpositionen im gemeinsam genutzten Bild (A) stattfinden, wobei die Synchronisierung auf einer vertikalen Position in den jeweiligen Bildern (A, B) beruht und die Synchronisierungsversätze auf einer maximalen vertikalen Komponente eines Bewegungsvektors (M) beruhen.

8. Videoverarbeitungsvorrichtung nach Anspruch 7, die eine Vielzahl von Verarbeitungseinheiten (11 x, 11 y, 11 z) aufweist, die jeweils gleichzeitig arbeiten und ein einzelnes Bild (A, B, C) verarbeiten, ferner eine Vielzahl von ersten Speichern (12x, 12y, 12z) umfassend, die jeweils einer entsprechenden Verarbeitungseinheit (11x, 11y, 11z) zugeordnet sind, wobei eine Kommunikationseinheit (14) zwischen den ersten Speichern (12x, 12y, 12z) vorgesehen ist, um Datenzugriff auf einen ersten Speicher bereitzustellen, der nicht direkt der Verarbeitungseinheit (11x, 11y, 11z) zugeordnet ist, die die Daten benötigt.

## Revendications

1. Procédé d'encodage/décodage d'un flux vidéo incluant une pluralité d'images (A ; B, C) dans un appareil de traitement vidéo ayant une unité de traitement (11) raccordée à une première mémoire (12), comprenant en outre une deuxième mémoire (13), le procédé comprenant les étapes suivantes :
  - fourniture à la première mémoire (12) d'un sous-ensemble des données d'images stockées dans la deuxième mémoire (13) ;
  - encodage/décodage simultanés de plus d'une image (B, C) du flux vidéo, par accès audit sous-ensemble, dans lequel l'encodage/décodage simultané est réalisé grâce à un partage d'accès à au moins une image (A) ; **caractérisé par**
  - la synchronisation des accès à la première mémoire (12) pour garantir que les accès se produisent sur des positions d'image substantiellement égales dans l'image partagée (A), dans lequel la synchronisation est fondée sur un emplacement vertical dans les images respectives (A, B) et dans lequel les écarts de synchronisation sont fondés sur la composante verticale

maximale du vecteur de déplacement (M).

2. Procédé selon la revendication 1, dans lequel au moins une des images encodées/décodées simultanément (B, C) est utilisée comme image de référence (B) pour encoder/décoder au moins l'une parmi les autres images encodées/décodées simultanément (C). 5
3. Procédé selon la revendication 1, dans lequel les images encodées/décodées simultanément (B, C) partagent un accès à une image de référence commune (A). 10
4. Procédé selon l'une des revendications 1 à 3, comprenant en outre : 15
  - la synchronisation de l'accès à la première mémoire (12) pour garantir que l'encodage/décodage de la première image (B) des images encodées/décodées simultanément (B, C) est en avance par rapport à la position de lecture de la deuxième image (C) qui dépend de la première image (B). 20
5. Procédé selon l'une des revendications 1 à 4 pour décoder un flux vidéo, comprenant en outre une recherche dans le flux vidéo pour chercher le début de l'image suivante, sans avoir encore décodé l'image. 25
6. Procédé selon l'une des revendications 1 à 4 pour encoder un flux vidéo, dans lequel les données des images encodées simultanément (A, B, C) sont concaténées de manière à produire un flux binaire d'image séquentiel. 30
7. Appareil vidéo comprenant : 35
  - une unité de traitement (11) pour réaliser un processus d'encodage/décodage de données vidéo ; 40
  - une première mémoire (12) raccordée à l'unité de traitement (11) pour stocker des données d'image requises pour l'encodage/décodage des données vidéo ; 45
  - une deuxième mémoire (13), dans laquelle le flux vidéo comporte une pluralité d'images (A, B, C) ;

dans lequel la première mémoire (12) est prévue pour stocker un sous-ensemble des données d'images de la deuxième mémoire (13) dans la première mémoire (12), et dans lequel l'unité de traitement (11) est conçue pour encoder/décoder simultanément plus d'une image (B, C) du flux vidéo en accédant audit sous-ensemble de données d'images dans la première mémoire (12), **caractérisé par le fait que** l'encodage/décodage simultané est réalisé 50

55

grâce à un partage d'accès à au moins une image, dans lequel la première mémoire (12) est en outre prévue pour permettre des accès synchronisés pour garantir que les accès à la première mémoire (12) se produisent sur des positions d'image approximativement égales dans l'image partagée (A), dans lequel la synchronisation est fondée sur une position verticale dans les images respectives (A, B) et dans lequel les écarts de synchronisation sont fondés sur la composante verticale maximum du vecteur de déplacement (M).

