

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7514330号
(P7514330)

(45)発行日 令和6年7月10日(2024.7.10)

(24)登録日 令和6年7月2日(2024.7.2)

(51)国際特許分類	F I
H 0 4 N 19/70 (2014.01)	H 0 4 N 19/70
H 0 4 N 19/186 (2014.01)	H 0 4 N 19/186
H 0 4 N 19/593 (2014.01)	H 0 4 N 19/593

請求項の数 13 (全150頁)

(21)出願番号	特願2022-573429(P2022-573429)	(73)特許権者	520476341
(86)(22)出願日	令和3年5月28日(2021.5.28)		北京字節跳動網絡技術有限公司
(65)公表番号	特表2023-527453(P2023-527453 A)		Beijing Bytedance Network Technology Co., Ltd.
(43)公表日	令和5年6月28日(2023.6.28)		中華人民共和國100041北京市石景山区実興大街30号院3号楼2層B-0035房間
(86)国際出願番号	PCT/CN2021/096707		Room B-0035, 2/F, No.3 Building, No.30, Shixing Road, Shijingshan District Beijing 100041 China
(87)国際公開番号	WO2021/244420		
(87)国際公開日	令和3年12月9日(2021.12.9)		
審査請求日	令和4年12月14日(2022.12.14)		
(31)優先権主張番号	PCT/CN2020/093641	(73)特許権者	520477474
(32)優先日	令和2年5月31日(2020.5.31)		バイトダンス インコーポレイテッド
(33)優先権主張国・地域又は機関	中国(CN)		最終頁に続く

(54)【発明の名称】 ローカルデュアルツリーモードタイプ定義を伴うパレットモード

(57)【特許請求の範囲】

【請求項1】

ビデオデータを処理する方法であって、
 ルールに従い、ビデオブロックを含むビデオと前記ビデオのビットストリームとの間の変換を実行するステップ、を含み、
 前記ビデオブロックは、1つ以上のコーディングブロックを含むコーディングツリーノードであり、
 前記ルールは、以下のコーディングモード：イントラ予測コーディングモード、イントラブロックコピー（IBC）モード、及びパレットコーディングモード、のうちの1つ以上が、前記1つ以上のコーディングブロックの1つ以上のルマブロックにより使用できることを指定し、ローカルデュアルツリーが前記コーディングツリーノード内の前記1つ以上のコーディングブロックに適用されるとき、前記イントラ予測コーディングモードは、前記1つ以上のコーディングブロックの1つ以上のクロマブロックに適用され、前記ローカルデュアルツリーが適用される前記コーディングツリーノード内の前記1つ以上のコーディングブロックに対して、インター予測コーディングモードが許可されない、方法。

【請求項2】

前記ルールは、前記イントラ予測コーディングモード、前記IBCモード、及び前記パレットコーディングモードのうちの1つより多くのコーディングモードが、複数のルマブロックに適用できることを指定し、前記1つより多くのコーディングモードは、前記イントラ予測コーディングモード、前記IBCモード、及び前記パレットコーディングモードのう

ちのいずれか 2 つ以上を含む、請求項 1 に記載の方法。

【請求項 3】

前記ルールは、前記パレットコーディングモードがコーディングブロックに対して有効であるかどうかを示す第 1 シンタックス要素の存在が、前記コーディングブロックの名前付き変数モードタイプの値が特定のモードタイプであることに少なくとも基づき、前記特定のモードタイプは、MODE_TYPE_INTRA、又はMODE_TYPE_ALLであり、MODE_TYPE_INTERではない、請求項 1 又は 2 に記載の方法。

【請求項 4】

MODE_TYPE_INTERは、前記インター予測コーディングモードのみが前記コーディングブロックに対して使用できることを指定し、

MODE_TYPE_INTRAは、前記イントラ予測コーディングモード、前記IBCモード、及び前記パレットコーディングモードのみが、前記コーディングブロックに対して使用できることを指定し、

MODE_TYPE_ALLは、前記インター予測コーディングモード、前記イントラ予測コーディングモード、前記IBCモード、及び前記パレットコーディングモードが前記コーディングブロックに対して使用できることを指定する、請求項 3 に記載の方法。

【請求項 5】

前記名前付き変数モードタイプは、名前付き変数modeTypeCondition、及び前記インター予測コーディングモードのみが許可されるかどうかを示す第 2 シンタックス要素non_inter_flagに基づく、請求項 3 又は 4 に記載の方法。

【請求項 6】

MODE_TYPE_INTER、MODE_TYPE_INTRA、及びMODE_TYPE_ALLは、パーティション処理でも使用され、前記パーティション処理は、4 分割処理、2 分割処理、又は 3 分割処理のうちの少なくとも 1 つを含む、請求項 3 ~ 5 のいずれか一項に記載の方法。

【請求項 7】

前記ルールは、一般的制約情報シンタックス構造の中の第 3 シンタックス要素が、シーケンスパラメータセットの中の第 4 シンタックス要素の値を制御することを指定し、前記第 4 シンタックス要素は、サブピクチャ情報がコーディングレイヤビデオシーケンスに存在するかどうか、及び 1 つより多くのサブピクチャが前記コーディングレイヤビデオシーケンスの各ピクチャに含まれるかどうかを示す、請求項 1 ~ 6 のいずれか一項に記載の方法。

【請求項 8】

前記ルールは、一般的制約情報シンタックス構造の中の第 5 シンタックス要素が、シーケンスパラメータセットの中の第 6 シンタックス要素の値を制御することを指定し、前記第 6 シンタックス要素は、ルマサンプルの中の最大変換サイズが 64 に等しいかどうかを示す、請求項 1 ~ 7 のいずれか一項に記載の方法。

【請求項 9】

前記変換を実行するステップは、前記ビデオを前記ビットストリームに符号化することを含む、請求項 1 ~ 8 のいずれか一項に記載の方法。

【請求項 10】

前記変換を実行するステップは、前記ビットストリームから前記ビデオを復号することを含む、請求項 1 ~ 8 のいずれか一項に記載の方法。

【請求項 11】

ビデオデータを処理する機器であって、プロセッサと、命令を有する非一時的メモリと、を含み、前記命令は前記プロセッサにより実行されると、前記プロセッサに、ルールに従い、ビデオブロックを含むビデオと前記ビデオのビットストリームとの間の変換を実行させ、

前記ビデオブロックは、1 つ以上のコーディングブロックを含むコーディングツリーノードであり、

前記ルールは、以下のコーディングモード：イントラ予測コーディングモード、イント

10

20

30

40

50

ラブロックコピー（IBC）モード、及びパレットコーディングモード、のうちの1つ以上が、前記1つ以上のコーディングブロックの1つ以上のルマブロックにより使用できることを指定し、ローカルデュアルツリーが前記コーディングツリーノード内の前記1つ以上のコーディングブロックに適用されるとき、前記イントラ予測コーディングモードは、前記1つ以上のコーディングブロックの1つ以上のクロマブロックに適用され、前記ローカルデュアルツリーが適用される前記コーディングツリーノード内の前記1つ以上のコーディングブロックに対して、インター予測コーディングモードが許可されない、機器。

【請求項12】

命令を格納している非一時的コンピュータ可読記憶媒体であって、前記命令は、プロセッサに、

10

ルールに従い、ビデオブロックを含むビデオと前記ビデオのビットストリームとの間の変換を実行させ、

前記ビデオブロックは、1つ以上のコーディングブロックを含むコーディングツリーノードであり、

前記ルールは、以下のコーディングモード：イントラ予測コーディングモード、イントラブロックコピー（IBC）モード、及びパレットコーディングモード、のうちの1つ以上が、前記1つ以上のコーディングブロックの1つ以上のルマブロックにより使用できることを指定し、ローカルデュアルツリーが前記コーディングツリーノード内の前記1つ以上のコーディングブロックに適用されるとき、前記イントラ予測コーディングモードは、前記1つ以上のコーディングブロックの1つ以上のクロマブロックに適用され、前記ローカルデュアルツリーが適用される前記コーディングツリーノード内の前記1つ以上のコーディングブロックに対して、インター予測コーディングモードが許可されない、非一時的コンピュータ可読記憶媒体。

20

【請求項13】

ビデオのビットストリームを格納する方法であって、

ルールに従い、ビデオブロックを含むビデオのビットストリームを生成するステップと、前記ビットストリームを非一時的コンピュータ可読記録媒体に格納するステップと、
_を含み、

前記ビデオブロックは、1つ以上のコーディングブロックを含むコーディングツリーノードであり、

30

前記ルールは、以下のコーディングモード：イントラ予測コーディングモード、イントラブロックコピー（IBC）モード、及びパレットコーディングモード、のうちの1つ以上が、前記1つ以上のコーディングブロックの1つ以上のルマブロックにより使用できることを指定し、ローカルデュアルツリーが前記コーディングツリーノード内の前記1つ以上のコーディングブロックに適用されるとき、前記イントラ予測コーディングモードは、前記1つ以上のコーディングブロックの1つ以上のクロマブロックに適用され、前記ローカルデュアルツリーが適用される前記コーディングツリーノード内の前記1つ以上のコーディングブロックに対して、インター予測コーディングモードが許可されない、方法。

【発明の詳細な説明】

【技術分野】

40

【0001】

[関連出願]

本願は、国際特許出願番号第PCT/CN2020/093641号、2020年5月31日出願、の優先権及び利益を請求する国際特許出願番号第PCT/CN2021/096707号、2021年5月28日出願、に基づく。前述の全ての特許出願は参照によりそれらの全体がここに組み込まれる。

【0002】

[技術分野]

本願明細書は、画像及びビデオコーディング及び復号に関する。

【背景技術】

50

【 0 0 0 3 】

デジタルビデオは、インターネット及び他のデジタル通信ネットワーク上で最大の帯域幅使用を占める。ビデオを受信及び表示可能な接続されたユーザ装置の数が増加するにつれ、デジタルビデオ使用のための帯域幅要求は増大し続けることが予想される。

【 発明の概要 】

【 0 0 0 4 】

本願明細書は、コーディング表現の復号に有用な制御情報を使用してビデオのコーディング表現を処理するために、ビデオエンコーダ及びデコーダによって使用できる技術を開示する。

【 0 0 0 5 】

1つの例示的な態様では、ビデオ処理方法が開示される。前記方法は、ルールに従って、1つ以上のビデオピクチャを含むビデオと、前記ビデオのビットストリームとの間の変換を実行するステップを含み、前記ルールは、一般的な制約情報シンタックス構造の第1シンタックス要素が、パラメータセット又はヘッダ内の1つ以上のシンタックス要素の存在又は前記1つ以上のシンタックス要素の1つ以上の値を制御することを指定する。

10

【 0 0 0 6 】

別の例示的な態様では、ビデオ処理方法が開示される。前記方法は、ルールに従って、ビデオブロックを含むビデオと前記ビデオのビットストリームとの間の変換を実行するステップを含み、前記ビデオブロックは、1つ以上のコーディングユニットを含むコーディングツリーノードであり、前記ルールは、前記ビデオブロックのコーディング情報が、前記ビデオブロックの1つ以上のコーディングユニットに対してコーディングモードが有効になっているかどうかを示すことを指定する。

20

【 0 0 0 7 】

別の例示的な態様では、ビデオ処理方法が開示される。前記方法は、ルールに従って、ビデオブロックを含むビデオピクチャを含むビデオと前記ビデオのビットストリームとの間の変換を実行するステップを含み、前記ルールは、前記ビデオブロックに対してパレットコーディングモードが有効であることを示すコーディングモードタイプを使用したパーティション処理を使用して、前記ビデオブロックを選択的にコーディングブロックにパーティションすることを指定実行するステップを含む。

【 0 0 0 8 】

別の例示的な態様では、ビデオ処理方法が開示される。前記方法は、ビデオのビデオ領域と前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、前記ビデオ内の色成分のスケーリングリストが、シーケンスパラメータセットのシンタックスフィールド値とは独立して適応パラメータセットに含まれるかどうかをフラグが示すことを指定する。

30

【 0 0 0 9 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、ビデオのビデオ領域と前記ビデオ領域のコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、各適応パラメータセットについて、前記ビデオのクロマ制約によりクロマ関連シンタックス要素が省略されるように、1つ以上の適応パラメータセットが前記コーディング表現に含まれることを指定する。

40

【 0 0 1 0 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、1つ以上のビデオユニットを含む1つ以上のビデオ領域を含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、第1変換コーディングシンタックスフィールドが、ビデオ領域のビデオユニットのレベルの前記コーディング表現に含まれるかどうか、及び/又はその値が、前記ビデオ領域のレベルの第2変換コーディングシンタックスフィールドの値に依存するかどうかを指定する。

50

【 0 0 1 1 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、1つ以上のビデオ領域を含むビデオであって、各ビデオ領域が1つ以上のビデオユニットを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、ビデオユニットレベルのフラグが、前記変換に対して量子化パラメータの差分シグナリングが有効かどうかを制御することを指定する。

【 0 0 1 2 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、1つ以上のビデオ領域を含むビデオであって、各ビデオ領域が1つ以上のビデオユニットを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、サブピクチャの数を示すピクチャレベルの第1フラグと、サブピクチャ内のスライスを示すサブピクチャレベルの第2フラグの解釈を指定する。

10

【 0 0 1 3 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、ビデオピクチャに関連するピクチャパラメータセット内のフィールドが、異なる高さ又は幅の複数のタイル行又は列に分割されることを指定する。

20

【 0 0 1 4 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、適応パラメータセットが適応ループフィルタリングの表示を除外する場合のビデオ領域への適応ループフィルタリングの適用可能性が第2ルールに基づくことを指定する。

【 0 0 1 5 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、幅と高さがビデオの最大幅と最大高さであるピクチャについて、ピクチャパラメータセット内の適合ウィンドウパラメータの明示的なシグナリングをスキップすることを指定する。

30

【 0 0 1 6 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、一般制約情報(GCI)シンタックス構造内のシンタックスフィールドが、前記コーディング表現内のパラメータセット又はヘッダ内の1つ以上のシンタックス要素の値に対する1つ以上の制約を制御することを指定する。

40

【 0 0 1 7 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマッ

50

トルールは、前記ビデオのコーディングツリーノードについて、ビデオユニットに対応する復号情報がルールに従って導出可能な変数の値を制御することを指定する。

【 0 0 1 8 】

別の例示的な態様では、別のビデオ処理方法が開示される。前記方法は、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、前記コーディング表現が、前記コーディングツリーノード内のビデオユニットの表現に対して3つの予測モードのいずれかが許可されるかどうかを示す値を有するコーディングツリーノードのシンタックス要素を含むことを指定する。

10

【 0 0 1 9 】

更に別の例示的な態様では、ビデオエンコーダ機器が開示される。前記ビデオエンコーダは、上記の方法を実装するように構成されたプロセッサを含む。

【 0 0 2 0 】

更に別の例示的な態様では、ビデオデコーダ機器が開示される。前記ビデオデコーダは、上記の方法を実装するように構成されたプロセッサを含む。

【 0 0 2 1 】

更に別の例示的な態様では、コードが格納されたコンピュータ可読媒体が開示される。前記コードは、ここで説明する方法のうちの1つをプロセッサ実行可能コードの形式で具体化したものである。

20

【 0 0 2 2 】

これら及び他の特徴は、本願明細書を通じて記載される。

【 図面の簡単な説明 】

【 0 0 2 3 】

【 図 1 】 例示的なビデオ処理システムのブロック図である。

【 0 0 2 4 】

【 図 2 】 ビデオ処理機器のブロック図である。

【 0 0 2 5 】

【 図 3 】 ビデオ処理の例示的な方法のフローチャートである。

【 0 0 2 6 】

【 図 4 】 本開示の幾つかの実施形態によるビデオコーディングシステムを示すブロック図である。

30

【 0 0 2 7 】

【 図 5 】 本開示の幾つかの実施形態によるエンコーダを示すブロック図である。

【 0 0 2 8 】

【 図 6 】 本開示の幾つかの実施形態によるデコーダを示すブロック図である。

【 0 0 2 9 】

【 図 7 】 ビデオ処理の例示的な方法のフローチャートである。

【 図 8 】 ビデオ処理の例示的な方法のフローチャートである。

【 図 9 】 ビデオ処理の例示的な方法のフローチャートである。

40

【 発明を実施するための形態 】

【 0 0 3 0 】

章見出しは、本願明細書において理解を容易にするために使用され、各章で開示される実施形態及び技術の適用可能性をその章にのみ限定するものではない。更に、H. 266の用語は、一部の説明において、単に理解しやすくするために使用されており、開示される技術の範囲を制限するために使用されていない。そのため、ここで説明する技術は他のビデオコーデックプロトコル及び設計にも適用できる。

【 0 0 3 1 】

1. 序論

本願明細書は、ビデオコーディング技術に関する。具体的には、ビデオコーディングに

50

おけるSH、PPS、APS、及びGCIシンタックス要素の設計に関するものである。これらのアイデアは、例えば開発中のVersatile Video Coding (VVC) のような、多層ビデオコーディングをサポートする任意のビデオコーディング標準又は非標準ビデオコーデックに、個別に又は様々な組み合わせで適用することができる。

2. 略語

APS : Adaptation Parameter Set (適応パラメータセット)	
AU : Access Unit (アクセスユニット)	
AUD : Access Unit Delimiter (アクセスユニットデリミタ)	
AVC : Advanced Video Coding (高度ビデオコーディング)	
CLVS : Coded Layer Video Sequence (コーディングレイヤビデオシーケンス)	10
CPB : Coded Picture Buffer (コーディングピクチャバッファ)	
CRA : Clean Random Access (クリーンランダムアクセス)	
CTU : Coding Tree Unit (コーディングツリーユニット)	
CVS : Coded Video Sequence (コーディングビデオシーケンス)	
DPB : Decoded Picture Buffer (復号ピクチャバッファ)	
DPS : Decoding Parameter Set (復号パラメータセット)	
EOB : End Of Bitstream (エンドオブビットストリーム)	
EOS : End Of Sequence (エンドオブシーケンス)	
GCI : General Constraint Information (一般的制約情報)	
GDR : Gradual Decoding Refresh (漸次デコーダリフレッシュ)	20
HEVC : High Efficiency Video Coding (高効率ビデオコーディング)	
HRD : Hypothetical Reference Decoder (仮説参照デコーダ)	
IDR : Instantaneous Decoding Refresh (瞬時デコードリフレッシュ)	
JEM : Joint Exploration Model (共同探索モデル)	
MCTS : Motion - Constrained Tile Sets (動き制約されたタイルセット)	
NAL : Network Abstraction Layer (ネットワーク抽象化レイヤ)	
OLS : Output Layer Set (出力レイヤセット)	
PH : Picture Header (ピクチャヘッダ)	
PPS : Picture Parameter Set (ピクチャパラメータセット)	
PTL : Profile, Tier and Level (プロファイル、ティア、及びレベル)	30
PU : Picture Unit (ピクチャユニット)	
RBSP : Raw Byte Sequence Payload (未処理バイトシーケンスペイロード)	
SEI : Supplemental Enhancement Information (補足拡張情報)	
SH : Slice Header (スライスヘッダ)	
SPS : Sequence Parameter Set (シーケンスパラメータセット)	
SVC : Scalable Video Coding (拡張可能ビデオコーディング)	
VCL : Video Coding Layer (ビデオコーディングレイヤ)	
VPS : Video Parameter Set (ビデオパラメータセット)	
VTM : VVC Test Model (VVCテストモデル)	
VUI : Video Usability Information (ビデオ利用可能性情報)	40
VVC : Versatile Video Coding (バーサタイルビデオコーディング)	

