

CLAIMS

1. A method of video decoding, the method comprising:

receiving a video bitstream including coded data for a block of video data;

deriving, from the video bitstream, a block partitioning structure corresponding to a block
 5 partitioning process including a binary tree partitioning process for the block of video data,
 wherein the block partitioning structure represents partitioning the block of video data into final
 sub-blocks, and wherein said deriving the block partitioning structure for the block comprises;

parsing a first indicator associated with a current block;

parsing a second indicator being used for the current block to indicate a binary
 10 tree partition type associated with the binary tree partition when the binary tree partition
 is applied to the current block, wherein the second indicator is inferred as the symmetric
 vertical partition when height of the current block reaches minimum allowed height, and
 the second indicator is inferred to be the symmetric horizontal partition when width of
 the current block reaches minimum allowed width; and

15 determining whether the binary tree partitioning process applies a binary tree
 partition to the current block based on a value of the first indicator, wherein when it is
 determined that the binary tree partition is applied to the current block, the current block
 is always split into two sub-blocks;

decoding the final sub-blocks based on the video bitstream; and

20 decoding the block of video data based on the final sub-blocks decoded according to the
 binary tree partitioning structure derived.

2. The method of Claim 1, wherein binary tree partition types associated with the binary
 tree partition comprises symmetric horizontal partition and symmetric vertical partition.

3. The method of Claim 1, wherein binary tree partition types associated with the binary
 25 tree partition comprises symmetric horizontal partition and symmetric vertical partition,
 asymmetric horizontal partition with narrower upper side, asymmetric horizontal partition with
 narrower lower side, asymmetric vertical partition with narrower left side and asymmetric
 vertical partition with narrower right side.

4. The method of Claim 1, wherein if the binary tree partition type consists of symmetric

horizontal partition and symmetric vertical partition, the second indicator corresponds to a 1-bit flag to indicate the symmetric horizontal partition or the symmetric vertical partition.

5 5. The method of Claim 4, wherein the minimum allowed height and the minimum allowed width are specified in high level syntax including sequence parameter set (SPS), picture parameter set (PPS) or slice header.

6. The method of Claim 1, wherein the first indicator is inferred as not applying binary tree partition when height and width of the current block both reach minimum allowed value, or tree depth reaches maximum allowed depth.

10 7. The method of Claim 1, wherein the block of video data corresponds to a coding tree unit (CTU) and the final sub-blocks associated with leaf nodes of the block partitioning structure correspond to coding units (CUs).

8. The method of Claim 7, wherein all CUs are used for prediction process and/or transform process respectively without any further explicit partitioning process.

15 9. The method of Claim 1, wherein the block of video data and the final sub-blocks correspond to picture and coding tree units (CTUs) respectively, slice and CTUs respectively, coding unit (CU) and prediction units (PUs) respectively, coding unit (CU) and transform units (TUs) respectively, or prediction unit (PU) and transform units (TUs) respectively.

20 10. The method of Claim 1, wherein the block of video data for luma component and non-luma components in an I-slice uses individual block partitioning process or the block of video data for two chroma components in an I-slice uses individual block partitioning process.

11. The method of Claim 1, wherein the block partitioning process further includes a quadtree partitioning process, wherein when the quadtree partitioning process decides to apply quadtree partition to one given block, said one given block is always split into four sub-blocks.

25 12. The method of Claim 11, wherein the quadtree partitioning process is firstly applied to the block of video data recursively before the binary tree partitioning process to generate quadtree leaf nodes, and the binary tree partitioning process is applied to one of the quadtree leaf nodes or not based on the first indicator associated with said one of the quadtree leaf nodes.

13. The method of Claim 412, wherein the quadtree partitioning process for a selected

node is implicitly terminated when the selected node reaches a minimum allowed quadtree leaf node size or quadtree depth associated with the selected node reaches a maximum allowed quadtree depth.

14. The method of Claim 13, wherein the minimum allowed quadtree leaf node size, the
5 maximum allowed quadtree depth are specified in high level syntax including sequence parameter set (SPS), picture parameter set (PPS) or slice header.

15. The method of Claim 11, wherein the block of video data and the final sub-blocks correspond to coding tree unit (CTU) and coding units (CUs) respectively.

16. The method of Claim 15, wherein all CUs are used for prediction process and/or
10 transform process respectively without any further explicit partitioning process.

17. The method of Claim 11, wherein the block of video data and the final sub-blocks correspond to picture and coding tree units (CTUs) respectively, slice and CTUs respectively, coding unit (CU) and prediction units (PUs) respectively, coding unit (CU) and transform units (TUs) respectively, or prediction unit (PU) and transform units (TUs) respectively.

18. A method of video encoding, the method comprising:

receiving input data associated with a block of video data;

determining a block partitioning structure corresponding to a block partitioning process including a binary tree partitioning process for the block of video data, wherein the block partitioning structure represents partitioning the block of video data into final sub-blocks,
20 wherein said determining the block partitioning structure for the block comprises:

incorporating a first indicator associated with a current block into a video bitstream, wherein the first indicator indicates whether the binary tree partitioning process applies a binary tree partition is applied to the current block, the current block is always split into two sub-blocks; and wherein the first indicator is inferred as not
25 applying binary tree partition when height and width of the current block both reach minimum allowed value, or tree depth reaches maximum allowed depth; and

applying an encoding process comprising prediction process, transform process or both to the block of video data by applying the encoding process at a level corresponding to the final sub-blocks.

19. The method of claim 18, wherein the binary tree partition types associated with the binary tree partition comprises at least two of symmetric horizontal partition and symmetric vertical partition, asymmetric horizontal partition with narrower upper side, asymmetric horizontal partition with narrower lower side, asymmetric vertical partition with narrower left side and asymmetric vertical partition with narrower right side.

20. The method of Claim 18, wherein a second indicator is used for the current block to indicate a binary tree partition type associated with the binary tree partition when the binary tree partition is applied to the current block.

21. The method of Claim 18, wherein the block of video data corresponds to a coding tree unit (CTU) and the final sub-blocks associated with leaf nodes of the block partitioning structure correspond to coding unit (CUs).

22. The method of Claim 21, wherein all CUs are used for prediction process and/or transform process respectively without any further explicit partitioning process.

23. The method of Claim 18, wherein the block partitioning process further includes a quadtree partitioning process, wherein when the quadtree partitioning process decides to apply quadtree partition to one given block, said one given block is always split into four sub-blocks, wherein the quadtree partitioning process is firstly applied to the block of video data recursively before the binary tree partitioning process to generate quadtree leaf nodes, and the binary tree partitioning process is applied to one of the quadtree leaf nodes or not based on the first indicator associated with said one of the quadtree leaf nodes.

24. The method of Claim 23, wherein the block of video data and the final sub-blocks correspond to coding tree unit (CTU) and coding units (CUs) respectively, and all CUs are used for prediction process and/or transform process respectively without any further explicit partitioning process.