8. Appareil de traitement vidéo selon la revendication 7, ayant de multiples unités de traitement (11x, 11y, 11z), fonctionnant chacune en même temps et traitant une seule image (A, B, C), comprenant en outre une pluralité de premières mémoires (12x, 12y, 12z), chacune étant associée à une unité de traitement correspondante (11x, 11y, 11z), dans lequel une unité de communication (14) est disposée entre les premières mémoires (12x, 12y, 12z) pour fournir un accès à des données dans une première mémoire qui n'est pas directement associée à l'unité de traitement (11x, 11y, 11z) qui a besoin des données.

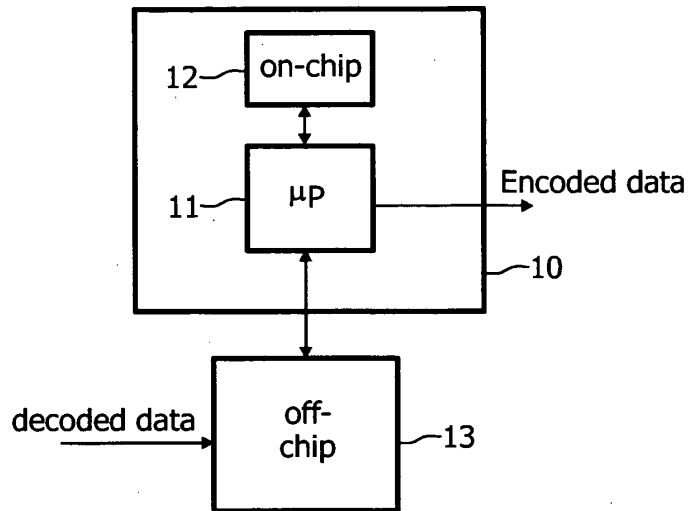


FIG. 1

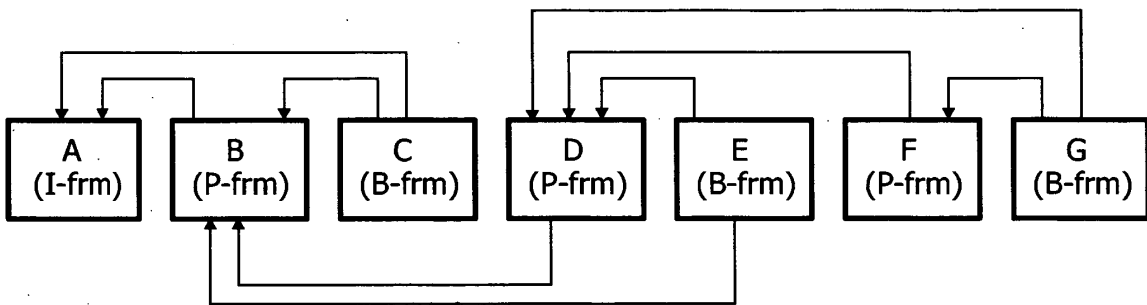


FIG. 2

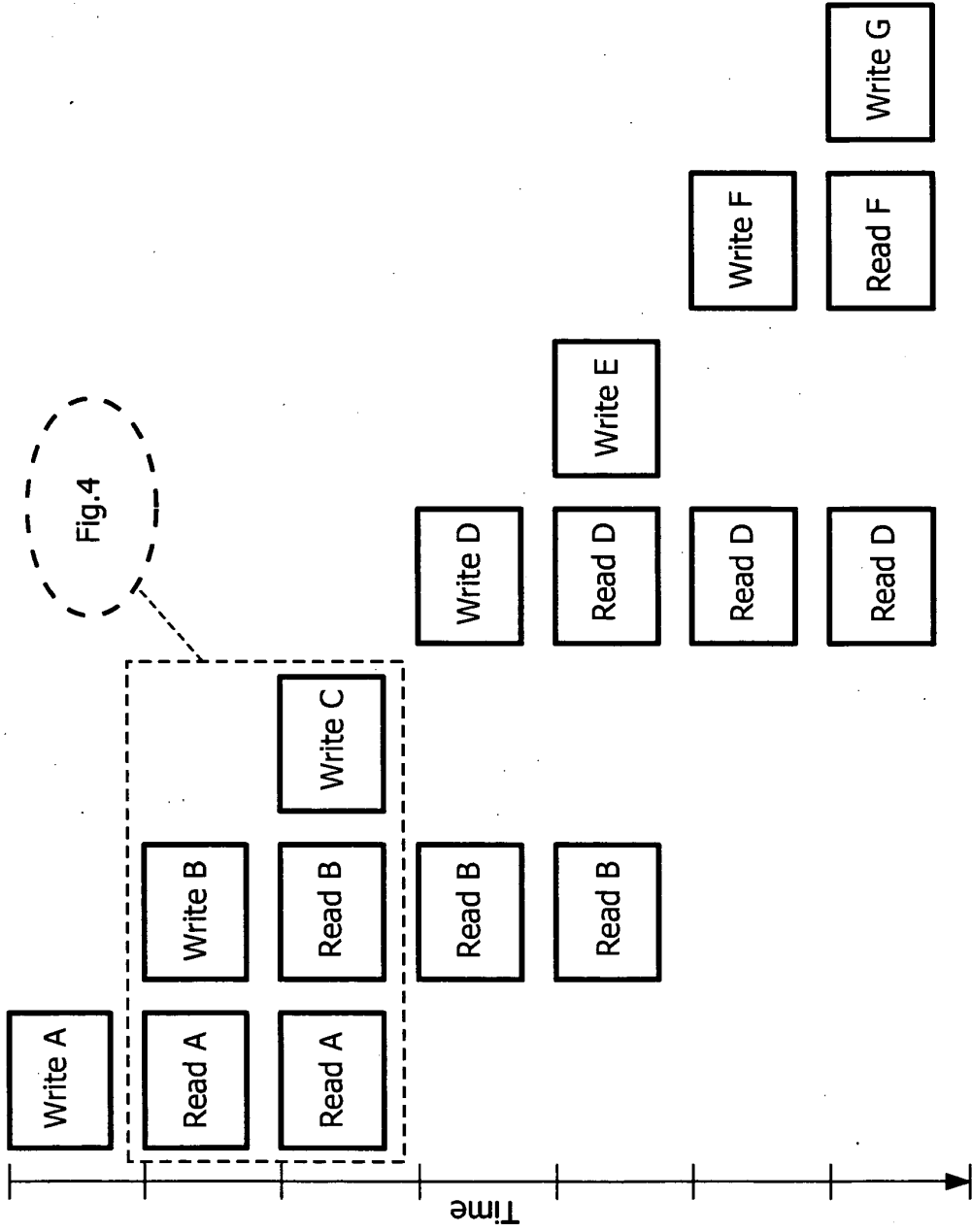


FIG. 3

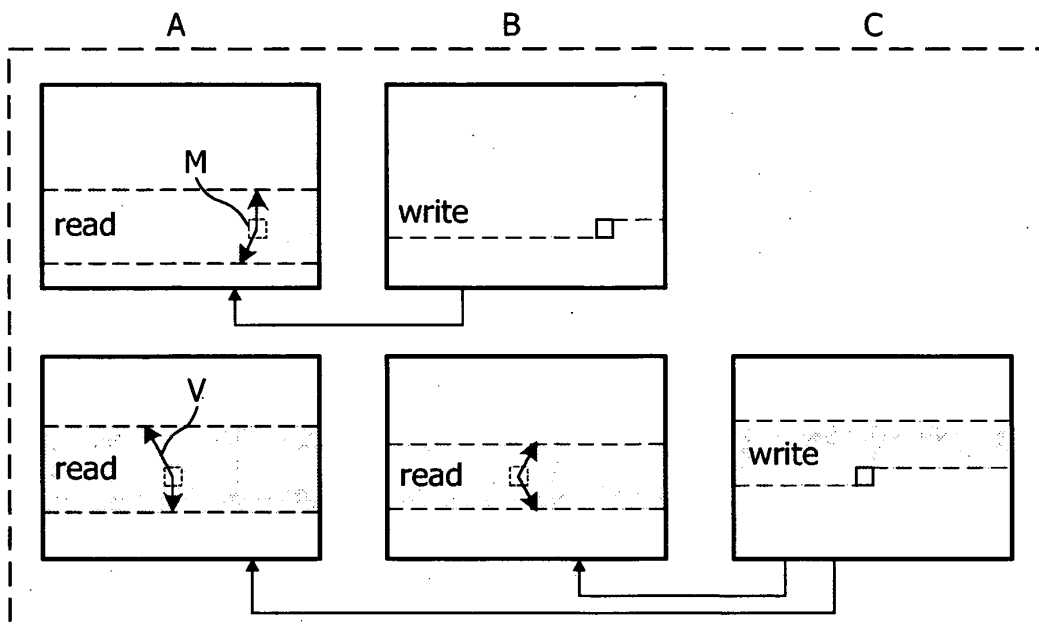


FIG. 4