【 0 0 3 2 】

3. 最初の議論

ビデオコーディング規格は、主によく知られたITU - T及びISO/IEC規格の発展を通じて進化している。ITU - TはH . 2 6 1及びH . 2 6 3を策定し、ISO/IECはMPEG - 1及びMPEG - 4 ビジュアルを策定し、及び2つの組織は共同でH . 2 6 2 / MPEG - 2 ビデオ、及びH . 2 6 4 / MPEG - 4 アドバンスドビデオコーディング (Advanced Video Coding) 及びH . 2 6 5 / HEVC規格を策定した。H . 2 6 2以降、ビデオコーディング規格は、ハイブリッドビデオコーディング構造に基づき、ここでは時間予測及び変換コーディングが利用される。HEVCより先の将来のビデオコーディング技

術を開発するために、共同ビデオ探索チーム（Joint Video Exploration Team（JVET））が2015年にVCEG及びMPEGにより共同で設立された。それ以来、多くの新しい方法がJVETにより採用され、共同探索モデル（Joint Exploration Model（JEM））と呼ばれる参照ソフトウェアに取り入れられてきた。JVET会議が、現在、四半期毎に行われており、新しいコーディング規格は、HEVCと比べて50%のビットレートの削減を目標としている。新しいビデオコーディング規格は、2018年4月のJVET会議においてVersatile Video Coding（VVC）として公式に命名され、そのとき、VVCテストモデル（VVC test model（VTM））の第1バージョンが公表された。VVC標準化に貢献する絶え間ない努力により、JVET会議の度にVVC規格に新しいコーディング技術が採用されている。VVCワーキングドラフト及びテストモデルVTMは、従って、毎回の会議の後に更新される。VVCプロジェクトは、目下、2020年7月の会議における技術的完成（FDIS）を目指している。

10

【0033】

3.1 GCIのシンタックスとセマンティクス

最新のVVCドラフトテキストでは、GCIシンタックスとセマンティクスは次の通りである。

20

30

40

50

【表 1】

general_constraint_info() {	Descriptor	
general_progressive_source_flag	u(1)	
general_interlaced_source_flag	u(1)	
general_non_packed_constraint_flag	u(1)	
general_frame_only_constraint_flag	u(1)	
general_non_projected_constraint_flag	u(1)	
intra_only_constraint_flag	u(1)	10
max_bitdepth_constraint_idc	u(4)	
max_chroma_format_constraint_idc	u(2)	
no_res_change_in_clvs_constraint_flag	u(1)	
one_tile_per_pic_constraint_flag	u(1)	
one_slice_per_pic_constraint_flag	u(1)	
one_subpic_per_pic_constraint_flag	u(1)	
no_qtbt_dual_tree_intra_constraint_flag	u(1)	
no_partition_constraints_override_constraint_flag	u(1)	
no_sao_constraint_flag	u(1)	20
no_alf_constraint_flag	u(1)	
no_ccalf_constraint_flag	u(1)	
no_joint_cbr_constraint_flag	u(1)	
no_ref_wraparound_constraint_flag	u(1)	
no_temporal_mvp_constraint_flag	u(1)	
no_sbtmvp_constraint_flag	u(1)	
no_amvr_constraint_flag	u(1)	
no_bdof_constraint_flag	u(1)	
no_dmvr_constraint_flag	u(1)	
no_cclm_constraint_flag	u(1)	30
no_mts_constraint_flag	u(1)	
no_sbt_constraint_flag	u(1)	
no_affine_motion_constraint_flag	u(1)	
no_bcw_constraint_flag	u(1)	
no_ibc_constraint_flag	u(1)	
no_ciip_constraint_flag	u(1)	
no_fpel_mmvd_constraint_flag	u(1)	
no_gpm_constraint_flag	u(1)	
no_ladf_constraint_flag	u(1)	40

no_transform_skip_constraint_flag	u(1)
no_bdpcm_constraint_flag	u(1)
no_qp_delta_constraint_flag	u(1)
no_dep_quant_constraint_flag	u(1)
no_sign_data_hiding_constraint_flag	u(1)
no_mixed_nalu_types_in_pic_constraint_flag	u(1)
no_trail_constraint_flag	u(1)
no_stsa_constraint_flag	u(1)
no_rasl_constraint_flag	u(1)
no_radl_constraint_flag	u(1)
no_idr_constraint_flag	u(1)
no_cra_constraint_flag	u(1)
no_gdr_constraint_flag	u(1)
no_aps_constraint_flag	u(1)
while(!byte_aligned())	
gci_alignment_zero_bit	f(1)
num_reserved_constraint_bytes	u(8)
for(i = 0; i < num_reserved_constraint_bytes; i++)	
gci_reserved_constraint_byte[i]	u(8)
}	

10

20

30

general_progressive_source_flag及びgeneral_interlaced_source_flagは次のように解釈される：

- general_progressive_source_flagが1に等しく、general_interlaced_source_flagが0に等しい場合、OlsInScope内のピクチャのソーススキャンタイプはプログレッシブのみとして解釈される必要がある。

- その他の場合、general_progressive_source_flagが0に等しく、general_interlaced_source_flagが1に等しい場合、OlsInScope内のピクチャのソーススキャンタイプはインターレースのみとして解釈される必要がある。

- その他の場合、general_progressive_source_flagが0に等しく、general_interlaced_source_flagが0に等しい場合、OlsInScope内のピクチャのソーススキャンタイプは不明又は指定されないとして解釈される必要がある。

- その他の場合 (general_progressive_source_flagが1、general_interlaced_source_flagが1)、OlsInScope内の各ピクチャのソーススキャンタイプは、フレームフィールド情報SEIメッセージ内のシンタックス要素source_scan_typeを使用してピクチャレベルで示される。general_progressive_source_flagが1に等しく、general_interlaced_source_flagが1に等しい場合、フレームフィールド情報SEIメッセージが各AUに存在することが、ビットストリーム準拠の要件である。

注1：デコーダは、general_progressive_source_flag及びgeneral_interlaced_source_flagの値を無視する場合がある。更に、ピクチャの実際のソーススキャンタイプは

40

50

この仕様の範囲外であり、エンコーダがgeneral_progressive_source_flagとgeneral_interlaced_source_flagの値を選択する方法は指定されていない。

general_non_packed_constraint_flagが1に等しい場合は、OlsInScopeのビットストリームにフレームパッキング配置SEIメッセージが存在しないことを指定する。general_non_packed_constraint_flagが0の場合、このような制約は課されない。

注2：フレームパッキング配置SEIメッセージの存在又は解釈に関連する復号処理要件がないため、デコーダはgeneral_non_packed_constraint_flagの値を無視してよい。

general_frame_only_constraint_flagが1に等しい場合、OlsInScopeがフレームを表すピクチャを伝えることを指定する。general_frame_only_constraint_flagが0に等しい場合、OlsInScopeがフレームを表してよい又は表さなくてもよいピクチャを伝えることを指定する。

10

注3：general_frame_only_constraint_flagに関連する復号処理要件がないため、デコーダはgeneral_frame_only_constraint_flagの値を無視してよい。

general_non_projected_constraint_flagが1に等しい場合は、OlsInScopeのビットストリームに存在する正距円筒投影SEIメッセージ又は一般化キューブマップ投影SEIメッセージが存在しないことを指定する。general_non_projected_constraint_flagが0の場合、このような制約は課されない。

注4：は、正距円筒投影SEIメッセージ及び一般化キューブマップ投影SEIメッセージの存在又は解釈に関連する復号処理要件がないため、デコーダはgeneral_non_projected_constraint_flagの値を無視してよい。

20

intra_only_constraint_flagが1に等しい場合、slice_typeがIに等しいことを指定する。intra_only_constraint_flagが0に等しい場合、このような制約は課されない。

max_bitdepth_constraint_idcは、bit_depth_minus_8が両端を含む0 ~ max_bitdepth_constraint_idcの範囲である必要があることを指定する。

max_chroma_format_constraint_idcは、chroma_format_idcが両端を含む0 ~ max_chroma_format_constraint_idcの範囲である必要があることを指定する。

no_res_change_in_clvs_constraint_flagが1に等しい場合、res_change_in_clvs_allowed_flagが0に等しい必要があることを指定する。no_res_change_in_clvs_constraint_flagが0に等しい場合、このような制約は課されない。

one_tile_per_pic_constraint_flagが1に等しい場合は、各ピクチャが1つのタイルのみを含む必要があることを指定する。one_tile_per_pic_constraint_flagが0に等しい場合、このような制約は課されない。

30

one_slice_per_pic_constraint_flagが1に等しい場合は、各ピクチャが1つのスライスのみを含む必要があることを指定する。one_slice_per_pic_constraint_flagが0に等しい場合、このような制約は課されない。

one_subpic_per_pic_constraint_flagが1に等しい場合は、各ピクチャが1つのサブピクチャのみを含む必要があることを指定する。one_subpic_per_pic_constraint_flagが0に等しい場合、このような制約は課されない。one_slice_per_pic_constraint_flagが1に等しい場合、one_subpic_per_pic_constraint_flagの値は1に等しい必要がある。

40

no_qtbt_dual_tree_intra_constraint_flagが1に等しい場合、qtbt_dual_tree_intra_constraint_flagが0に等しい必要があることを指定する。no_qtbt_dual_tree_intra_constraint_flagが0に等しい場合、このような制約は課されない。

no_partition_constraints_override_constraint_flagが1に等しい場合、partition_constraints_override_enabled_flagが0に等しい必要があることを指定する。no_partition_constraints_override_constraint_flagが0に等しい場合、このような制約は課されない。

no_sao_constraint_flagが1に等しい場合、sps_sao_enabled_flagが0に等しい必要があることを指定する。no_sao_constraint_flagが0に等しい場合、このような制約は課されない。

50

no_alf_constraint_flagが1に等しい場合、sps_alf_enabled_flagが0に等しい必要があることを指定する。no_alf_constraint_flagが0に等しい場合、このような制約は課されない。

no_ccalf_constraint_flagが1に等しい場合、sps_ccalf_enabled_flagが0に等しい必要があることを指定する。no_ccalf_constraint_flagが0に等しい場合、このような制約は課されない。

no_joint_cbcr_constraint_flagが1に等しい場合、sps_joint_cbcr_enabled_flagが0に等しい必要があることを指定する。no_joint_cbcr_constraint_flagが0に等しい場合、このような制約は課されない。

no_ref_wraparound_constraint_flagが1に等しい場合、sps_ref_wraparound_enabled_flagが0に等しい必要があることを指定する。no_ref_wraparound_constraint_flagが0に等しい場合、このような制約は課されない。

10

no_temporal_mvp_constraint_flagが1に等しい場合、sps_temporal_mvp_enabled_flagが0に等しい必要があることを指定する。no_temporal_mvp_constraint_flagが0に等しい場合、このような制約は課されない。

no_sbtmvp_constraint_flagが1に等しい場合、sps_sbtmvp_enabled_flagが0に等しい必要があることを指定する。no_sbtmvp_constraint_flagが0に等しい場合、このような制約は課されない。

no_amvr_constraint_flagが1に等しい場合、sps_amvr_enabled_flagが0に等しい必要があることを指定する。no_amvr_constraint_flagが0に等しい場合、このような制約は課されない。

20

no_bdof_constraint_flagが1に等しい場合、sps_bdof_enabled_flagが0に等しい必要があることを指定する。no_bdof_constraint_flagが0に等しい場合、このような制約は課されない。

no_dmvr_constraint_flagが1に等しい場合、sps_dmvr_enabled_flagが0に等しい必要があることを指定する。no_dmvr_constraint_flagが0に等しい場合、このような制約は課されない。

no_cclm_constraint_flagが1に等しい場合、sps_cclm_enabled_flagが0に等しい必要があることを指定する。no_cclm_constraint_flagが0に等しい場合、このような制約は課されない。

30

no_mts_constraint_flagが1に等しい場合、sps_mts_enabled_flagが0に等しい必要があることを指定する。no_mts_constraint_flagが0に等しい場合、このような制約は課されない。

no_sbt_constraint_flagが1に等しい場合、sps_sbt_enabled_flagが0に等しい必要があることを指定する。no_sbt_constraint_flagが0に等しい場合、このような制約は課されない。

no_affine_motion_constraint_flagが1に等しい場合、sps_affine_enabled_flagが0に等しい必要があることを指定する。no_affine_motion_constraint_flagが0に等しい場合、このような制約は課されない。

no_bcw_constraint_flagが1に等しい場合、sps_bcw_enabled_flagが0に等しい必要があることを指定する。no_bcw_constraint_flagが0に等しい場合、このような制約は課されない。

40

no_ibc_constraint_flagが1に等しい場合、sps_ibc_enabled_flagが0に等しい必要があることを指定する。no_ibc_constraint_flagが0に等しい場合、このような制約は課されない。

no_ciip_constraint_flagが1に等しい場合、sps_ciip_enabled_flagが0に等しい必要があることを指定する。no_cipp_constraint_flagが0に等しい場合、このような制約は課されない。

no_fpel_mmvd_constraint_flagが1に等しい場合、sps_fpel_mmvd_enabled_flagが0に等しい必要があることを指定する。no_fpel_mmvd_constraint_flagが0に等しい

50

い場合、このような制約は課されない。

no_gpm_constraint_flagが1に等しい場合、sps_gpm_enabled_flagが0に等しい必要があることを指定する。no_gpm_constraint_flagが0に等しい場合、このような制約は課されない。

no_ladf_constraint_flagが1に等しい場合、sps_ladf_enabled_flagが0に等しい必要があることを指定する。no_ladf_constraint_flagが0に等しい場合、このような制約は課されない。

no_transform_skip_constraint_flagが1に等しい場合、sps_transfrom_skip_enabled_flagが0に等しい必要があることを指定する。no_transform_skip_constraint_flagが0に等しい場合、このような制約は課されない。

10

no_bdpcm_constraint_flagが1に等しい場合、sps_bdpcm_enabled_flagが0に等しい必要があることを指定する。no_bdpcm_constraint_flagが0に等しい場合、このような制約は課されない。

no_qp_delta_constraint_flagが1に等しい場合、cu_qp_delta_enabled_flagが0に等しいことがビットストリーム準拠の要件であることを指定する。no_qp_delta_constraint_flagが0に等しい場合、このような制約は課されない。

no_dep_quant_constraint_flagが1に等しい場合、sps_dep_quant_enabled_flagが0に等しいことがビットストリーム準拠の要件であることを指定する。no_dep_quant_constraint_flagが0に等しい場合、このような制約は課されない。

no_sign_data_hiding_constraint_flagが1に等しい場合、sps_sign_data_hiding_enabled_flagが0に等しいことがビットストリーム準拠の要件であることを指定する。no_sign_data_hiding_constraint_flagが0に等しい場合、このような制約は課されない。

20

no_mixed_nalu_types_in_pic_constraint_flagが1に等しい場合、mixed_nalu_types_in_pic_flagが0に等しいことがビットストリーム準拠の要件であることを指定する。no_mixed_nalu_types_in_pic_constraint_flagが0に等しい場合、このような制約は課されない。

no_trail_constraint_flagが1に等しい場合、TRAIL_NUTと等しいnuh_unit_typeを有するNALユニットがOlsInScopeに存在しないことを指定する。no_trail_constraint_flagが0に等しい場合、このような制約は課されない。

30

no_stsa_constraint_flagが1に等しい場合、STSA_NUTと等しいnuh_unit_typeを有するNALユニットがOlsInScopeに存在しないことを指定する。no_stsa_constraint_flagが0に等しい場合、このような制約は課されない。

no_rasl_constraint_flagが1に等しい場合、RASL_NUTと等しいnuh_unit_typeを有するNALユニットがOlsInScopeに存在しないことを指定する。no_rasl_constraint_flagが0に等しい場合、このような制約は課されない。

no_radl_constraint_flagが1に等しい場合、RADL_NUTと等しいnuh_unit_typeを有するNALユニットがOlsInScopeに存在しないことを指定する。no_radl_constraint_flagが0に等しい場合、このような制約は課されない。

no_idr_constraint_flagが1に等しい場合、IDR_W_RADL又はIDR_N_LPと等しいnuh_unit_typeを有するNALユニットがOlsInScopeに存在しないことを指定する。no_idr_constraint_flagが0に等しい場合、このような制約は課されない。

40

no_cra_constraint_flagが1に等しい場合、CRA_NUTと等しいnuh_unit_typeを有するNALユニットがOlsInScopeに存在しないことを指定する。no_cra_constraint_flagが0に等しい場合、このような制約は課されない。

no_gdr_constraint_flagが1に等しい場合、GDR_NUTと等しいnuh_unit_typeを有するNALユニットがOlsInScopeに存在しないことを指定する。no_gdr_constraint_flagが0に等しい場合、このような制約は課されない。

no_aps_constraint_flagが1に等しい場合、PREFIX_APS_NUT又はSUFFIX_APS_NUTと等しいnuh_unit_typeを有するNALユニットがOlsInScopeに存在しないことを指定

50

する。no_aps_constraint_flagが0に等しい場合、このような制約は課されない。

gci_alignment_zero_bitsは0に等しい必要がある。

num_reserved_constraint_bytesは、予約された制約バイト数を指定する。num_reserved_constraint_bytesの値は0である必要がある。num_reserved_constraint_bytesの他の値は、ITU - T、ISO/IECにより将来の使用のために予約されており、このバージョンの仕様の準拠するビットストリームには存在しないものとする。

gci_reserved_constraint_byte[i]は任意の値を持つことができる。その存在及び値は、この仕様のこのバージョン内で指定されるプロファイルへのデコーダ準拠性に影響しない。このバージョンの仕様に準拠するデコーダは、全てのgci_reserved_constraint_byte[i]シンタックス要素を無視する必要がある。

【 0 0 3 4 】

3.2 SPSのシンタックスとセマンティクス

最新のVVCドラフトテキストでは、SPSシンタックスとセマンティクスは次の通りである。

10

20

30

40

50

【表 2】

	Descriptor	
seq_parameter_set_rbsp() {		
sps_seq_parameter_set_id	u(4)	
sps_video_parameter_set_id	u(4)	
sps_max_sublayers_minus1	u(3)	
sps_reserved_zero_4bits	u(4)	
sps_ptl_dpb_hrd_params_present_flag	u(1)	
if(sps_ptl_dpb_hrd_params_present_flag)		
profile_tier_level(1, sps_max_sublayers_minus1)		10
gdr_enabled_flag	u(1)	
chroma_format_idc	u(2)	
if(chroma_format_idc == 3)		
separate_colour_plane_flag	u(1)	
res_change_in_clvs_allowed_flag	u(1)	
pic_width_max_in_luma_samples	ue(v)	
pic_height_max_in_luma_samples	ue(v)	
sps_conformance_window_flag	u(1)	
if(sps_conformance_window_flag) {		20
sps_conf_win_left_offset	ue(v)	
sps_conf_win_right_offset	ue(v)	
sps_conf_win_top_offset	ue(v)	
sps_conf_win_bottom_offset	ue(v)	
}		
sps_log2_ctu_size_minus5	u(2)	
subpic_info_present_flag	u(1)	
if(subpic_info_present_flag) {		
sps_num_subpics_minus1	ue(v)	30
sps_independent_subpics_flag	u(1)	
for(i = 0; sps_num_subpics_minus1 > 0 && i <=		
sps_num_subpics_minus1; i++) {		
if(i > 0 && pic_width_max_in_luma_samples > CtbSizeY)		
subpic_ctu_top_left_x[i]	u(v)	
if(i > 0 && pic_height_max_in_luma_samples > CtbSizeY) {		
subpic_ctu_top_left_y[i]	u(v)	
if(i < sps_num_subpics_minus1 &&		
pic_width_max_in_luma_samples > CtbSizeY)		
subpic_width_minus1[i]	u(v)	40

50

if(i < sps_num_subpics_minus1 && pic_height_max_in_luma_samples > CtbSizeY)		
subpic_height_minus1[i]	u(v)	
if(!sps_independent_subpics_flag) {		
subpic_treated_as_pic_flag[i]	u(1)	
loop_filter_across_subpic_enabled_flag[i]	u(1)	
}		
}		10
sps_subpic_id_len_minus1	ue(v)	
subpic_id_mapping_explicitly_signalled_flag	u(1)	
if(subpic_id_mapping_explicitly_signalled_flag) {		
subpic_id_mapping_in_sps_flag	u(1)	
if(subpic_id_mapping_in_sps_flag)		
for(i = 0; i <= sps_num_subpics_minus1; i++)		
sps_subpic_id[i]	u(v)	
}		
}		20
bit_depth_minus8	ue(v)	
sps_entropy_coding_sync_enabled_flag	u(1)	
if(sps_entropy_coding_sync_enabled_flag)		
sps_wpp_entry_point_offsets_present_flag	u(1)	
sps_weighted_pred_flag	u(1)	
sps_weighted_bipred_flag	u(1)	
log2_max_pic_order_cnt_lsb_minus4	u(4)	
sps_poc_msb_flag	u(1)	
if(sps_poc_msb_flag)		
poc_msb_len_minus1	ue(v)	30
num_extra_ph_bits_bytes	u(2)	
extra_ph_bits_struct(num_extra_ph_bits_bytes)		
num_extra_sh_bits_bytes	u(2)	
extra_sh_bits_struct(num_extra_sh_bits_bytes)		
if(sps_max_sublayers_minus1 > 0)		
sps_sublayer_dpb_params_flag	u(1)	
if(sps_ptl_dpb_hrd_params_present_flag)		
dpb_parameters(sps_max_sublayers_minus1, sps_sublayer_dpb_params_flag)		40

10

20

30

40

50

long_term_ref_pics_flag	u(1)
inter_layer_ref_pics_present_flag	u(1)
sps_idr_rpl_present_flag	u(1)
rpl1_same_as_rpl0_flag	u(1)
for(i = 0; i < !rpl1_same_as_rpl0_flag ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j)	
}	
if(ChromaArrayType != 0)	
qtbtt_dual_tree_intra_flag	u(1)
log2_min_luma_coding_block_size_minus2	ue(v)
partition_constraints_override_enabled_flag	u(1)
sps_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_luma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)
}	
sps_log2_diff_min_qt_min_cb_inter_slice	ue(v)
sps_max_mtt_hierarchy_depth_inter_slice	ue(v)
if(sps_max_mtt_hierarchy_depth_inter_slice != 0) {	
sps_log2_diff_max_bt_min_qt_inter_slice	ue(v)
sps_log2_diff_max_tt_min_qt_inter_slice	ue(v)
}	
if(qtbtt_dual_tree_intra_flag) {	
sps_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)
sps_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)
if(sps_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {	
sps_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)
sps_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)
}	
}	
sps_max_luma_transform_size_64_flag	u(1)
if(ChromaArrayType != 0) {	
sps_joint_cbr_enabled_flag	u(1)

10

20

30

40

50

same_qp_table_for_chroma	u(1)	
numQpTables = same_qp_table_for_chroma ? 1 : (sps_joint_cbr_enabled_flag ? 3 : 2)		
for(i = 0; i < numQpTables; i++) {		
qp_table_start_minus26[i]	se(v)	
num_points_in_qp_table_minus1[i]	ue(v)	
for(j = 0; j <= num_points_in_qp_table_minus1[i]; j++) {		
delta_qp_in_val_minus1[i][j]	ue(v)	10
delta_qp_diff_val[i][j]	ue(v)	
}		
}		
}		
sps_sao_enabled_flag	u(1)	
sps_alf_enabled_flag	u(1)	
if(sps_alf_enabled_flag && ChromaArrayType != 0)		
sps_ccalf_enabled_flag	u(1)	
sps_transform_skip_enabled_flag	u(1)	20
if(sps_transform_skip_enabled_flag) {		
log2_transform_skip_max_size_minus2	ue(v)	
sps_bdpcm_enabled_flag	u(1)	
}		
sps_ref_wraparound_enabled_flag	u(1)	
sps_temporal_mvp_enabled_flag	u(1)	
if(sps_temporal_mvp_enabled_flag)		
sps_sbtmvp_enabled_flag	u(1)	
sps_amvr_enabled_flag	u(1)	
sps_bdof_enabled_flag	u(1)	30
if(sps_bdof_enabled_flag)		
sps_bdof_pic_present_flag	u(1)	
sps_smvd_enabled_flag	u(1)	
sps_dmvr_enabled_flag	u(1)	
if(sps_dmvr_enabled_flag)		
sps_dmvr_pic_present_flag	u(1)	
sps_mmvd_enabled_flag	u(1)	
sps_isp_enabled_flag	u(1)	
sps_mrl_enabled_flag	u(1)	40

sps_mip_enabled_flag	u(1)	
if(ChromaArrayType != 0)		
sps_cclm_enabled_flag	u(1)	
if(chroma_format_idc == 1) {		
sps_chroma_horizontal_collocated_flag	u(1)	
sps_chroma_vertical_collocated_flag	u(1)	
}		
sps_mts_enabled_flag	u(1)	10
if(sps_mts_enabled_flag) {		
sps_explicit_mts_intra_enabled_flag	u(1)	
sps_explicit_mts_inter_enabled_flag	u(1)	
}		
six_minus_max_num_merge_cand	ue(v)	
sps_sbt_enabled_flag	u(1)	
sps_affine_enabled_flag	u(1)	
if(sps_affine_enabled_flag) {		
five_minus_max_num_subblock_merge_cand	ue(v)	20
sps_affine_type_flag	u(1)	
if(sps_amvr_enabled_flag)		
sps_affine_amvr_enabled_flag	u(1)	
sps_affine_prof_enabled_flag	u(1)	
if(sps_affine_prof_enabled_flag)		
sps_prof_pic_present_flag	u(1)	
}		
sps_palette_enabled_flag	u(1)	
if(ChromaArrayType == 3 && !sps_max_luma_transform_size_64_flag)		30
sps_act_enabled_flag	u(1)	
if(sps_transform_skip_enabled_flag sps_palette_enabled_flag)		
min_qp_prime_ts_minus4	ue(v)	
sps_bcw_enabled_flag	u(1)	
sps_ibc_enabled_flag	u(1)	
if(sps_ibc_enabled_flag)		
six_minus_max_num_ibc_merge_cand	ue(v)	
sps_ciip_enabled_flag	u(1)	
if(sps_mmvd_enabled_flag)		40

sps_fpel_mmvd_enabled_flag	u(1)
if(MaxNumMergeCand >= 2) {	
sps_gpm_enabled_flag	u(1)
if(sps_gpm_enabled_flag && MaxNumMergeCand >= 3)	
max_num_merge_cand_minus_max_num_gpm_cand	ue(v)
}	
sps_lmcs_enabled_flag	u(1)
sps_lfnst_enabled_flag	u(1)
sps_ladf_enabled_flag	u(1)
if(sps_ladf_enabled_flag) {	
sps_num_ladf_intervals_minus2	u(2)
sps_ladf_lowest_interval_qp_offset	se(v)
for(i = 0; i < sps_num_ladf_intervals_minus2 + 1; i++) {	
sps_ladf_qp_offset[i]	se(v)
sps_ladf_delta_threshold_minus1[i]	ue(v)
}	
}	
log2_parallel_merge_level_minus2	ue(v)
sps_scaling_list_enabled_flag	u(1)
sps_dep_quant_enabled_flag	u(1)
if(!sps_dep_quant_enabled_flag)	
sps_sign_data_hiding_enabled_flag	u(1)
sps_virtual_boundaries_enabled_flag	u(1)
if(sps_virtual_boundaries_enabled_flag) {	
sps_virtual_boundaries_present_flag	u(1)
if(sps_virtual_boundaries_present_flag) {	
sps_num_ver_virtual_boundaries	u(2)
for(i = 0; i < sps_num_ver_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_x[i]	u(13)
sps_num_hor_virtual_boundaries	u(2)
for(i = 0; i < sps_num_hor_virtual_boundaries; i++)	
sps_virtual_boundaries_pos_y[i]	u(13)
}	
}	
if(sps_ptl_dpb_hrd_params_present_flag) {	
sps_general_hrd_params_present_flag	u(1)

10

20

30

40

50

if(sps_general_hrd_params_present_flag) {		
general_hrd_parameters()		
if(sps_max_sublayers_minus1 > 0)		
sps_sublayer_cpb_params_present_flag	u(1)	
firstSubLayer = sps_sublayer_cpb_params_present_flag ? 0 : sps_max_sublayers_minus1		
ols_hrd_parameters(firstSubLayer, sps_max_sublayers_minus1)		10
}		
}		
field_seq_flag	u(1)	
vui_parameters_present_flag	u(1)	
if(vui_parameters_present_flag)		
vui_parameters() /* Specified in ITU-T H.SEI ISO/IEC 23002-7 */		
sps_extension_flag	u(1)	20
if(sps_extension_flag)		
while(more_rbsp_data())		
sps_extension_data_flag	u(1)	
rbsp_trailing_bits()		
}		

SPS RBSPは、参照され、0に等しいTemporalIdを有する少なくとも1つのAUに含まれ、又は外部手段を通じて提供される前に、復号処理に利用可能である必要がある。

CVS内で特定の値のsps_seq_parameter_set_idを有する全部のSPS NALユニットは同じコンテンツを有する必要がある。

sps_seq_parameter_set_idは、他のシンタックス要素による参照のためにSPSの識別子を提供する。

SPS NALユニットは、nuh_layer_idの値に関係なく、sps_seq_parameter_set_idの同じ値空間を共有する。

spsLayerIdを特定のSPS NALユニットのnuh_layer_idの値とし、vclLayerIdを特定のVCL NALユニットのnuh_layer_idの値とする。spsLayerIdがvclLayerId以下で、nuh_layer_idがspsLayerIdと等しいレイヤが、nuh_layer_idがvclLayerIdと等しいレイヤを含む少なくとも1つのOLSに含まれていない限り、特定のVCL NALユニットは特定のSPS NALユニットを参照しないものとする。

sps_video_parameter_set_idは、0より大きい場合、SPSによって参照されるVPSのvps_video_parameter_set_idの値を指定する。

sps_video_parameter_set_idが0に等しいとき、以下が適用される：

- SPSはVPSを参照しない、
- SPSを参照する各CLVSを復号する場合、VPSは参照されない、
- vps_max_layers_minus1の値は0に等しいと推定される、
- CVSは1層のみを含むものとする（つまり、CVS内のすべてのVCL NALユニットは、同じ値nuh_layer_idを持つものとする）、

30

40

50

- GeneralLayerIdx[nuh_layer_id]の値は0に等しいと推定される、
 - vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]の値は1に等しいと推定される、

vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]が1に等しい場合、特定のnuh_layer_id値であるnuhLayerIdを持つCLVSによって参照されるSPSは、nuhLayerIdに等しいnuh_layer_idを持つものとする。

sps_video_parameter_set_idの値は、CVS内のCLVSにより参照される全部のSPSの中で同じであるとする。

sps_max_sublayers_minus_1に1を加えたものは、SPSを参照する各CLVS内に存在し得る時間的サブレイヤの最大数を指定する。sps_max_sublayers_minus_1の値は、両端を含む0 ~ vps_max_sublayers_minus_1の範囲に包含されるものとする。

10

sps_reserved_zero_4 bitsは、このバージョンの仕様に準拠するビットストリームでは0に等しい必要がある。sps_reserved_zero_4 bitsの他の値は、ITU - T、ISO/IECによる将来の使用のために予約されている。

sps_ptl_dpb_hrd_params_present_flagが1に等しい場合は、profile_tier_level () シンタックス構造とdpb_parameters () シンタックス構造がSPSに存在し、general_hrd_parameters () シンタックス構造とols_hrd_parameters () シンタックス構造がSPSに存在し得ることを指定する。sps_ptl_dpb_hrd_params_present_flagが0に等しいことは、これらの4つのシンタックス構造がいずれもSPS内に存在しないことを指定する。sps_ptl_dpb_hrd_params_present_flagの値は、vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]と等しいものとする。

20

gdr_enabled_flagが1に等しい場合は、GDRピクチャがSPSを参照するCLVSに存在する可能性があることを指定する。gdr_enabled_flagが0に等しい場合は、GDRピクチャがSPSを参照するCLVSに存在しないことを指定する。

chroma_format_idcは、6.2項で指定されているルマサンプリングに対するクロマサンプリングを指定する。

separate_colour_plane_flagが1に等しい場合は、4:4:4クロマフォーマットの3つの色成分が別々にコーディングされることを指定する。separate_colour_plane_flagが0に等しい場合は、色成分を個別にコーディングしないことを指定する。separate_colour_plane_flagが存在しない場合、それは0に等しいと推定される。separate_colour_plane_flagが1に等しい場合、コーディングされたピクチャは3つの個別の成分で構成され、各々が1つの色平面(Y、Cb、又はCr)のコーディングされたサンプルで構成され、モノクロコーディングシンタックスを使用する。この場合、各色平面は特定のcolour_plane_id値に関連付けられている。

30

注1：異なるcolour_plane_id値を持つ色平面の間の復号処理に依存性はない。例えば、colour_plane_idの1つの値を有するモノクロピクチャの復号処理は、colour_plane_idの値が異なるモノクロピクチャからのデータをインター予測に使用しない。

separate_colour_plane_flagの値に応じて、変数ChromaArrayTypeの値は次のように割り当てられる。

- separate_colour_plane_flagが0に等しい場合、ChromaArrayTypeはchroma_format_idcに等しく設定される。

40

- その他の場合 (separate_colour_plane_flagが1に等しい)、ChromaArrayTypeは0に等しく設定される。

res_change_in_clvs_allowed_flagが1に等しい場合は、ピクチャ空間解像度は、SPSを参照するCLVSの範囲内で変化してよいことを指定する。res_change_in_clvs_allowed_flagが0に等しい場合は、ピクチャ空間解像度は、SPSを参照するCLVSの範囲内で変化しないことを指定する。

pic_width_max_in_luma_samplesは、SPSを参照する各復号ピクチャの最大幅を、ルマサンプルの単位で指定する。pic_width_max_in_luma_samplesは0に等しくなく、及びMax(8, MinCbSizeY)の整数倍であるものとする。

50

SPSを参照する1つ以上のレイヤを含むOLSインデックス*i*を持つ任意のOLSについて、pic_width_max_in_luma_samplesの値がols_dpb_pic_width[*i*]の値以下であることがビットストリーム準拠の要件である。

pic_height_max_in_luma_samplesは、SPSを参照する各復号ピクチャの最大高さを、ルマサンプルの単位で指定する。pic_height_max_in_luma_samplesは0に等しくなく、及びMax(8, MinCbSizeY)の整数倍であるものとする。

SPSを参照する1つ以上のレイヤを含むOLSインデックス*i*を持つ任意のOLSについて、pic_width_max_in_luma_samplesの値がols_dpb_pic_width[*i*]の値以下であることがビットストリーム準拠の要件である。

sps_conformance_window_flagが1に等しい場合は、適合クロッピングウィンドウオフセットパラメータがSPSの次に続くことを示す。sps_conformance_window_flagが0に等しいことは、適合クロッピングウィンドウオフセットパラメータがSPS内に存在しないことを示す。

sps_conf_win_left_offset、sps_conf_win_right_offset、sps_conf_win_top_offset、及びsps_conf_win_bottom_offsetは、pic_width_in_luma_samplesがpic_width_max_in_luma_samplesに等しい、及びpic_height_in_luma_samplesがpic_height_max_in_luma_samplesと等しいピクチャに適用されるクロップウィンドウを指定する。sps_conformance_window_flagが0の場合、sps_conf_win_left_offset、sps_conf_win_right_offset、sps_conf_win_top_offset、sps_conf_win_bottom_offsetの値は0に等しいと推定される。

適合クロッピングウィンドウは、両端を含むSubWidthC*sps_conf_win_left_offsetからpic_width_max_in_luma_samples-(SubWidthC*sps_conf_win_right_offset+1)までの水平ピクチャ座標、及び両端を含むSubHeightC*sps_conf_win_top_offsetからpic_height_max_in_luma_samples-(SubHeightC*sps_conf_win_bottom_offset+1)までの垂直ピクチャ座標を有するルマサンプルを含む。

SubWidthC*(sps_conf_win_left_offset+sps_conf_win_right_offset)の値はpic_width_max_in_luma_samplesより小さいものとし、SubHeightC*(sps_conf_win_top_offset+sps_conf_win_bottom_offset)の値はpic_height_max_in_luma_samplesより小さいものとする。

ChromaArrayTypeが0に等しくないとき、2つのクロマアレイの対応する指定されたサンプルは、ピクチャ座標(x/SubWidthC,y/SubHeightC)を有するサンプルである。ここで、(x,y)は指定されたルマサンプルのピクチャ座標である。

注2：適合クロッピングウィンドウオフセットパラメータは出力においてのみ適用される。すべての内部復号処理は、トリミングされていないピクチャサイズに適用される。

sps_log2_ctu_size_minus5に5を加えたものは、各CTUのルマコーディングツリーブロックサイズを指定する。sps_log2_ctu_size_minus5の値は、両端を含む0~2の範囲であるものとする。sps_log2_ctu_size_minus5の値3は、ITU-T、ISO/IECによる将来の使用のために予約されている。