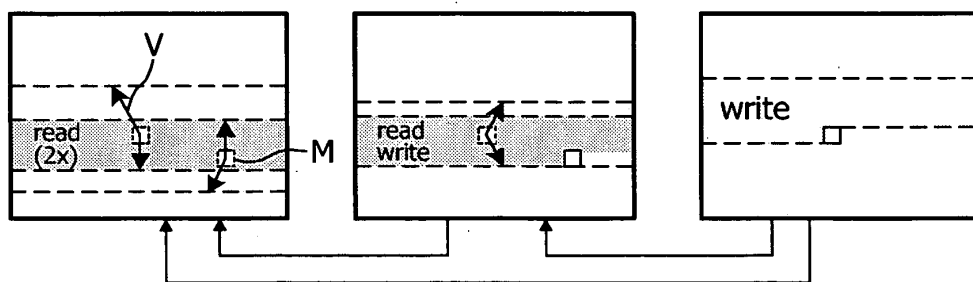


FIG. 5

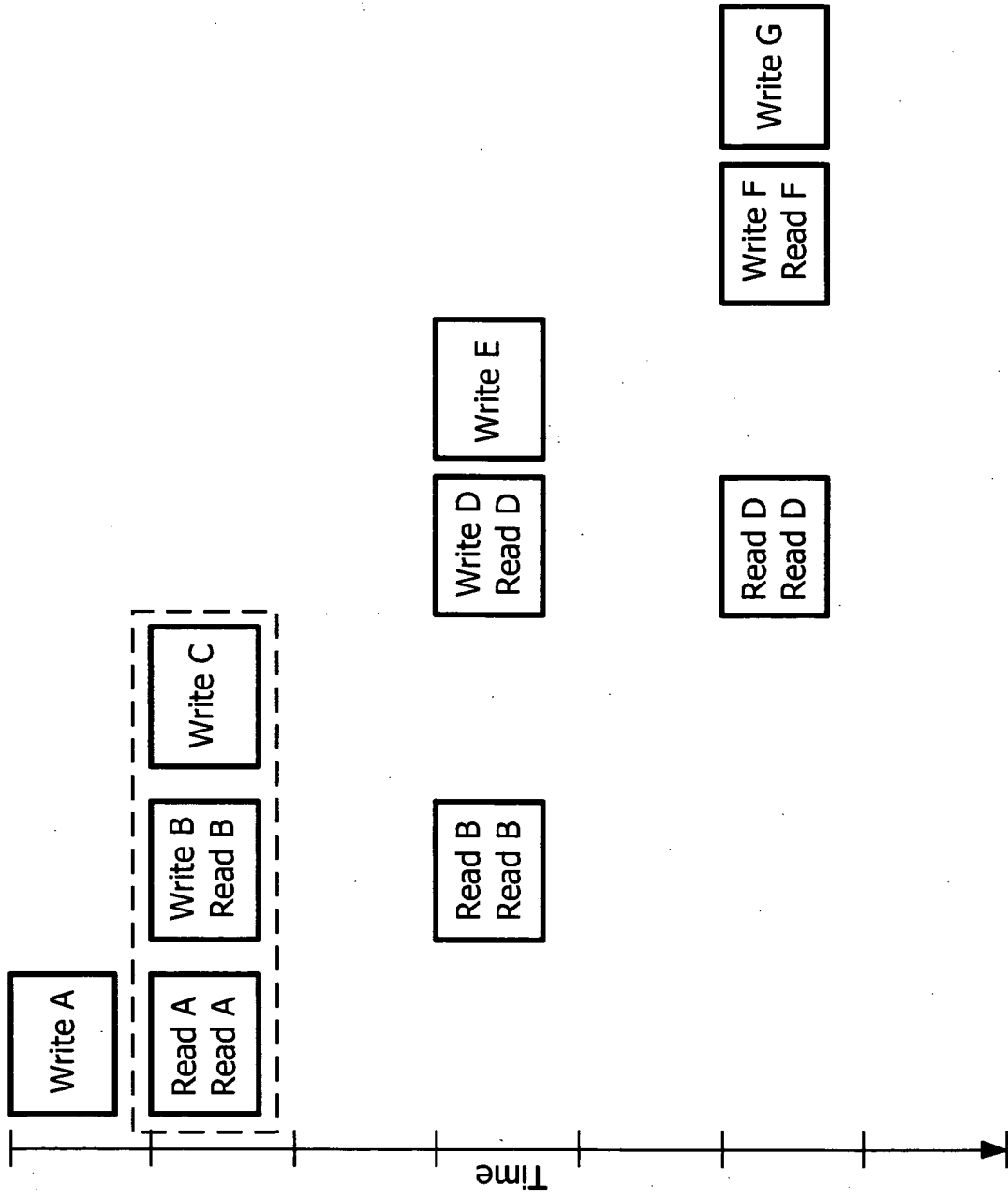


FIG. 6

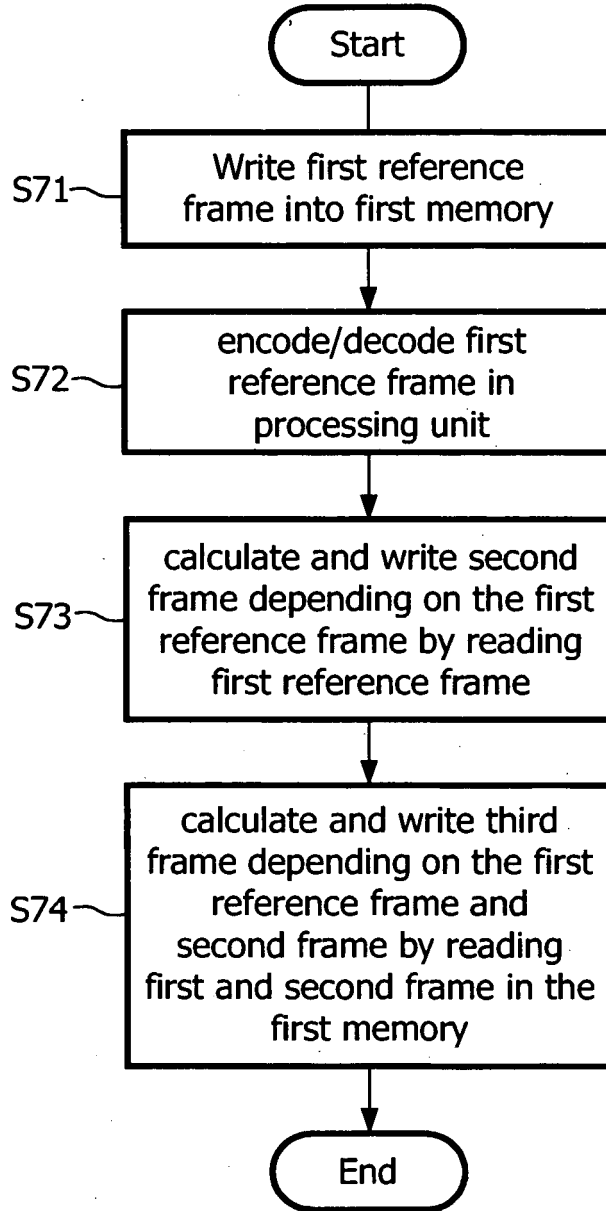


FIG. 7

**REFERENCES CITED IN THE DESCRIPTION**

*This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.*

**Patent documents cited in the description**

- US 6263112 B1 [0014]

**Non-patent literature cited in the description**

- **Ge. S. et al.** Efficient Multithreading implementation of H.264 Encoder on Intel Hyper-Threading architecture. *Proceedings of the fourth international conference on information*, December 2003 [0011]
- **Tsuo, H.K. et al.** An efficient macroblock sync decoding architecture for MPEG-4 FGS Video. *proceedings of the 45th midwest symposium, on circuits and systems*, 04 August 2002 [0012]
- **Chen Y-K et al.** *Media applications on Hyper-Threading Technology* [0013]