変数CtbLog2SizeY及びCtbSizeYは、以下のように導出される：

【数1】

$$\text{CtbLog2SizeY} = \text{sps_log2_ctu_size_minus5} + 5 \quad (43)$$

$$\text{CtbSizeY} = 1 \ll \text{CtbLog2SizeY} \quad (44)$$

subpic_info_present_flagが1に等しい場合、CLVSのサブピクチャ情報が存在し、CLVSの各ピクチャに1つ以上のサブピクチャが存在する可能性があることを指定する。subpic_info_present_flagが0に等しい場合、CLVSのサブピクチャ情報が存在せず、CLVSの各ピクチャに1つのみのサブピクチャが存在することを指定する。

res_change_in_clvs_allowed_flagが1に等しい場合、subpic_info_present_flag

10

20

30

40

50

の値は 0 に等しい必要がある。

注 3：ビットストリームがサブビットストリーム抽出処理の結果であり、サブビットストリーム抽出処理への入力ビットストリームのサブピクチャの一部のみを含むとき、SPS のRBSP内のsubpic_info_present_flagの値を 1 に等しく設定することが要求され得る。

sps_num_subpics_minus 1 に 1 を加えたものは、CLVS内の各ピクチャ内のサブピクチャの数を指定する。sps_num_subpics_minus 1 の値は、両端を含む 0 から $\text{Ceil}(\text{pic_width_max_in_luma_samples} \div \text{CtbSizeY}) * \text{Ceil}(\text{pic_height_max_in_luma_samples} \div \text{CtbSizeY}) - 1$ の範囲であるものとする。存在しないとき、sps_num_subpics_minus 1 の値は 0 に等しいと推定される。

sps_independent_subpics_flagが 1 に等しい場合、CLVS内のサブピクチャ境界に跨り、イントラ予測、インター予測、及びインループフィルタリング操作が実行できないことを指定する。sps_independent_subpics_flagが 0 に等しい場合、CLVS内のサブピクチャ境界に跨り、インター予測又はインループフィルタリング操作が許可されることを指定する。存在しないとき、sps_independent_subpics_flagの値は 0 に等しいと推定される。

subpic_ctu_top_left_x[i]は、CtbSizeYのユニットのi番目のサブピクチャの左上CTUの水平位置を指定する。シンタックス要素の長さは、 $\text{Ceil}(\text{Log}_2((\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) \text{ CtbLog}_2 \text{ SizeY}))$ ビットである。存在しないとき、subpic_ctu_top_left_x[i]の値は 0 に等しいと推定される。

subpic_ctu_top_left_y[i]は、CtbSizeYのユニットのi番目のサブピクチャの左上CTUの垂直位置を指定する。シンタックス要素の長さは、 $\text{Ceil}(\text{Log}_2((\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) \text{ CtbLog}_2 \text{ SizeY}))$ ビットである。存在しないとき、subpic_ctu_top_left_y[i]の値は 0 に等しいと推定される。

subpic_width_minus 1 [i]に 1 を加えたものは、CtbSizeYの単位でi番目のサブピクチャの幅を指定する。シンタックス要素の長さは、 $\text{Ceil}(\text{Log}_2((\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) \text{ CtbLog}_2 \text{ SizeY}))$ ビットである。存在しないとき、subpic_width_minus 1 [i]の値は、 $(\text{pic_width_max_in_luma_samples} + \text{CtbSizeY} - 1) \text{ CtbLog}_2 \text{ SizeY} - \text{subpic_ctu_top_left_x}[i] - 1$ に等しいと推定される。

subpic_height_minus 1 [i]に 1 を加えたものは、CtbSizeYの単位でi番目のサブピクチャの高さを指定する。シンタックス要素の長さは、 $\text{Ceil}(\text{Log}_2((\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) \text{ CtbLog}_2 \text{ SizeY}))$ ビットである。存在しないとき、subpic_height_minus 1 [i]の値は、 $(\text{pic_height_max_in_luma_samples} + \text{CtbSizeY} - 1) \text{ CtbLog}_2 \text{ SizeY} - \text{subpic_ctu_top_left_y}[i] - 1$ に等しいと推定される。

subpic_treated_as_pic_flag[i]が 1 に等しい場合、CLVS内の各コーディングピクチャのi番目のサブピクチャが、インループフィルタリング操作を除き復号処理でピクチャとして取り扱われることを指定する。subpic_treated_as_pic_flag[i]が 0 に等しいことは、CLVS内の各コーディングピクチャのi番目のサブピクチャが、インループフィルタリング動作を除く復号処理でピクチャとして扱われないことを指定する。存在しないとき、subpic_treated_as_pic_flag[i]の値はsps_independent_subpics_flagに等しいと推定される。

subpic_treated_as_pic_flag[i]が 1 に等しい場合、i番目のサブピクチャを含むレイヤを出力レイヤとして含むOLSで、各出力レイヤとその参照レイヤについて次のすべての条件が当てはまるのがビットストリーム準拠の要件である。

- 出力レイヤとその参照レイヤのすべてのピクチャは、同じ値のpic_width_in_luma_samplesと同じ値のpic_height_in_luma_samplesを持つものとする、

- 出力レイヤとその参照レイヤによって参照されるすべてのSPSは、同じ値のsubpic_ctu_top_left_x[j]、subpic_ctu_top_left_y[j]、subpic_width_minus 1 [j]、subpic_height_minus 1 [j]、及びloop_filter_across_subpic_enabled_flag[j]を持つものとし、各々のjの値は、両端を含む 0 からsps_num_subpics_minus 1 までの範囲にある、

- 出力レイヤとその参照レイヤの各アクセスユニット内のすべてのピクチャは、同じ値のSubpicIdVal[j]を持つものとし、各々のjの値は、両端を含む0からsps_num_subpics_minus 1までの範囲である。

loop_filter_across_subpic_enabled_flag[i]が1に等しい場合、インループフィルタリング操作が、CLVSにおける各コーディングピクチャの中のi番目のサブピクチャの境界に跨がり実行されてよいことを指定する。loop_filter_across_subpic_enabled_flag[i]が0に等しいことは、CLVS内の各コーディングピクチャ内のi番目のサブピクチャの境界に跨がり、インループフィルタリング動作が実行されないことを指定する。存在しないとき、vloop_filter_across_subpic_enabled_pic_flag[i]の値は1-sps_independent_subpics_flagに等しいと推定される。

10

各サブピクチャが、復号される時、ピクチャ境界を構成する又は前に復号されたサブピクチャの境界を構成する、その左境界全体及び上境界全体を有するようなサブピクチャの形状が、ビットストリーム準拠の要件である。

sps_subpic_id_len_minus 1に1を加えたものは、シンタックス要素sps_subpic_id[i]、存在する場合はシンタックス要素pps_subpic_id[i]、及び存在する場合はシンタックス要素slice_subpic_idを表すために使用されるビット数を指定する。sps_subpic_id_len_minus 1の値は、両端を含む0 ~ 15の範囲であるものとする。1 (sps_subpic_id_len_minus 1 + 1)の値はsps_num_subpics_minus 1 + 1以上であるものとする。

subpic_id_mapping_explicitly_signalled_flagが1に等しい場合は、サブピクチャIDマッピングが、SPS内又はCLVSのコーディングされたピクチャによって参照されるPPS内のいずれかで明示的にシグナリングされることを指定する。subpic_id_mapping_explicitly_signalled_flagが0に等しい場合、サブピクチャIDマッピングがCLVSについて明示的にシグナリングされないことを指定する。存在しないとき、subpic_id_mapping_explicitly_signalled_flagの値は0に等しいと推定される。

20

subpic_id_mapping_in_sps_flagが1に等しい場合は、subpic_id_mapping_explicitly_signalled_flagが1に等しいときに、サブピクチャIDマッピングがSPS内でシグナリングされることを指定する。subpic_id_mapping_in_sps_flagが0に等しい場合は、subpic_id_mapping_explicitly_signalled_flagが1に等しいときに、サブピクチャIDマッピングがCLVSのコーディングピクチャにより参照されるPPS内でシグナリングされることを指定する。

30

sps_subpic_id[i]は、i番目のサブピクチャのサブピクチャIDを指定する。sps_subpic_id[i]シンタックス要素の長さは、sps_subpic_id_len_minus 1 + 1ビットである。

bit_depth_minus 8は、ルマアレイとクロマアレイのサンプルのビット深度BitDepth、及びルマ及びクロマ量子化パラメータ範囲オフセットの値QpBdOffsetを次のように指定する：

【数2】

$$\text{BitDepth} = 8 + \text{bit_depth_minus8} \quad (45)$$

$$\text{QpBdOffset} = 6 * \text{bit_depth_minus8} \quad (46)$$

40

bit_depth_minus 8は、両端を含む0 ~ 8の範囲内であるものとする。

sps_entropy_coding_sync_enabled_flagが1に等しい場合は、SPSを参照する各ピクチャの各タイルのCTB行の第1CTBを含むCTUを復号する前に、コンテキスト変数の特定の同期処理が呼び出され、SPSを参照する各ピクチャの各タイルのCTB行の第1CTBを含むCTUを復号した後に、コンテキスト変数の特定の記憶処理が呼び出されることを指定する。sps_entropy_coding_sync_enabled_flagが0に等しい場合は、SPSを参照する各ピクチャの各タイルのCTB行の第1CTBを含むCTUを復号する前に、コンテキスト変数

50

の特定の同期処理が呼び出される必要がなく、SPSを参照する各ピクチャの各タイルのCTB行の第1CTBを含むCTUを復号した後に、コンテキスト変数の特定の記憶処理が呼び出される必要がないことを指定する。

sps_wpp_entry_point_offsets_present_flagが1に等しい場合、sps_entropy_coding_sync_enabled_flagが1に等しいとき、CTU行のエントリポイントオフセットのシグナリングがSPSを参照するピクチャのスライスヘッダ内に存在する可能性があることを指定する。sps_wpp_entry_point_offsets_present_flagが0に等しい場合、CTU行のエントリポイントオフセットのシグナリングがSPSを参照するピクチャのスライスヘッダ内に存在しないことを指定する。存在しないとき、sps_wpp_entry_point_offsets_present_flagの値は0に等しいと推定される。

10

sps_weighted_pred_flagが1に等しい場合は、加重予測がSPSを参照するPスライスに適用されてよいことを指定する。sps_weighted_pred_flagが0に等しい場合は、加重予測がSPSを参照するPスライスに適用されないことを指定する。

sps_weighted_bipred_flagが1に等しい場合は、明示的加重予測がSPSを参照するBスライスに適用されてよいことを指定する。sps_weighted_bipred_flagが0に等しい場合は、明示的加重予測がSPSを参照するBスライスに適用されないことを指定する。

log2_max_pic_order_cnt_lsb_minus4は、ピクチャオーダカウンタの復号処理で使用される変数MaxPicOrderCntLsbの値を次のように指定する：

【数3】

$$\text{MaxPicOrderCntLsb} = 2^{(\text{log2_max_pic_order_cnt_lsb_minus4} + 4)} \quad (47)$$

20

log2_max_pic_order_cnt_lsb_minus4の値は、両端を含む0～12の範囲であるものとする。

sps_poc_msb_flagが1に等しい場合は、ph_poc_msb_present_flagシンタックス要素がSPSを参照するPH内に存在することを指定する。sps_poc_msb_flagが0に等しい場合は、ph_poc_msb_present_flagシンタックス要素がSPSを参照するPH内に存在しないことを指定する。

poc_msb_len_minus1に1を加えたものは、SPSを参照するPH内に存在する場合、poc_msb_valsyntaxelementsの長さをビット単位で指定する。poc_msb_len_minus1の値は、両端を含む0～32-log2_max_pic_order_cnt_lsb_minus4-5の範囲であるものとする。

30

num_extra_ph_bits_bytesは、SPSを参照するコーディングされたピクチャのPHシンタックス構造の追加ビットのバイト数を指定する。num_extra_ph_bits_bytesの値は、このバージョンの仕様に準拠するビットストリームでは0に等しく設定されるものとする。このバージョンの仕様では、num_extra_ph_bits_bytesの値は0に等しい必要があるが、このバージョンの仕様に準拠したデコーダは、シンタックス内に1又は2に等しいnum_extra_ph_bits_bytesの値の出現を許容するものとする。

num_extra_sh_bits_bytesは、SPSを参照するコーディングされたピクチャのスライスヘッダ内の追加ビットのバイト数を指定する。num_extra_sh_bits_bytesの値は、このバージョンの仕様に準拠するビットストリームでは0に等しいものとする。このバージョンの仕様では、num_extra_sh_bits_bytesの値は0に等しい必要があるが、このバージョンの仕様に準拠したデコーダは、シンタックス内に1又は2に等しいnum_extra_sh_bits_bytesの値の出現を許容するものとする。

40

sps_sublayer_dpb_params_flagは、SPS内のdpb_parameters()シンタックス構造内のmax_dec_pic_buffering_minus1[i]、max_num_reorder_pics[i]、及びmax_latency_increase_plus1[i]シンタックス要素の存在を制御するために使用される。存在しないとき、sps_sub_dpb_params_info_present_flagの値は0に等しいと推定される。

50

long_term_ref_pics_flagが0に等しい場合は、CLVS内のコーディングされたピクチャのインター予測にLTRPが使用されないことを指定する。long_term_ref_pics_flagが1に等しい場合は、CLVS内の1つ以上のコーディングされたピクチャのインター予測にLTRPが使用されてよいことを指定する。

inter_layer_ref_pics_present_flagが0に等しい場合は、CLVS内のコーディングされたピクチャのインター予測にILRPが使用されないことを指定する。inter_layer_ref_pic_flagが1に等しい場合は、CLVS内の1つ以上のコーディングされたピクチャのインター予測にLTRPが使用されてよいことを指定する。sps_video_parameter_set_idが0に等しい場合、inter_layer_ref_pics_present_flagの値は0に等しいと推定される。vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]が1に等しい場合、inter_layer_ref_pics_present_flagの値は0に等しいものとする。[Ed.(YK):このシンタックス要素により良い名前がないか確認する。]

10

sps_idr_rpl_present_flagが1に等しい場合、IDRピクチャのスライスヘッダ内に参照ピクチャリストシンタックス要素が存在することを指定する。sps_idr_rpl_present_flagが0に等しい場合、IDRピクチャのスライスヘッダ内に参照ピクチャリストシンタックス要素が存在しないことを指定する。

rpl1_same_as_rpl0_flagが1に等しい場合、シンタックス要素num_ref_pic_lists_in_sps[1]及びシンタックス構造ref_pic_list_struct(1,rplIdx)が存在しないことを指定し、以下を適用する：

- num_ref_pic_lists_in_sps[1]の値はnum_ref_pic_lists_in_sps[0]の値に等しいと推定される、

20

ref_pic_list_struct(1,rplIdx)の各シンタックス要素の値は、0からnum_ref_pic_lists_in_sps[0]-1の範囲のrplIdxについて、ref_pic_list_struct(0,rplIdx)内の対応するシンタックス要素の値と等しいと推定される。

num_ref_pic_lists_in_sps[i]は、SPSに含まれるiに等しいlistIdxを有するref_pic_list_struct(listIdx,rplIdx)シンタックス構造の数を指定する。num_ref_pic_lists_in_sps[i]の値は、両端を含む0~64の範囲であるものとする。

注4：現在のピクチャのスライスヘッダ内で直接シグナリングされる1つのref_pic_list_struct(listIdx,rplIdx)シンタックス構造が存在する可能性があるため、listIdxの値(0又は1に等しい)ごとに、デコーダは合計でnum_ref_pic_lists_in_sps[i]+1個のref_pic_list_struct(listIdx,rplIdx)シンタックス構造にメモリを割り当てる必要がある。

30

qtbtt_dual_tree_intra_flagが1に等しい場合、1スライスについて、各CTUが暗黙的4分木分割を用いて64x64ルマサンプルを有するコーディングユニットに分割されること、及びこれらのコーディングユニットがルマ及びクロマの2個の別個のcoding_treeシンタックス構造のルートであること、を指定する。qtbtt_dual_tree_intra_flagが0に等しい場合、1スライスについて、個別のcoding_treeシンタックス構造が使用されないことを指定する。qtbtt_dual_tree_intra_flagが存在しない場合、それは0に等しいと推定される。

log2_min_luma_coding_block_size_minus_2に2を加えたものは、最小ルマコーディングブロックサイズを指定する。log2_min_luma_coding_block_size_minus_2の値範囲は、両端を含む0~Min(4,sps_log2_ctu_size_minus_5+3)の範囲であるものとする。

40

変数MinCbLog2SizeY、MinCbSizeY、IbcBufWidthY、IbcBufWidthC、及びVsiz eは、以下のように導出される：

【数4】

$$\text{MinCbLog2SizeY} = \log2_min_luma_coding_block_size_minus2 + 2 \quad (48)$$

$$\text{MinCbSizeY} = 1 \ll \text{MinCbLog2SizeY} \quad (49)$$

$$\text{IbcBufWidthY} = 256 * 128 / \text{CtbSizeY} \quad (50)$$

$$\text{IbcBufWidthC} = \text{IbcBufWidthY} / \text{SubWidthC} \quad (51)$$

$$\text{VSize} = \text{Min}(64, \text{CtbSizeY}) \quad (52)$$

10

MinCbSizeYの値は、VSize以下であるものとする。

各クロマCTBのアレイの幅と高さを各々指定する変数CtbWidthC及びCtbHeightCは、次のように導出される：

- chroma_format_idcが0（モノクロ）に等しいか、又はseparate_colour_plane_flagが1に等しい場合、CtbWidthC及びCtbHeightCは両方とも0に等しい。

- その他の場合、CtbWidthC及びCtbHeightCは、以下のように導出される：

【数5】

$$\text{CtbWidthC} = \text{CtbSizeY} / \text{SubWidthC} \quad (53)$$

$$\text{CtbHeightC} = \text{CtbSizeY} / \text{SubHeightC} \quad (54)$$

20

両端を含む、0から4までのlog2 BlockWidth、及び0から4までのlog2 BlockHeightでは、6 . 5 . 2項で指定されている右上の斜め方向のスキャン順序アレイ初期化プロセスが、1 log2 BlockWidth及び1 log2 BlockHeightを入力として呼び出され、出力がDiagScanOrder[log2 BlockWidth][log2 BlockHeight]に割り当てられる。

両端を含む、0から6までのlog2 BlockWidth、及び0から6までのlog2 BlockHeightでは、6 . 5 . 3項で指定されている水平及び垂直トラバーススキャン順序アレイ初期化プロセスが、1 log2 BlockWidth及び1 log2 BlockHeightを入力として呼び出され、出力がHorTravScanOrder[log2 BlockWidth][log2 BlockHeight]及びVerTravScanOrder[log2 BlockWidth][log2 BlockHeight]に割り当てられる。

30

partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHにpartition_constraints_override_flagが存在することを指定する。partition_constraints_override_enabled_flagが0に等しい場合、SPSを参照するPHにpartition_constraints_override_flagが存在しないことを指定する。

sps_log2_diff_min_qt_min_cb_intra_slice_lumaは、CTUの4分木分割により生じたルマリーフブロックのルマサンプルのうちの最小サイズの底2の対数と、slice_typeがSPSを参照する2(1)に等しいスライスブロックのルマサンプルのうちの最小コーディングブロックサイズの底2の対数との間のデフォルトの差を指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_log2_diff_min_qt_min_cb_intra_slice_lumaによってデフォルトの差を上書きできる。sps_log2_diff_min_qt_min_cb_intra_slice_lumaの値は、0 ~ CtbLog2SizeY - MinCbLog2SizeYの範囲であるものとする。CTUの4分木分割から生じるルマリーフブロックのルマサンプルにおける最小サイズの底2の対数は、次のように導出される：

40

【数6】

50

$$\text{MinQtLog2SizeIntraY} = \text{sps_log2_diff_min_qt_min_cb_intra_slice_luma} + \text{MinCbLog2SizeY} \quad (55)$$

sps_max_mtt_hierarchy_depth_intra_slice_lumaは、SPSを参照する2(I)に等しいslice_typeを有するスライスの中の4分木リーフのマルチタイプ木分割から生じるコーディングユニットのデフォルトの最大階層構造深さを指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_max_mtt_hierarchy_depth_intra_slice_lumaによってデフォルトの最大階層構造深さを上書きできる。sps_max_mtt_hierarchy_depth_intra_slice_lumaの値は、両端を含む0 ~ 2 * (CtbLog2SizeY - MinCbLog2SizeY)の範囲であるものとする。

10

sps_log2_diff_max_bt_min_qt_intra_slice_lumaは、2分割を使用して分割できるルマコーディングブロックのルマサンプルのうち最大サイズ(幅又は高さ)の底2の対数と、slice_typeがSPSを参照する2(I)に等しいスライス内のCTUの4分木分割によって生じるルマリーフブロックのルマサンプルのうちの最小サイズ(幅又は高さ)との間のデフォルトの差を指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_log2_diff_max_bt_min_qt_lumaによってデフォルトの差を上書きできる。sps_log2_diff_max_bt_min_qt_intra_slice_lumaの値は、0 ~ CtbLog2SizeY - MinQtLog2SizeIntraYの範囲であるものとする。sps_log2_diff_max_bt_min_qt_intra_slice_lumaが存在しないとき、sps_log2_diff_max_bt_min_qt_intra_slice_lumaの値は0に等しいと推定される。

20

sps_log2_diff_max_tt_min_qt_intra_slice_lumaは、3分割を使用して分割できるルマコーディングブロックのルマサンプルのうち最大サイズ(幅又は高さ)の底2の対数と、slice_typeがSPSを参照する2(I)に等しいスライス内のCTUの4分木分割によって生じるルマリーフブロックのルマサンプルのうちの最小サイズ(幅又は高さ)との間のデフォルトの差を指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_log2_diff_max_tt_min_qt_lumaによってデフォルトの差を上書きできる。sps_log2_diff_max_tt_min_qt_intra_slice_lumaの値は、0 ~ CtbLog2SizeY - MinQtLog2SizeIntraYの範囲に包含されるものとする。sps_log2_diff_max_tt_min_qt_intra_slice_lumaが存在しないとき、sps_log2_diff_max_tt_min_qt_intra_slice_lumaの値は0に等しいと推定される。

30

sps_log2_diff_min_qt_min_cb_inter_sliceは、CTUの4分木分割により生じたルマリーフブロックのルマサンプルのうちの最小サイズの底2の対数と、SPSを参照するslice_typeが0(B)又は1(P)に等しいスライスのルマCUのルマサンプルのうちの最小ルマコーディングブロックサイズの底2の対数との間のデフォルトの差を指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_log2_diff_min_qt_min_cb_lumaによってデフォルトの差を上書きできる。sps_log2_diff_min_qt_min_cb_inter_sliceの値は、0 ~ CtbLog2SizeY - MinCbLog2SizeYの範囲に包含されるものとする。CTUの4分木分割から生じるルマリーフブロックのルマサンプルにおける最小サイズの底2の対数は、次のように導出される：

40

【数7】

$$\text{MinQtLog2SizeInterY} = \text{sps_log2_diff_min_qt_min_cb_inter_slice} + \text{MinCbLog2SizeY} \quad (56)$$

sps_max_mtt_hierarchy_depth_inter_sliceは、SPSを参照するslice_typeが0(B)又は1(P)に等しいスライス内の4分木リーフのマルチタイプツリー分割から生じるコーディングユニットのデフォルトの最大階層構造深さを指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_max_mtt_hierarchy_depth_inter_sliceによってデフォルトの最大階層構造深さを上書きできる。sps_max_mtt_hierarchy_depth_inter_sliceの値は、両端を含む0 ~ 2 * (CtbL

50

og 2 SizeY-MinCbLog 2 SizeY) の範囲であるものとする。

sps_log 2_diff_max_bt_min_qt_inter_sliceは、2分割を使用して分割できるルマコーディングブロックのルマサンプルのうち最大サイズ(幅又は高さ)の底2の対数と、SPSを参照するslice_typeが0(B)又は1(P)に等しいスライス内のCTUの4分木分割によって生じるルマリーフブロックのルマサンプルのうちの最小サイズ(幅又は高さ)との間のデフォルトの差を指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_log 2_diff_max_bt_min_qt_lumaによってデフォルトの差を上書きできる。sps_log 2_diff_max_bt_min_qt_inter_sliceの値は、0 ~ CtbLog 2 SizeY-MinQtLog 2 SizeInterYの範囲であるものとする。sps_log 2_diff_max_bt_min_qt_inter_sliceが存在しないとき、sps_log 2_diff_max_bt_min_qt_inter_sliceの値は0に等しいと推定される。

10

sps_log 2_diff_max_tt_min_qt_inter_sliceは、3分割を使用して分割できるルマコーディングブロックのルマサンプルのうち最大サイズ(幅又は高さ)の底2の対数と、SPSを参照するslice_typeが0(B)又は1(P)に等しいスライス内のCTUの4分木分割によって生じるルマリーフブロックのルマサンプルのうちの最小サイズ(幅又は高さ)との間のデフォルトの差を指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_log 2_diff_max_tt_min_qt_lumaによってデフォルトの差を上書きできる。sps_log 2_diff_max_tt_min_qt_inter_sliceの値は、0 ~ CtbLog 2 SizeY-MinQtLog 2 SizeInterYの範囲であるものとする。sps_log 2_diff_max_tt_min_qt_inter_sliceが存在しないとき、sps_log 2_diff_max_tt_min_qt_inter_sliceの値は0に等しいと推定される。

20

sps_log 2_diff_min_qt_min_cb_intra_slice_chromaは、treeTypeがDUAL_TREE_CHROMAに等しいクロマCTUの4分木分割により生じたクロマリーフブロックのクロマサンプルのうちの最小サイズの底2の対数と、slice_typeがSPSを参照する2(I)に等しいスライスのうち、treeTypeがDUAL_TREE_CHROMAに等しいクロマCUのルマサンプルのうちの最小コーディングブロックサイズの底2の対数との間のデフォルトの差を指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_log 2_diff_min_qt_min_cb_chromaによってデフォルトの差を上書きできる。sps_log 2_diff_min_qt_min_cb_intra_slice_chromaの値は、0 ~ CtbLog 2 SizeY-MinCbLog 2 SizeYの範囲に包含であるものとする。存在しないとき、sps_log 2_diff_min_qt_min_cb_intra_slice_chromaの値は0に等しいと推定される。treeTypeがDUAL_TREE_CHROMAに等しいCTUの4分木分割から生じるクロマリーフブロックのルマサンプルにおける最小サイズの底2の対数は、次のように導出される：

30

【数 8】

$$\text{MinQtLog2SizeIntraC} = \text{sps_log2_diff_min_qt_min_cb_intra_slice_chroma} + \text{MinCbLog2SizeY} \quad (57)$$

sps_max_mtt_hierarchy_depth_intra_slice_chromaは、SPSを参照する2(I)に等しいslice_typeを有するスライスの中のtreeTypeがDUAL_TREE_CHROMAに等しいクロマ4分木リーフのマルチタイプ木分割から生じるクロマコーディングユニットのデフォルトの最大階層構造深さを指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_max_mtt_hierarchy_depth_chromaによってデフォルトの最大階層構造深さを上書きできる。sps_max_mtt_hierarchy_depth_intra_slice_chromaの値は、両端を含む0 ~ 2 * (CtbLog 2 SizeY-MinCbLog 2 SizeY) の範囲であるものとする。存在しないとき、sps_max_mtt_hierarchy_depth_intra_slice_chromaの値は0に等しいと推定される。

40

sps_log 2_diff_max_bt_min_qt_intra_slice_chromaは、2分割を使用して分割できるクロマコーディングブロックのルマサンプルのうち最大サイズ(幅又は高さ)の底2の対数と、slice_typeがSPSを参照する2(I)に等しいスライス内のtreeTypeがDUAL_

50

TREE_CHROMAに等しいクロマCTUの4分木分割によって生じるクロマリーフブロックのルマサンプルのうちの最小サイズ（幅又は高さ）との間のデフォルトの差を指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_log2_diff_max_bt_min_qt_chromaによってデフォルトの差を上書きできる。sps_log2_diff_max_bt_min_qt_intra_slice_chromaの値は、0 ~ CtbLog2SizeY-MinQtLog2SizeIntraCの範囲に包含されるものとする。sps_log2_diff_max_bt_min_qt_intra_slice_chromaが存在しないとき、sps_log2_diff_max_bt_min_qt_intra_slice_chromaの値は0に等しいと推定される。

sps_log2_diff_max_tt_min_qt_intra_slice_chromaは、3分割を使用して分割できるクロマコーディングブロックのルマサンプルのうち最大サイズ（幅又は高さ）の底2の対数と、slice_typeがSPSを参照する2(1)に等しいスライス内のtreeTypeがDUAL_TREE_CHROMAに等しいクロマCTUの4分木分割によって生じるクロマリーフブロックのルマサンプルのうちの最小サイズ（幅又は高さ）との間のデフォルトの差を指定する。partition_constraints_override_enabled_flagが1に等しい場合、SPSを参照するPHに存在するph_log2_diff_max_tt_min_qt_chromaによってデフォルトの差を上書きできる。sps_log2_diff_max_tt_min_qt_intra_slice_chromaの値は、0 ~ CtbLog2SizeY-MinQtLog2SizeIntraCの範囲に包含されるものとする。sps_log2_diff_max_tt_min_qt_intra_slice_chromaが存在しないとき、sps_log2_diff_max_tt_min_qt_intra_slice_chromaの値は0に等しいと推定される。

sps_max_luma_transform_size_64_flagが1に等しい場合、ルマサンプルの最大変換サイズが64に等しいことを指定する。sps_max_luma_transform_size_64_flagが0に等しい場合、ルマサンプルの最大変換サイズが32に等しいことを指定する。

CtbSizeYが64未満の場合、sps_max_luma_transform_size_64_flagの値は0に等しいものとする。

変数MinTbLog2SizeY、MaxTbLog2SizeY、MinTbSizeY、及びMaxTbSizeYは、以下のように導出される：

【数9】

$$\text{MinTbLog2SizeY} = 2 \quad (58)$$

$$\text{MaxTbLog2SizeY} = \text{sps_max_luma_transform_size_64_flag} ? 6 : 5 \quad (59)$$

$$\text{MinTbSizeY} = 1 \ll \text{MinTbLog2SizeY} \quad (60)$$

$$\text{MaxTbSizeY} = 1 \ll \text{MaxTbLog2SizeY} \quad (61)$$

sps_joint_cbr_enabled_flagが0に等しい場合、クロマ残差の結合コーディングが無効になる。sps_joint_cbr_enabled_flagが1に等しい場合、クロマ残差の結合コーディングが有効になる。存在しないとき、sps_joint_cbr_enabled_flagの値は0に等しいと推定される。

same_qp_table_for_chromaが1に等しい場合、1つのクロマQPマッピングテーブルのみがシグナリングされ、このテーブルは、sps_joint_cbr_enabled_flagが1に等しい場合に、Cb及びCrの残差、更に結合Cb - Cr残差にも適用されることを指定する。same_qp_table_for_chromaが0に等しい場合、sps_joint_cbr_enabled_flagが1に等しい場合に、Cb及びCrのために2つ、及び結合Cb - Crのためにもう1つの、クロマQPマッピングテーブルがSPSでシグナリングされることを指定する。same_qp_table_for_chromaがビットストリーム内に存在しないとき、same_qp_table_for_chromaの値は1に等しいと推定される。

qp_table_start_minus_26[i]に26を加えたものは、i番目のクロマQPマッピングテ

ーブルを記述するために使用される開始ルマ及びクロマQPを指定する。qp_table_start_minus 26 [i]の値は両端を含む - 26 - QpBdOffset ~ 36 の範囲であるものとする。qp_table_start_minus 26 [i]がビットストリーム内に存在しないとき、qp_table_start_minus 26 [i]の値は0に等しいと推定される。

num_points_in_qp_table_minus 1 [i]に1を加えたものは、i番目のクロマQPマッピングテーブルを記述するために使用されるポイントの数を指定する。num_points_in_qp_table_minus 1 [i]の値は、両端を含む0 ~ 63 + QpBdOffsetの範囲であるものとする。num_points_in_qp_table_minus 1 [0]がビットストリーム内に存在しないとき、num_points_in_qp_table_minus 1 [0]の値は0に等しいと推定される。

delta_qp_in_val_minus 1 [i][j]はi番目のクロマQPマッピングテーブルのj番目のピボットポイントの入力座標を導出するために使用されるデルタ値を指定する。delta_qp_in_val_minus 1 [0][j]がビットストリーム内に存在しないとき、delta_qp_in_val_minus 1 [0][j]の値は0に等しいと推定される。

delta_qp_diff_val[i][j]はi番目のクロマQPマッピングテーブルのj番目のピボットポイントの出力座標を導出するために使用されるデルタ値を指定する。

i = 0 . . numQpTables - 1 の場合に、i番目のクロマQPマッピングテーブルChromaQpTable[i]は以下のように導出される：

【数 10】

```

qpInVal[ i ][ 0 ] = qp_table_start_minus26[ i ] + 26
qpOutVal[ i ][ 0 ] = qpInVal[ i ][ 0 ]
for( j = 0; j <= num_points_in_qp_table_minus1[ i ]; j++ ) {
    qpInVal[ i ][ j + 1 ] = qpInVal[ i ][ j ] +
    delta_qp_in_val_minus1[ i ][ j ] + 1
    qpOutVal[ i ][ j + 1 ] = qpOutVal[ i ][ j ] +
    ( delta_qp_in_val_minus1[ i ][ j ] ^ delta_qp_diff_val[ i ][ j ] )
}
ChromaQpTable[ i ][ qpInVal[ i ][ 0 ] ] = qpOutVal[ i ][ 0 ]
for( k = qpInVal[ i ][ 0 ] - 1; k >= -QpBdOffset; k -- )
    ChromaQpTable[ i ][ k ] = Clip3( -QpBdOffset, 63,
    ChromaQpTable[ i ][ k + 1 ] - 1 )
for( j = 0; j <= num_points_in_qp_table_minus1[ i ]; j++ ) {
    sh = ( delta_qp_in_val_minus1[ i ][ j ] + 1 ) >> 1
    for( k = qpInVal[ i ][ j ] + 1, m = 1; k <= qpInVal[ i ][ j + 1 ]; k++, m++ )
        ChromaQpTable[ i ][ k ] = ChromaQpTable[ i ][ qpInVal[ i ][ j ] ] +
        ( ( qpOutVal[ i ][ j + 1 ] - qpOutVal[ i ][ j ] ) * m + sh ) /
        ( delta_qp_in_val_minus1[ i ][ j ] + 1 )
}
for( k = qpInVal[ i ][ num_points_in_qp_table_minus1[ i ] + 1 ] + 1; k <= 63; k++ )
    ChromaQpTable[ i ][ k ] = Clip3( -QpBdOffset, 63,
    ChromaQpTable[ i ][ k - 1 ] + 1 )

```

same_qp_table_for_chromaが1に等しい場合、両端を含む-QpBdOffset ~ 63 の範囲のkについて、ChromaQpTable[1][k]及びChromaQpTable[2][k]は、ChromaQpTable[0][k]に等しく設定される。

qpInVal[i][j]及びqpOutVal[i][j]の値が、両端を含む0 ~ numQpTables - 1 の範囲のi

、及び両端を含む $0 \sim \text{num_points_in_qp_table_minus_1}[i] + 1$ の j について、 $-\text{QpBdOffset} \sim 63$ の範囲であることが、ビットストリーム準拠の要件である。

$\text{sps_sao_enabled_flag}$ が 1 に等しい場合、サンプル適応オフセット処理は、デブロッキングフィルタ処理の後に再構築されたピクチャに適用される。 $\text{sps_sao_enabled_flag}$ が 0 に等しい場合、サンプル適応オフセット処理は、デブロッキングフィルタ処理の後に再構築されたピクチャに適用されない。

$\text{sps_alf_enabled_flag}$ が 0 に等しい場合、適応ループフィルタが無効であることを指定する。 $\text{sps_alf_enabled_flag}$ が 1 に等しい場合、適応ループフィルタが有効であることを指定する。

$\text{sps_ccalf_enabled_flag}$ が 0 に等しい場合、成分間適応ループフィルタが無効であることを指定する。 $\text{sps_ccalf_enabled_flag}$ が 1 に等しい場合、成分間適応ループフィルタが有効であってよいことを指定する。

$\text{sps_transform_skip_enabled_flag}$ が 1 に等しい場合、変換ユニットシンタックスに $\text{transform_skip_flag}$ が存在する可能性があることを指定する。 $\text{sps_transform_skip_enabled_flag}$ が 0 に等しい場合、 $\text{transform_skip_flag}$ が変換ユニットシンタックスに存在しないことを指定する。

$\log_2_transform_skip_max_size_minus_2$ は、変換スキップに使用される最大ブロックサイズを指定し、両端を含む $0 \sim 3$ の範囲であるものとする。

変数 MaxTsSize は、 $1 \sim (\log_2_transform_skip_max_size_minus_2 + 2)$ に等しく設定される。

$\text{sps_bdpcm_enabled_flag}$ が 1 に等しい場合、 $\text{intra_bdpcm_luma_flag}$ 及び $\text{intra_bdpcm_chroma_flag}$ がイントラコーディングユニットのコーディングユニットシンタックスに存在する可能性があることを指定する。 $\text{sps_bdpcm_enabled_flag}$ が 0 に等しい場合、 $\text{intra_bdpcm_luma_flag}$ 及び $\text{intra_bdpcm_chroma_flag}$ がイントラコーディングユニットのコーディングユニットシンタックスに存在しないことを指定する。存在しないとき、 $\text{sps_bdpcm_enabled_flag}$ の値は 0 に等しいと推定される。

$\text{sps_ref_wraparound_enabled_flag}$ が 1 に等しい場合、インター予測で水平ラップアラウンド動き補償が適用されることを指定する。 $\text{sps_ref_wraparound_enabled_flag}$ が 0 に等しい場合、水平ラップアラウンド動き補償が適用されないことを指定する。 $(\text{CtbSizeY} / \text{MinCbSizeY} + 1)$ の値が $(\text{pic_width_in_luma_samples} / \text{MinCbSizeY} - 1)$ よりも大きい場合、ここで、 $\text{pic_width_in_luma_samples}$ は、SPSを参照する任意のPPSの $\text{pic_width_in_luma_samples}$ の値であり、 $\text{sps_ref_wraparound_enabled_flag}$ の値は 0 に等しいものとする。[Ed. (YK) : ここでのセマンティクスは依然としてPPSシンタックス要素に依存している。]

$\text{sps_temporal_mvp_enabled_flag}$ が 1 に等しい場合、CLVSで時間的動きベクトル予測が使用される可能性があることを指定する。 $\text{sps_temporal_mvp_enabled_flag}$ が 0 に等しい場合、CLVSで時間的動きベクトル予測が使用されないことを指定する。

$\text{sps_sbtmvp_enabled_flag}$ が 1 に等しい場合は、CLVSで slice_type が I と等しくないすべてのスライスを含むピクチャの復号で、サブブロックに基づく時間的動きベクトル予測子を使用できることを指定する。 $\text{sps_sbtmvp_enabled_flag}$ が 0 に等しい場合、CLVSでサブブロックに基づく時間的動きベクトル予測が使用されないことを指定する。 $\text{sps_sbtmvp_enabled_flag}$ が存在しない場合、それは 0 に等しいと推定される。

$\text{sps_amvr_enabled_flag}$ が 1 に等しい場合、動きベクトルコーディングで適応動きベクトル差解像度が使用される。 amvr_enabled_flag が 0 に等しい場合、動きベクトルコーディングで適応動きベクトル差解像度を使用しないことを指定する。

$\text{sps_bdof_enabled_flag}$ が 0 に等しい場合、双方向オプティカルフローのインター予測が無効であることを指定する。 $\text{sps_bdof_enabled_flag}$ が 1 に等しい場合、双方向オプティカルフローのインター予測が有効であることを指定する。

$\text{sps_bdof_pic_present_flag}$ が 1 に等しい場合、SPSを参照するPHに $\text{ph_disable_bdof_flag}$ が存在することを指定する。 $\text{sps_bdof_pic_present_flag}$ が 0 に等しい場合、p

10

20

30

40

50

h_disable_bdof_flagがSPSを参照するPH内に存在しないことを指定する。sps_bdof_pic_present_flagが存在しないとき、sps_bdof_pic_present_flagの値は0に等しいと推定される。

sps_smvd_enabled_flagが1に等しい場合、対称動きベクトル差を動きベクトル復号で使用できることを指定する。sps_smvd_enabled_flagが0に等しい場合、動きベクトルコーディングで対称動きベクトル差を使用しないことを指定する。

sps_dmvr_enabled_flagが1に等しい場合、デコーダの動きベクトル精緻化に基づくインター双予測が有効であることを指定する。sps_dmvr_enabled_flagが0に等しい場合、デコーダの動きベクトル精緻化に基づくインター双予測が無効であることを指定する。

sps_dmvr_pic_present_flagが1に等しい場合、SPSを参照するPHにph_disable_dmvr_flagが存在することを指定する。sps_dmvr_pic_present_flagが0に等しい場合、ph_disable_dmvr_flagがSPSを参照するPH内に存在しないことを指定する。sps_dmvr_pic_present_flagが存在しないとき、sps_dmvr_pic_present_flagの値は0に等しいと推定される。

10

sps_mmvd_enabled_flagが1に等しい場合、動きベクトル差によるマージモードが有効であることを指定する。sps_mmvd_enabled_flagが0に等しい場合、動きベクトル差によるマージモードが無効であることを指定する。

sps_isp_enabled_flagが1に等しい場合、サブパーティションを使用したイントラ予測が有効であることを指定する。sps_isp_enabled_flagが0に等しい場合、サブパーティションを使用したイントラ予測が無効であることを指定する。

20

sps_mrl_enabled_flagが1に等しい場合、複数の参照ラインによるイントラ予測が有効であることを指定する。sps_mrl_enabled_flagが0に等しい場合、複数の参照ラインによるイントラ予測が無効であることを指定する。

sps_mip_enabled_flagが1に等しい場合、マトリクスに基づくイントラ予測が有効であることを指定する。sps_mip_enabled_flagが0に等しい場合、マトリクスに基づくイントラ予測が無効であることを指定する。

sps_cclm_enabled_flagが0に等しい場合、ルマ成分からクロマ成分への成分間線形モデルイントラ予測が無効であることを指定する。sps_cclm_enabled_flagが1に等しい場合、ルマ成分からクロマ成分への成分間線形モデルイントラ予測が有効であることを指定する。sps_cclm_enabled_flagが存在しない場合、それは0に等しいと推定される。

30

sps_chroma_horizontal_collocated_flagが1に等しい場合、予測処理が、対応するルマサンプル位置に対して水平方向にシフトしないクロマサンプル位置用に設計された方法で動作することを指定する。sps_chroma_horizontal_collocated_flagが0に等しい場合、予測処理が、対応するルマサンプル位置に対して、ルマサンプルの単位で0.5だけ右にシフトしたクロマサンプル位置用に設計された方法で動作することを指定する。sps_chroma_horizontal_collocated_flagが存在しない場合、それは、1に等しいと推定される。

sps_chroma_vertical_collocated_flagが1に等しい場合、予測処理が、対応するルマサンプル位置に対して垂直にシフトされないクロマサンプル位置用に設計された方法で動作することを指定する。sps_chroma_vertical_collocated_flagが0に等しい場合、予測処理が、対応するルマサンプル位置に対して、ルマサンプルの単位で0.5だけ下へシフトしたクロマサンプル位置用に設計された方法で動作することを指定する。sps_chroma_vertical_collocated_flagが存在しない場合、それは、1に等しいと推定される。

40

sps_mts_enabled_flagが1に等しい場合、シーケンスパラメータセットRBSPシンタックスに、sps_explicit_mts_intra_enabled_flagが存在し、シーケンスパラメータセットRBSPシンタックスにsps_explicit_mts_inter_enabled_flagが存在することを指定する。sps_mts_enabled_flagが0に等しい場合、シーケンスパラメータセットRBSPシンタックスに、sps_explicit_mts_intra_enabled_flagが存在せず、シーケンスパラメータセットRBSPシンタックスにsps_explicit_mts_inter_enabled_flagが存在しないことを指定する。

50

sps_explicit_mts_intra_enabled_flagが1に等しい場合、イントラコーディングユニットシンタックスにmts_idxが存在する可能性があることを指定する。sps_explicit_mts_intra_enabled_flagが0に等しい場合、イントラコーディングユニットシンタックスにmts_idxが存在しないことを指定する。存在しないとき、sps_explicit_mts_intra_enabled_flagの値は0に等しいと推定される。

sps_explicit_mts_inter_enabled_flagが1に等しい場合、インターコーディングユニットシンタックスにmts_idxが存在する可能性があることを指定する。sps_explicit_mts_inter_enabled_flagが0に等しい場合、インターコーディングユニットシンタックスにmts_idxが存在しないことを指定する。存在しないとき、sps_explicit_mts_inter_enabled_flagの値は0に等しいと推定される。

10

six_minus_max_num_merge_candは、6からSPSでサポートされるマージ動きベクトル予測(MVP)候補の最大数を引いたものを指定する。six_minus_max_num_merge_candの値は、両端を含む0~5の範囲であるものとする。

マージMVP候補の最大数MaxNumMergeCandは次のように導出される：

【数11】

$$\text{MaxNumMergeCand} = 6 - \text{six_minus_max_num_merge_cand} \quad (63)$$

sps_sbt_enabled_flagが0に等しい場合、インター予測CUのサブブロック変換が無効であることを指定する。sps_sbt_enabled_flagが1に等しい場合、インター予測CUのサブブロック変換が有効であることを指定する。

20

sps_affine_enabled_flagは、アフィンモデルに基づく動き補償をインター予測に使用できるかどうかを指定する。sps_affine_enabled_flagが0に等しい場合、シンタックスが制限されて、CLVSでアフィンモデルに基づく動き補償が使用されないように、及びCLVSのコーディングユニットシンタックスにinter_affine_flagとcu_affine_type_flagが存在しないようにする。それ以外の場合(sps_affine_enabled_flagが1に等しい)、CLVSでアフィンモデルに基づく動き補償を使用できる。

five_minus_max_num_subblock_merge_candは、5からSPSでサポートされるマージ動きベクトル予測候補の最大数を引いたものを指定する。

30

sps_affine_type_flagは、6パラメータのアフィンモデルに基づく動き補償をインター予測に使用できるかどうかを指定する。sps_affine_type_flagが0に等しい場合、シンタックスが制限されて、CLVSで6パラメータのアフィンモデルに基づく動き補償が使用されないように、及びCLVSのコーディングユニットシンタックスにcu_affine_type_flagが存在しないようにする。その他の場合(sps_affine_type_flagが1に等しい)、CLVSでアフィンモデルに基づく動き補償を使用できる。存在しないとき、sps_affine_type_flagの値は0に等しいと推定される。

sps_affine_amvr_enabled_flagが1に等しい場合は、アフィンインターモードの動きベクトルコーディングで適応動きベクトル差解像度が使用されることを指定する。sps_affine_amvr_enabled_flagが0に等しい場合は、アフィンインターモードの動きベクトルコーディングで適応動きベクトル差解像度が使用されないことを指定する。存在しないとき、sps_affine_amvr_enabled_flagの値は0に等しいと推定される。

40

sps_affine_prof_enabled_flagは、オプティカルフローによる予測の精密化をアフィン動き補償に使用できるかどうかを指定する。sps_affine_prof_enabled_flagが0に等しい場合、アフィン動き補償はオプティカルフローで精密化されないものとする。その他の場合(sps_affine_prof_enabled_flagが1に等しい)、アフィン動き補償はオプティカルフローで精密化できる。存在しないとき、sps_affine_prof_enabled_flagの値は0に等しいと推定される。

sps_prof_pic_present_flagが1に等しい場合、SPSを参照するPHにph_disable_prof_flagが存在することを指定する。sps_prof_pic_present_flagが0に等しい場合、ph

50

_disable_prof_flagがSPSを参照するPH内に存在しないことを指定する。sps_prof_pic_present_flagが存在しないとき、sps_prof_pic_present_flagの値は0に等しいと推定される。

sps_palette_enabled_flagが1に等しい場合、コーディングユニットシンタックスにpred_mode_plt_flagが存在する可能性があることを指定する。sps_palette_enabled_flagが0に等しい場合、コーディングユニットシンタックスにpred_mode_plt_flagが存在しないことを指定する。sps_palette_enabled_flagが存在しない場合、それは0に等しいと推定される。

sps_act_enabled_flagが1に等しい場合は、適応型色変換を使用でき、cu_act_enabled_flagがコーディングユニットシンタックスに存在する可能性があることを指定する。sps_act_enabled_flagが0に等しい場合は、適応型色変換が使用されず、cu_act_enabled_flagはコーディングユニットシンタックスに存在しないことを指定する。sps_act_enabled_flagが存在しない場合、それは0に等しいと推定される。

min_qp_prime_ts_minus4は、変換スキップモードについての最小許容量子化パラメータを次のように指定する：

【数 1 2】

$$\text{QpPrimeTsMin} = 4 + \text{min_qp_prime_ts_minus4} \quad (64)$$

min_qp_prime_ts_minus4の値は、両端を含む0～48の範囲であるものとする。

sps_bcw_enabled_flagは、動き補償をインター予測に使用できるかどうかを指定する。sps_bcw_enabled_flagが0に等しい場合、シンタックスが制限されて、CLVSでCU重みを有する双予測が使用されないように、及びCLVSのコーディングユニットシンタックスにbcw_idxが存在しないようにする。それ以外の場合（sps_bcw_enabled_flagが1に等しい）、CLVSでCU重みを有する双予測を使用できる。

sps_ibc_enabled_flagが1に等しい場合、CLVS内のピクチャの復号でIBC予測モードが使用される可能性があることを指定する。sps_ibc_enabled_flagが0に等しい場合、CLVSでIBC予測モードが使用されないことを指定する。sps_ibc_enabled_flagが存在しない場合、それは0に等しいと推定される。

six_minus_max_num_ibc_merge_candは、6からSPSでサポートされるIBCマージブロックベクトル予測（BVP）候補の最大数を引いたものを指定する。six_minus_max_num_ibc_merge_candの値は、両端を含む0～5の範囲であるものとする。

IBCマージBVP候補の最大数MaxNumIbcMergeCandは次のように導出される：

【数 1 3】

$$\begin{aligned} & \text{if}(\text{sps_ibc_enabled_flag}) \\ & \quad \text{MaxNumIbcMergeCand} = 6 - \text{six_minus_max_num_ibc_merge_cand} \quad (65) \\ & \text{else} \\ & \quad \text{MaxNumIbcMergeCand} = 0 \end{aligned}$$

sps_ciip_enabled_flagは、インターコーディングユニットのコーディングユニットシンタックスにciip_flagが存在する可能性があることを指定する。sps_ciip_enabled_flagが0に等しい場合、インターコーディングユニットのコーディングユニットシンタックスにciip_flagが存在しないことを指定する。

sps_fpel_mmvd_enabled_flagが1に等しい場合は、動きベクトルの差によるマージモードが整数サンプル精度を使用していることを指定する。sps_fpel_mmvd_enabled_flagが0に等しい場合は、動きベクトルの差によるマージモードが分数サンプル精度を使用できることを指定する。

sps_gpm_enabled_flagは、幾何分割に基づく動き補償がインター予測に使用できるか

どうかを指定する。sps_gpm_enabled_flagが0に等しい場合は、CLVSで幾何分割に基づく動き補償が使用されず、merge_gpm_partition_idx、merge_gpm_idx 0、及びmerge_gpm_idx 1がCLVSのコーディングユニットシンタックスに存在しないように、シンタックスが制約されることを指定する。sps_gpm_enabled_flagが1に等しい場合、幾何分割に基づく動き補償がCLVSに使用できることを指定する。存在しないとき、sps_gpm_enabled_flagの値は0に等しいと推定される。

max_num_merge_cand_minus_max_num_gpm_candは、MaxNumMergeCandからSPSでサポートされる幾何分割マージモード候補の最大数を減算したものを指定する。

幾何パーティションマージモード候補の最大数MaxNumIbcMergeCandは次のように導出される：

【数 1 4】

```

if( sps_gpm_enabled_flag && MaxNumMergeCand >= 3 )
    MaxNumGpmMergeCand = MaxNumMergeCand -
        max_num_merge_cand_minus_max_num_gpm_cand          (66)
else if( sps_gpm_enabled_flag && MaxNumMergeCand == 2 )
    MaxNumMergeCand = 2
else
    MaxNumGpmMergeCand = 0

```

MaxNumGpmMergeCandの値は、両端を含む2 ~ MaxNumMergeCandの範囲であるものとする。

sps_lmcs_enabled_flagが1に等しい場合、CLVSでクロマスケールによるルママッピングが使用されることを指定する。sps_lmcs_enabled_flagが0に等しい場合、クロマスケールによるルママッピングがCLVSにおいて使用されないことを指定する。

sps_lfnst_enabled_flagが1に等しい場合、イントラコーディングユニットシンタックスにlfnst_idxが存在する可能性があることを指定する。sps_lfnst_enabled_flagが0に等しい場合、イントラコーディングユニットシンタックスにlfnst_idxが存在しないことを指定する。

sps_ladf_enabled_flagが1に等しい場合、sps_num_ladf_intervals_minus_2、sps_ladf_lowest_interval_qp_offset、sps_ladf_qp_offset[i]、及びsps_ladf_delta_threshold_minus_1[i]がSPSに存在することを指定する。

sps_num_ladf_intervals_minus_2に1を加えたものは、SPSに存在するsps_ladf_delta_threshold_minus_1[i]及びsps_ladf_qp_offset[i]シンタックス要素の数を指定する。sps_num_ladf_intervals_minus_2の値は、両端を含む0 ~ 3の範囲であるものとする。

sps_ladf_lowest_interval_qp_offsetは、8 . 8 . 3 . 6 . 1項で指定されているように、変数qPを導出するために使用されるオフセットを指定する。sps_ladf_lowest_interval_qp_offsetの値は、両端を含む-63 ~ 63の範囲であるものとする。

sps_ladf_qp_offset[i]は、8 . 8 . 3 . 6 . 1項で指定されているように、変数qPを導出するために使用されるオフセットアレイを指定する。sps_ladf_qp_offset[i]の値は、両端を含む-63 ~ 63の範囲であるものとする。

sps_ladf_delta_threshold_minus_1[i]は、i番目のルマ強度レベル間隔の下限を指定するSpsLadfIntervalLowerBound[i]の値を計算するために使用される。sps_ladf_delta_threshold_minus_1[i]の値は、0 ~ 2^{BitDepth}-3の範囲であるものとする。

SpsLadfIntervalLowerBound[0]の値は0に設定されている。

両端を含む0 ~ sps_num_ladf_intervals_minus_2の範囲内のiの各値について、変数SpsLadfIntervalLowerBound[i+1]は次のように導出される：

10

20

30

40

50

【数 1 5】

$$\begin{aligned} \text{SpsLadfIntervalLowerBound}[i+1] = & \text{SpsLadfIntervalLowerBound}[i] & (67) \\ & + \text{sps_ladf_delta_threshold_minus1}[i] + 1 \end{aligned}$$

log2_parallel_merge_level_minus2 に 2 を加えたもの変数 Log2ParMrgLevel の値を指定する。該変数は、8 . 5 . 2 . 3 項に指定されたような空間マージ候補の導出処理、8 . 5 . 5 . 2 項で指定されたようなサブブロックマージモードで動きベクトル及び参照インデックスの導出処理で、及び 8 . 5 . 2 . 1 項の履歴に基づく動きベクトル予測子の更新処理の実施を制御するために、使用される。log2_parallel_merge_level_minus2 の値は、0 ~ CtbLog2SizeY-2 の範囲に包含されるものとする。変数 Log2ParMrgLevel は、以下のように導出される：

10

【数 1 6】

$$\text{Log2ParMrgLevel} = \text{log2_parallel_merge_level_minus2} + 2 \quad (68)$$

sps_scaling_list_enabled_flag が 1 に等しい場合、変換係数のスケーリング処理にスケーリングリストを使用することを指定する。sps_scaling_list_enabled_flag が 0 に等しい場合、変換係数のスケーリング処理にスケーリングリストを使用されないことを指定する。

20

sps_dep_quant_enabled_flag が 0 に等しい場合、SPSを参照するピクチャに対して依存量子化を無効にすることを指定する。sps_dep_quant_enabled_flag が 1 に等しい場合、SPSを参照するピクチャに対して依存量子化を有効にすることを指定する。

sps_sign_data_hiding_enabled_flag が 0 の場合、SPSを参照するピクチャに対して符号ビット非表示を無効にすることを指定する。sps_sign_data_hiding_enabled_flag が 1 の場合、SPSを参照するピクチャに対して符号ビット非表示が有効にされる可能性があることを指定する。sps_sign_data_hiding_enabled_flag が存在しない場合、0 と等しいと推測される。

30

sps_virtual_boundaries_enabled_flag が 1 に等しい場合、仮想境界を跨がるインループフィルタリングの無効化がCLVS内のコーディングされたピクチャに適用される可能性があることを指定する。sps_virtual_boundaries_enabled_flag が 0 に等しい場合、仮想境界を跨がるインループフィルタリングの無効化がCLVS内のコーディングされたピクチャに適用されないことを指定する。インループフィルタリング操作には、デブロッキングフィルタ、サンプル適応オフセットフィルタ、及び適応ループフィルタ操作が含まれる。

sps_virtual_boundaries_present_flag が 1 に等しい場合、仮想境界の情報がSPSでシグナリングされることを指定する。sps_virtual_boundaries_present_flag が 0 に等しい場合、仮想境界の情報がSPSでシグナリングされないことを指定する。SPSで信号がシグナリングされている仮想境界が 1 つ以上ある場合、インループフィルタリング操作は、SPSを参照するピクチャ内の仮想境界に渡り無効になる。インループフィルタリング操作には、デブロッキングフィルタ、サンプル適応オフセットフィルタ、及び適応ループフィルタ操作が含まれる。

40

res_change_in_clvs_allowed_flag の値が 1 に等しい場合、sps_virtual_boundaries_present_flag の値が 0 に等しくなければならないことは、ビットストリーム準拠の要件である。

sps_num_ver_virtual_boundaries は、SPSに存在する sps_virtual_boundaries_pos_x[i] シンタクス要素の数を指定する。sps_num_ver_virtual_boundaries が存在しない場合、0 と等しいと推測される。

sps_virtual_boundaries_pos_x[i] は、ルマサンプルを 8 で割った単位で i 番目の垂直

50

仮想境界の位置を指定する。sps_virtual_boundaries_pos_x[i]の値は、両端を含む $1 \sim \text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$ の範囲とする。[Ed. (VD):pic_width_in_luma_samplesはSPS内ではなくPPS内にある。]

sps_num_hor_virtual_boundariesは、SPSに存在するsps_virtual_boundaries_pos_y[i]シンタックス要素の数を指定する。sps_num_hor_virtual_boundariesが存在しない場合は、0に等しいと推測される。

sps_virtual_boundaries_enabled_flagが1に等しく、sps_virtual_boundaries_present_flagが1に等しい場合、sps_num_ver_virtual_boundariesとsps_num_hor_virtual_boundariesの和は0より大きくなければならない。

sps_virtual_boundaries_pos_y[i]は、ルマサンプルを8で割った単位でi番目の水平仮想境界の位置を指定する。sps_virtual_boundaries_pos_y[i]の値は、両端を含む $1 \sim \text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$ の範囲であるものとする。[Ed. (VD):pic_height_in_luma_samplesはSPS内ではなくPPS内にある。]

sps_general_hrd_params_present_flagが1に等しい場合は、シンタックス構造general_hrd_parameters()がSPS RBSPシンタックス構造内に存在することを指定する。sps_general_hrd_params_present_flagが0に等しい場合は、シンタックス構造general_hrd_parameters()がSPS RBSPシンタックス構造内に存在しないことを指定する。

sps_sublayer_cpb_params_present_flagが1に等しい場合、SPS RBSPのシンタックス構造old_hrd_parameters()が、両端を含む $0 \sim \text{sps_max_sublayers_minus_1}$ の範囲のTemporalIdを持つサブレイヤ表現のHRDパラメータを含むことを指定する。sps_sublayer_cpb_params_present_flagが0に等しい場合、SPS RBSPのシンタックス構造ols_hrd_parameters()が、sps_max_sublayers_minus_1に等しいTemporalIdを持つサブレイヤ表現のHRDパラメータのみを含むことを指定する。sps_max_sublayers_minus_1が0に等しい場合、sps_sublayer_cpb_params_present_flagの値は0に等しいと推定される。

sps_sublayer_cpb_params_present_flagが0に等しい場合、TemporalIdが両端を含む $0 \sim \text{sps_max_sublayers_minus_1} - 1$ の範囲のサブレイヤ表現のHRDパラメータは、TemporalIdがsps_max_sublayers_minus_1に等しいサブレイヤ表現のHRDパラメータと同じであると推定される。これには、fixed_pic_rate_general_flag[i]シンタックス要素から始まり、ols_hrd_parametersシンタックス構造の中の「if(general_vcl_hrd_params_present_flag)」という条件のすぐ下のsublayer_hrd_parameters(i)シンタックス構造までのHRDパラメータが含まれる。

field_seq_flagが1に等しい場合は、CLVSがフィールドを表すピクチャを伝達することを示す。field_seq_flagが0に等しい場合は、CLVSがフレームを表すピクチャを伝達することを示す。general_frame_only_constraint_flagが1に等しい場合、field_seq_flagの値は0に等しくなるものとする。

field_seq_flagが1に等しい場合、フレームフィールド情報SEIメッセージは、CLVS内のすべてのコーディングされたピクチャに対して存在するものとする。

注5：指定された復号処理では、フィールド又はフレームを表すピクチャを異なる方法で処理しない。したがって、フィールドを表す一連の画像は、個々のフィールドのピクチャの寸法でコーディングされる。例えば、1080iフィールドを表すピクチャは、一般的に1920x540のクロップされた出力寸法を持ち、一方で、シーケンスピクチャレートは、一般的にソースフレームレート（一般的に25Hzから30Hzの間）の代わりにソースフィールドのレート（一般的に50Hzから60Hzの間）を表す。

vui_parameters_present_flagが1に等しい場合、シンタックス構造vui_parameters()がSPS RBSPシンタックス構造内に存在することを指定する。vui_parameters_present_flagが0に等しい場合、シンタックス構造vui_parameters()がSPS RBSPシンタックス構造内に存在しないことを指定する。

sps_extension_flagが0に等しい場合は、SPS RBSPシンタックス構造にsps_extens

10

20

30

40

50

ion_data_flagシンタックス要素が存在しないことを指定する。sps_extension_flagが1に等しい場合は、SPS RBSPシンタックス構造にsps_extension_data_flagシンタックス要素が存在することを指定する。

sps_extension_data_flagは、任意の値を有してよい。その存在及び値は、この仕様のこのバージョン内で指定されるプロファイルへのデコーダ準拠性に影響しない。この仕様のこのバージョンに準拠するデコーダは、全てのsps_extension_data_flagシンタックス要素を無視するものとする。

【 0 0 3 5 】

3 . 3 PPSのシンタックスとセマンティクス

最新のVVCドラフトテキストでは、PPSシンタックスとセマンティクスは次の通りである。

10

20

30

40

50

【表 3】

pic_parameter_set_rbsp() {	Descriptor	
pps_pic_parameter_set_id	ue(v)	
pps_seq_parameter_set_id	u(4)	
mixed_nalu_types_in_pic_flag	u(1)	
pic_width_in_luma_samples	ue(v)	
pic_height_in_luma_samples	ue(v)	
pps_conformance_window_flag	u(1)	10
if(pps_conformance_window_flag) {		
pps_conf_win_left_offset	ue(v)	
pps_conf_win_right_offset	ue(v)	
pps_conf_win_top_offset	ue(v)	
pps_conf_win_bottom_offset	ue(v)	
}		
scaling_window_explicit_signalling_flag	u(1)	
if(scaling_window_explicit_signalling_flag) {		
scaling_win_left_offset	ue(v)	20
scaling_win_right_offset	ue(v)	
scaling_win_top_offset	ue(v)	
scaling_win_bottom_offset	ue(v)	
}		
output_flag_present_flag	u(1)	
subpic_id_mapping_in_pps_flag	u(1)	
if(subpic_id_mapping_in_pps_flag) {		
pps_num_subpics_minus1	ue(v)	
pps_subpic_id_len_minus1	ue(v)	
for(i = 0; i <= pps_num_subpic_minus1; i++)		30
pps_subpic_id[i]	u(v)	
}		
no_pic_partition_flag	u(1)	
if(!no_pic_partition_flag) {		
pps_log2_ctu_size_minus5	u(2)	
num_exp_tile_columns_minus1	ue(v)	
num_exp_tile_rows_minus1	ue(v)	
for(i = 0; i <= num_exp_tile_columns_minus1; i++)		
tile_column_width_minus1[i]	ue(v)	40

for(i = 0; i <= num_exp_tile_rows_minus1; i++)		
tile_row_height_minus1[i]	ue(v)	
if(NumTilesInPic > 1)		
rect_slice_flag	u(1)	
if(rect_slice_flag)		
single_slice_per_subpic_flag	u(1)	
if(rect_slice_flag && !single_slice_per_subpic_flag) {		
num_slices_in_pic_minus1	ue(v)	10
if(num_slices_in_pic_minus1 > 0)		
tile_idx_delta_present_flag	u(1)	
for(i = 0; i < num_slices_in_pic_minus1; i++) {		
if(NumTileColumns > 1)		
slice_width_in_tiles_minus1[i]	ue(v)	
if(NumTileRows > 1 && (tile_idx_delta_present_flag tileIdx % NumTileColumns == 0))		
slice_height_in_tiles_minus1[i]	ue(v)	20
if(slice_width_in_tiles_minus1[i] == 0 && slice_height_in_tiles_minus1[i] == 0 && RowHeight[SliceTopLeftTileIdx[i] / NumTileColumns] > 1) {		
num_exp_slices_in_tile[i]	ue(v)	
for(j = 0; j < num_exp_slices_in_tile[i]; j++)		
exp_slice_height_in_ctus_minus1[j]	ue(v)	
i += NumSlicesInTile[i] - 1		
}		
if(tile_idx_delta_present_flag && i < num_slices_in_pic_minus1)		
tile_idx_delta[i]	se(v)	30
}		
}		
loop_filter_across_tiles_enabled_flag	u(1)	
loop_filter_across_slices_enabled_flag	u(1)	
}		
cabac_init_present_flag	u(1)	
for(i = 0; i < 2; i++)		
num_ref_idx_default_active_minus1[i]	ue(v)	
rpl1_idx_present_flag	u(1)	
init_qp_minus26	se(v)	40

cu_qp_delta_enabled_flag	u(1)	
pps_chroma_tool_offsets_present_flag	u(1)	
if(pps_chroma_tool_offsets_present_flag) {		
pps_cb_qp_offset	se(v)	
pps_cr_qp_offset	se(v)	
pps_joint_cbr_qp_offset_present_flag	u(1)	
if(pps_joint_cbr_qp_offset_present_flag)		
pps_joint_cbr_qp_offset_value	se(v)	10
pps_slice_chroma_qp_offsets_present_flag	u(1)	
pps_cu_chroma_qp_offset_list_enabled_flag	u(1)	
}		
if(pps_cu_chroma_qp_offset_list_enabled_flag) {		
chroma_qp_offset_list_len_minus1	ue(v)	
for(i = 0; i <= chroma_qp_offset_list_len_minus1; i++) {		
cb_qp_offset_list[i]	se(v)	
cr_qp_offset_list[i]	se(v)	
if(pps_joint_cbr_qp_offset_present_flag)		20
joint_cbr_qp_offset_list[i]	se(v)	
}		
}		
}		
pps_weighted_pred_flag	u(1)	
pps_weighted_bipred_flag	u(1)	
deblocking_filter_control_present_flag	u(1)	
if(deblocking_filter_control_present_flag) {		
deblocking_filter_override_enabled_flag	u(1)	
pps_deblocking_filter_disabled_flag	u(1)	30
if(!pps_deblocking_filter_disabled_flag) {		
pps_beta_offset_div2	se(v)	
pps_tc_offset_div2	se(v)	
pps_cb_beta_offset_div2	se(v)	
pps_cb_tc_offset_div2	se(v)	
pps_cr_beta_offset_div2	se(v)	
pps_cr_tc_offset_div2	se(v)	
}		
}		
rpl_info_in_ph_flag	u(1)	40

10

20

30

40

50

if(deblocking_filter_override_enabled_flag)		
dbf_info_in_ph_flag	u(1)	
sao_info_in_ph_flag	u(1)	
alf_info_in_ph_flag	u(1)	
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && rpl_info_in_ph_flag)		
wp_info_in_ph_flag	u(1)	10
qp_delta_info_in_ph_flag	u(1)	
pps_ref_wraparound_enabled_flag	u(1)	
if(pps_ref_wraparound_enabled_flag)		
pps_ref_wraparound_offset	ue(v)	
picture_header_extension_present_flag	u(1)	
slice_header_extension_present_flag	u(1)	
pps_extension_flag	u(1)	20
if(pps_extension_flag)		
while(more_rbsp_data())		
pps_extension_data_flag	u(1)	
rbsp_trailing_bits()		
}		

実施形態では、PPS RBSPは、参照されPPS NALユニットのTemporalId以下のTemporalIdを有する少なくとも1つのAUに含まれ、又は外部手段を通じて提供される前に、復号処理に利用可能であってよい。

PU内で特定の値のpps_pic_parameter_set_idを有する全部のPPS NALユニットは同じコンテンツを有するものとする。

pps_pic_parameter_set_idは、他のシンタックス要素による参照のためにPPSを識別する。pps_pic_parameter_set_idの値は、両端を含む0～63の範囲であるものとする。

PPS NALユニットは、nuh_layer_idの値に関係なく、pps_pic_parameter_set_idの同じ値空間を共有する。

ppsLayerIdを特定のPPS NALユニットのnuh_layer_idの値とし、vclLayerIdを特定のVCL NALユニットのnuh_layer_idの値とする。ppsLayerIdがvclLayerId以下で、nuh_layer_idがppsLayerIdと等しいレイヤが、nuh_layer_idがvclLayerIdと等しいレイヤを含む少なくとも1つのOLSに含まれていない限り、特定のVCL NALユニットは特定のPPSNALユニットを参照しないものとする。

pps_seq_parameter_set_idは、SPSのsps_seq_parameter_set_idの値を指定する。pps_seq_parameter_set_idの値は、両端を含む0～15の範囲であるものとする。pps_seq_parameter_set_idの値は、CLVS内のコーディングピクチャにより参照される全部のPPSの中で同じであるものとする。

mixed_nal_types_in_pic_flagが1に等しい場合は、PPSを参照する各ピクチャが複数のVCL NALユニットを有し、VCL NALユニットが同じnal_unit_type値を有さず、

10

20

30

40

50

ピクチャがIRAPピクチャではないことを指定する。mixed_nalu_types_in_pic_flagが0に等しい場合、PPSを参照する各ピクチャには1つ以上のVCL NALユニットを有し、PPSを参照する各ピクチャのVCL NALユニットは同じ値のnal_unit_typeを持つことを指定する。

no_mixed_nalu_types_in_pic_constraint_flagが1に等しい場合、mixed_nalu_types_in_pic_flagの値は0に等しくなるものとする。

nal_unit_typeの値が両端を含むIDR_W_RADL~CRA_NUTまでの範囲にあるnalUnitTypeAの各スライスについて、nal_unit_typeの別の値(つまり、ピクチャpicAのmixed_nalu_types_in_pic_flagの値は1に等しい)を持つ1つ以上のスライスも含むピクチャpicAでは、次が適用される：

- スライスは、対応するsubpic_treated_as_pic_flag[i]の値が1に等しいサブピクチャsubpicAに属するものとする、
- スライスは、nal_unit_typeがnalUnitTypeAと等しくないVCL NALユニットを含むpicAのサブピクチャに属してはならない、
- nalUnitTypeAがCRAに等しい場合、復号順序と出力順序でCLVS内の現在ピクチャに続く以降のすべてのPUについて、それらのPU内のsubpicA内のスライスのRefPicList[0]もRefPicList[1]も、アクティブエントリ内の復号順序でpicAより前のピクチャを含まないものとする、
- それ以外の場合(つまり、nalUnitTypeAはIDR_W_RADL又はIDR_N_LPと等しい)、復号順序で現在ピクチャに続くCLVS内のすべてのPUについて、それらのPU内のsubpicA内のスライスのRefPicList[0]もRefPicList[1]も、アクティブエントリ内の復号順序でpicAより前のピクチャを含まないものとする。

注1：mixed_nalu_types_in_pic_flagが1に等しい場合は、PPSを参照するピクチャが異なるNALユニットタイプのスライスを含むことを示す。例えば、サブピクチャビットストリームのマージ操作から発生するコーディングされたピクチャでは、エンコーダは一致するビットストリーム構造と元のビットストリームのパラメータのさらなる整合を保証する必要がある。このような整合の例は次の通りである：sps_idr_rpl_flagの値が0に等しく、mixed_nalu_types_in_pic_flagが1に等しい場合、PPSを参照するピクチャは、IDR_W_RADL又はIDR_N_LPと等しいnal_unit_typeのスライスを有することはできない。

pic_width_in_luma_samplesは、ルマサンプルの単位でPPSを参照する各復号ピクチャの幅を指定する。pic_width_in_luma_samplesは0に等しくてはならない、 $\text{Max}(8, \text{MinCbSizeY})$ の整数倍であるものとする、及びpic_width_max_in_luma_samples以下であるものとする。

res_change_in_clvs_allowed_flagが0に等しい場合、pic_width_in_luma_samplesの値はpic_width_max_in_luma_samplesに等しいものとする。

pic_height_in_luma_samplesは、ルマサンプルの単位でPPSを参照する各復号ピクチャの高さを指定する。pic_height_in_luma_samplesは0に等しくてはならない、 $\text{Max}(8, \text{MinCbSizeY})$ の整数倍であるものとする、及びpic_height_max_in_luma_samples以下であるものとする。

res_change_in_clvs_allowed_flagが0に等しい場合、pic_height_in_luma_samplesの値はpic_height_max_in_luma_samplesに等しいものとする。

変数PicWidthInCtbsY、PicHeightInCtbsY、PicSizeInCtbsY、PicWidthInMinCbSizeY、PicHeightInMinCbSizeY、PicSizeInMinCbSizeY、PicSizeInSamplesY、PicWidthInSamplesC、及びPicHeightInSamplesCは、以下のように導出される：

【数17】

10

20

30

40

50

$$\text{PicWidthInCtbsY} = \text{Ceil}(\text{pic_width_in_luma_samples} \div \text{CtbSizeY}) \quad (69)$$

$$\text{PicHeightInCtbsY} = \text{Ceil}(\text{pic_height_in_luma_samples} \div \text{CtbSizeY}) \quad (70)$$

$$\text{PicSizeInCtbsY} = \text{PicWidthInCtbsY} * \text{PicHeightInCtbsY} \quad (71)$$

$$\text{PicWidthInMinCbsY} = \text{pic_width_in_luma_samples} / \text{MinCbSizeY} \quad (72)$$

$$\text{PicHeightInMinCbsY} = \text{pic_height_in_luma_samples} / \text{MinCbSizeY} \quad (73)$$

$$\text{PicSizeInMinCbsY} = \text{PicWidthInMinCbsY} * \text{PicHeightInMinCbsY} \quad (74)$$

$$\text{PicSizeInSamplesY} = \text{pic_width_in_luma_samples} * \text{pic_height_in_luma_samples} \quad (75)$$

$$\text{PicWidthInSamplesC} = \text{pic_width_in_luma_samples} / \text{SubWidthC} \quad (76)$$

$$\text{PicHeightInSamplesC} = \text{pic_height_in_luma_samples} / \text{SubHeightC} \quad (77)$$

pps_conformance_window_flagが1に等しい場合は、適合クロッピングウィンドウオフセットパラメータがPPS内で次に続くことを示す。pps_conformance_window_flagが0に等しい場合は、適合クロッピングウィンドウオフセットパラメータがPPS内に存在しないことを示す。

pps_conf_win_left_offset、pps_conf_win_right_offset、pps_conf_win_top_offset、及びpps_conf_win_bottom_offsetは、出力のためのピクチャ座標の中で指定される長方形領域の観点から、復号処理から出力されるCLVS内のピクチャのサンプルを指定する。sps_conformance_window_flagが0の場合、sps_conf_win_left_offset、sps_conf_win_right_offset、sps_conf_win_top_offset、sps_conf_win_bottom_offsetの値は0に等しいと推定される。

適合クロップウィンドウは、両端を含む、SubWidthC*pps_conf_win_left_offset ~ pic_width_in_luma_samples - (SubWidthC*pps_conf_win_right_offset + 1)の水平ピクチャ座標と、SubHeightC*pps_conf_win_top_offset ~ pic_height_in_luma_samples - (SubHeightC*pps_conf_win_bottom_offset + 1)の垂直ピクチャ座標と、を含むルマサンプルを含む。

SubWidthC * (pps_conf_win_left_offset + pps_conf_win_right_offset)の値は、pic_width_in_luma_samplesより小さいものとし、SubHeightC * (pps_conf_win_top_offset + pps_conf_win_bottom_offset)の値は、pic_height_in_luma_samplesより小さいものとする。

ChromaArrayTypeが0に等しくないとき、2つのクロマアレイの対応する指定されたサンプルは、ピクチャ座標(x/SubWidthC,y/SubHeightC)を有するサンプルである。ここで、(x,y)は指定されたルマサンプルのピクチャ座標である。

注2：適合クロッピングウィンドウオフセットパラメータは出力においてのみ適用される。すべての内部復号処理は、トリミングされていないピクチャサイズに適用される。

ppsA及びppsBを、同じSPSを参照する任意の2つのPPSであるとする。ppsAとppsBが各々、同じ値のpic_width_in_luma_samplesとpic_height_in_luma_samplesを有する場合、ppsAとppsBが各々、同じ値のpps_conf_win_left_offset、pps_conf_win_right_offset、pps_conf_win_top_offset、pps_conf_win_bottom_offsetを有することが、ビットストリーム準拠の要件である。

pic_width_in_luma_samplesがpic_width_max_in_luma_samplesと等しく、pic_height_in_luma_samplesがpic_height_max_in_luma_samplesと等しい場合、pps_conf_win_left_offset、pps_conf_win_right_offset、pps_conf_win_top_offset、及びpps_conf_win_bottom_offsetが各々sps_conf_win_left_offset、sps_conf_win

10

20

30

40

50

_right_offset、sps_conf_win_top_offset、及びsps_conf_win_bottom_offsetと等しいことが、ビットストリーム準拠の要件となる。

scaling_window_explicit_signalling_flagが1に等しい場合は、スケーリングウィンドウオフセットパラメータがPPS内に存在することを指定する。scaling_window_explicit_signalling_flagが0に等しい場合は、スケーリングウィンドウオフセットパラメータがPPS内に存在しないことを指定する。res_change_in_clvs_allowed_flagが0に等しい場合、scaling_window_explicit_signalling_flagの値は0に等しいものとする。

scaling_win_left_offset、scaling_win_right_offset、scaling_win_top_offset、scaling_win_bottom_offsetは、倍率計算のためにピクチャサイズに適用されるオフセットを指定する。存在しない場合、scaling_win_left_offset、scaling_win_right_offset、scaling_win_top_offset、及びscaling_win_bottom_offsetの値は、各々pps_conf_win_left_offset、pps_conf_win_right_offset、pps_conf_win_top_offset、及びpps_conf_win_bottom_offsetと等しいと推定される。

10

SubWidthC * (scaling_win_left_offset + scaling_win_right_offset) の値は、pic_width_in_luma_samples未満であり、SubHeightC * (scaling_win_top_offset + scaling_win_bottom_offset) の値は、pic_height_in_luma_samples未満であるものとする。

変数PicOutputWidthL及びPicOutputHeightLは、以下のように導出される：

【数 1 8】

$$\text{PicOutputWidthL} = \text{pic_width_in_luma_samples} - \text{SubWidthC} * (\text{scaling_win_right_offset} + \text{scaling_win_left_offset}) \quad (78)$$

20

$$\text{PicOutputHeightL} = \text{pic_height_in_luma_samples} - \text{SubWidthC} * (\text{scaling_win_bottom_offset} + \text{scaling_win_top_offset}) \quad (79)$$

refPicOutputWidthL及びrefPicOutputHeightLを、各々このPPSを参照する現在ピクチャの参照ピクチャのPicOutputWidthL及びPicOutputHeightLとする。ビットストリーム準拠の要件は、以下の条件の全部が満たされることである：

- PicOutputWidthL * 2 はrefPicWidthInLumaSamples以上とする、
- PicOutputHeightL * 2 はrefPicHeightInLumaSamples以上とする、
- PicOutputWidthLはrefPicWidthInLumaSamples * 8 以下とする、
- PicOutputHeightL * 2 はrefPicHeightInLumaSamples * 8 以下とする、
- PicOutputWidthL * pic_width_max_in_luma_samplesは、refPicOutputWidthL * (pic_width_in_luma_samples - Max (8, MinCbSizeY)) 以上とする、
- PicOutputHeightL * pic_height_max_in_luma_samplesは、refPicOutputHeightL * (pic_height_in_luma_samples - Max (8, MinCbSizeY)) 以上とする。

30

output_flag_present_flagが1に等しい場合は、PPSを参照するスライスヘッダにpic_output_flagシンタックス要素が存在することを示す。output_flag_present_flagが0に等しい場合は、PPSを参照するスライスヘッダにpic_output_flagシンタックス要素が存在しないことを示す。

40

subpic_id_mapping_in_pps_flagが1に等しい場合は、サブピクチャIDマッピングがPPSでシグナルされることを指定する。subpic_id_mapping_in_pps_flagが0に等しい場合は、サブピクチャIDマッピングがPPSでシグナルされないことを指定する。subpic_id_mapping_explicitly_signalled_flagが0であるか、又はsubpic_id_mapping_in_pps_flagが1に等しい場合、subpic_id_mapping_in_pps_flagの値は0に等しくなるものとする。それ以外の場合 (subpic_id_mapping_explicitly_signalled_flagが1に等しく、subpic_id_mapping_in_sps_flagが0に等しい)、subpic_id_mapping_in_pps_flagの値は1に等しくなるものとする。

pps_num_subpics_minus_1は、sps_num_subpics_minus_1と等しいものとする。

50

pps_subpic_id_len_minus 1 は、sps_subpic_id_len_minus 1 と等しいものとする。

pps_subpic_id[i]はi番目のサブピクチャのサブピクチャIDを指定する。pps_subpic_id[i]シンタックス要素の長さは、pps_subpic_id_len_minus 1 + 1 ビットである。

変数SubpicIdVal[i]は、0 からsps_num_subpics_minus 1 までの範囲のiの各値について、次のように導出される：

【数 1 9】

```

for( i = 0; i <= sps_num_subpics_minus1; i++ )
    if( subpic_id_mapping_explicitly_signalled_flag )
        SubpicIdVal[ i ] = subpic_id_mapping_in_pps_flag ? pps_subpic_id[ i ] :
sps_subpic_id[ i ]                (80)
    else
        SubpicIdVal[ i ] = i

```

以下の両方の制約が適用されることは、ビットストリーム準拠の要件である：

- 両端を含む 0 からsps_num_subpics_minus 1 までの範囲のiとjの2つの異なる値について、SubpicIdVal[i]がSubpicIdVal[j]と等しくないものとする、

- 現在ピクチャがCLVSの第1ピクチャでない場合、両端を含む 0 からsps_num_subpics_minus 1 までの範囲のiの各値について、SubpicIdVal[i]の値が同じレイヤの復号順序で前のピクチャのSubpicIdVal[i]の値と等しくない場合、サブピクチャインデックスiを持つ現在ピクチャのサブピクチャのすべてのコーディングされたスライスNALユニットのnal_unit_typeは、両端を含むIDR_W_RADLからCRA_NUTまでの範囲の特定の値に等しくなるものとする。

no_pic_partition_flagが1に等しい場合は、PPSを参照する各ピクチャにピクチャパーティションが適用されないことを指定する。no_pic_partition_flagが0に等しい場合は、PPSを参照する各ピクチャが複数のタイル又はスライスにパーティションされてよいことを指定する。

ビデオビットストリーム準拠の要件は、no_pic_partition_flagの値が、CLVS内のコーディングピクチャにより参照される全部のPPSについて同じであることである。

sps_num_subpics_minus 1 + 1 の値が1より大きい場合、no_pic_partition_flagの値が1に等しくないことがビットストリーム準拠の要件である。

pps_log2_ctu_size_minus 5 に 5 を加えたものは、各CTUのルマコーディングツリーブロックサイズを指定する。pps_log2_ctu_size_minus 5 は、sps_log2_ctu_size_minus 5 と等しいものとする。

num_exp_tile_columns_minus 1 に 1 を加えたものは、明示的に提供されたタイルの列幅の数値を指定する。num_exp_tile_columns_minus 1 の値は、両端を含む 0 からPicWidthInCtbsY- 1 の範囲であるものとする。no_pic_partition_flagが1に等しい場合、num_exp_tile_columns_minus 1 の値は0に等しいと推定される。

num_exp_tile_rows_minus 1 に 1 を加えたものは、明示的に提供されたタイルの行高さの数値を指定する。num_exp_tile_rows_minus 1 の値は、両端を含む 0 からPicHeightInCtbsY- 1 の範囲であるものとする。no_pic_partition_flagが1に等しい場合、num_exp_tile_rows_minus 1 の値は0に等しいと推定される。

tile_column_width_minus 1 [i]に 1 を加えたものは、両端を含む 0 からnum_exp_tile_columns_minus 1 - 1 までのiについて、i番目のタイル列の幅をCTBの単位で指定する。tile_column_width_minus 1 [num_exp_tile_columns_minus 1]は、6 . 5 . 1 項で指定されているように、インデックスがnum_exp_tile_columns_minus 1 以上のタイル列の幅を導出するために使用される。tile_column_width_minus 1 [i]の値は、両端を含む 0 からPicWidthInCtbsY- 1 の範囲であるものとする。存在しないとき、tile_colu

`mn_width_minus 1 [0]`の値は`PicWidthInCtbsY- 1`に等しいと推定される。

`tile_row_height_minus 1 [i]`に1を加えたものは、両端を含む0から`num_exp_tile_rows_minus 1 - 1`までの*i*について、*i*番目のタイル行の高さをCTBの単位で指定する。`tile_row_height_minus 1 [num_exp_tile_rows_minus 1]`は、6 . 5 . 1項で指定されているように、インデックスが`num_exp_tile_rows_minus 1`以上のタイル行の高さを導出するために使用される。`tile_row_height_minus 1`の値は、両端を含む0から`PicHeightInCtbsY- 1`の範囲であるものとする。存在しないとき、`tile_row_height_minus 1 [0]`の値は`PicHeightInCtbsY- 1`に等しいと推定される。

`rect_slice_flag`が0に等しいことは、各スライス内のタイルがラストスキャン順であり、スライス情報がPPSでシグナリングされないことを指定する。`rect_slice_flag`が1に等しいことは、各スライス内のタイルが、ピクチャの長方形領域をカバーし、スライス情報がPPS内でシグナリングされることを指定する。存在しない場合、`rect_slice_flag`は、1に等しいと推定される。`subpic_info_present_flag`が1に等しいとき、`rect_slice_flag`の値は1に等しいものとする。

10

`single_slice_per_subpic_flag`が1に等しいことは、各サブピクチャが1つ及び1つのみの長方形スライスで構成されることを指定する。`single_slice_per_subpic_flag`が0に等しいことは、各サブピクチャが1つ以上の長方形スライスで構成されてよいことを指定する。`single_slice_per_subpic_flag`が1に等しい場合、`num_slices_in_pic_minus 1`は`sps_num_subpics_minus 1`に等しいと推定される。存在しないとき、`single_slice_per_subpic_flag`の値は0に等しいと推定される。

20

`num_slices_in_pic_minus 1`に1を加えたものは、PPSを参照する各ピクチャ内の長方形スライスの数を指定する。`num_slices_in_pic_minus 1`の値は、両端を含む0から`MaxSlicesPerPicture- 1`の範囲であるものとし、`MaxSlicesPerPicture`はAnnex Aにおいて指定される。`no_pic_partition_flag`が1に等しい場合、`num_slices_in_pic_minus 1`の値は0に等しいと推定される。

`tile_idx_delta_present_flag`が0に等しいことは、`tile_idx_delta`値がPPSに存在せず、PPSを参照するピクチャ内のすべての長方形スライスが6 . 5 . 1項で定義された処理に従ってラスト順序で指定されることを指定する。`tile_idx_delta_present_flag`が1に等しいことは、`tile_idx_delta`値がPPSに存在する可能性があり、PPSを参照するピクチャ内のすべての長方形スライスが`tile_idx_delta`の値によって示される順序で指定されることを指定する。存在しないとき、`tile_idx_delta_present_flag`の値は0に等しいと推定される。

30

`slice_width_in_tiles_minus 1 [i]`に1を加えたものは、*i*番目のスライスの幅をタイル列の単位で指定する。`slice_width_in_tiles_minus 1`の値は、両端を含む0から`NumTileColumns- 1`の範囲であるものとする。

`slice_width_in_tiles_minus 1 [i]`が存在しない場合、以下が適用される：

- `NumTileColumns`が1に等しい場合、`slice_width_in_tiles_minus 1 [i]`の値は0に等しいと推定される、

- それ以外の場合、`slice_width_in_tiles_minus 1 [i]`の値は6 . 5 . 1項で指定されているように推定される。

40

`slice_height_in_tiles_minus 1 [i]`に1を加えたものは、*i*番目のスライスの高さをタイル行の単位で指定する。`slice_height_in_tiles_minus 1`の値は、両端を含む0から`NumTileRows- 1`の範囲であるものとする。

`slice_height_in_tiles_minus 1 [i]`が存在しない場合、以下が適用される：

- `NumTileRows`が1に等しいか、又は`tile_idx_delta_present_flag`が0に等しく、`tileIdx%NumTileColumns`が0より大きい場合)、`slice_height_in_tiles_minus 1 [i]`の値は0に等しいと推定される、

- それ以外の場合 (`NumTileRows`が1に等しくなく、`tile_idx_delta_present_flag`が1に等しいか、`tileIdx%NumTileColumns`が0に等しい)、`tile_idx_delta_present_flag`が1に等しいか、`tileIdx%NumTileColumns`が0に等しい場合、`slice_height_in`

50

_tiles_minus 1 [i]の値はslice_height_in_tiles_minus 1 [i- 1]と等しいと推定される。

num_exp_slices_in_tile[i]は、複数の長方形スライスを含む現在タイルで明示的に提供されたスライスの高さの数値を指定する。num_exp_slices_in_tile[i]の値は、両端を含む0 からRowHeight[tileY]- 1 の範囲であるものとし、tileYはi番目のスライスを含むタイル行インデックスである。存在しないとき、num_exp_slices_in_tile[i]の値は0 に等しいと推定される。num_exp_slices_in_tile[i]が0 に等しい場合、変数NumSlicesInTile[i]の値は1 に等しくなるように導出される。

exp_slice_height_in_ctus_minus 1 [j]に1を加えたものは、現在タイルのj番目の長方形スライスの高さをCTU行の単位で指定する。exp_slice_height_in_ctus_minus 1 [j]の値は、両端を含む0 からRowHeight[tileY]- 1 までの範囲であるものとし、tileYは現在タイルのタイル行インデックスである。

num_exp_slices_in_tile[i]が0 より大きい場合、0 からNumSlicesInTile[i]- 1 の範囲のkについて、変数NumSlicesInTile[i]及びSliceHeightInCtusMinus 1 [i + k]は次のように導出される：

【数 2 0】

```
remainingHeightInCtbsY = RowHeight[ SliceTopLeftTileIdx[ i ] / NumTileColumns ]
```

```
numExpSliceInTile = num_exp_slices_in_tile[ i ]
```

```
for( j = 0; j < numExpSliceInTile - 1; j++ ) {
```

```
    SliceHeightInCtusMinus1[ i++ ] = exp_slice_height_in_ctu_minus1[ j ]
```

```
    remainingHeightInCtbsY -= SliceHeightInCtusMinus1[ j ]
```

```
}
```

```
uniformSliceHeightMinus1 = SliceHeightInCtusMinus1[ i - 1 ]
```

(81)

```
while( remainingHeightInCtbsY >= (uniformSliceHeightMinus1 + 1) ) {
```

```
    SliceHeightInCtusMinus1[ i++ ] = uniformSliceHeightMinus1
```

```
    remainingHeightInCtbsY -= (uniformSliceHeightMinus1 + 1)
```

```
    j++
```

```
}
```

```
if( remainingHeightInCtbsY > 0 ) {
```

```
    SliceHeightInCtusMinus1[ i++ ] = remainingHeightInCtbsY
```

```
    j++
```

```
}
```

```
NumSlicesInTile[ i ] = j
```

tile_idx_delta[i]はi番目の長方形スライスの第1タイルのタイルインデックスと(i + 1)番目の長方形スライスの第1タイルのタイルインデックスとの間の差を指定する。tile_idx_delta[i]の値は、両端を含む-NumTilesInPic + 1 からNumTilesInPic - 1 までの範囲にあるものとする。存在しない場合、tile_idx_delta[i]の値は0 に等しいと推定される。存在するとき、tile_idx_delta[i]の値は0 に等しくないものとする。

loop_filter_across_tiles_enabled_flagが1 に等しい場合は、インループフィルタリング操作が、PPSを参照しているピクチャ内のタイル境界に跨がり実行できることを指定する。loop_filter_across_tiles_enabled_flagが0 に等しい場合は、インループフィルタリング操作が、PPSを参照しているピクチャ内のタイル境界に跨がり実行されないことを指定する。インループフィルタリング操作には、デブロッキングフィルタ、サンプル適応オフセットフィルタ、及び適応ループフィルタ操作が含まれる。存在しないとき、loop_filter_across_tiles_enabled_flagの値は1 に等しいと推定される。

loop_filter_across_slices_enabled_flagが1 に等しいことは、インループフィルタ

10

20

30

40

50

リング操作がPPSを参照するピクチャ内のスライス境界に跨がり実行できることを指定する。loop_filter_across_tiles_enabled_flagが0に等しいことは、インループフィルタリング操作が、PPSを参照しているピクチャ内のスライス境界に跨がり実行されないことを指定する。インループフィルタリング操作には、デブロッキングフィルタ、サンプル適応オフセットフィルタ、及び適応ループフィルタ操作が含まれる。存在しないとき、loop_filter_across_slices_enabled_flagの値は0に等しいと推定される。

cabac_init_present_flagが1に等しいことは、PPSを参照するスライスヘッダにcabac_init_flagが存在することを指定する。cabac_init_present_flagが0に等しいことは、PPSを参照するスライスヘッダにcabac_init_flagが存在しないことを指定する。

num_ref_idx_default_active_minus_1[i]に1を加えたものは、iが0に等しい場合、num_ref_idx_active_override_flagが0に等しいPスライス又はBスライスに対して変数NumRefIdxActive[0]の推定値を指定し、iが1に等しい場合、num_ref_idx_active_override_flagが0に等しいBスライスに対してNumRefIdxActive[1]の推定値を指定する。num_ref_idx_default_active_minus_1[i]の値は、両端を含む0～14の範囲であるものとする。

rpl1_idx_present_flagが0に等しいことは、ref_pic_list_sps_flag[1]及びref_pic_list_idx[1]がPHシンタックス構造又はPPSを参照するピクチャのスライスヘッダに存在しないことを指定する。rpl1_idx_present_flagが1に等しいことは、ref_pic_list_sps_flag[1]及びref_pic_list_idx[1]がPHシンタックス構造又はPPSを参照するピクチャのスライスヘッダに存在する可能性あることを指定する。

init_qp_minus_26に26を加えたものは、PPSを参照する各スライスのSliceQpYの初期値を指定する。SliceQpYの初期値は、ph_qp_deltaの0ではない値が復号された場合はピクチャレベルで、slice_qp_deltaの0ではない値が復号された場合はスライスレベルで変更される。init_qp_minus_26の値は、両端を含む-(26 + QpBdOffset)～+37の範囲であるものとする。

cu_qp_delta_enabled_flagが1に等しいことは、PPSを参照するPHにph_cu_qp_delta_subdiv_intra_slice及びph_cu_qp_delta_subdiv_inter_sliceシンタックス要素が存在し、変換ユニットシンタックスにcu_qp_delta_absが存在する可能性あることを指定する。cu_qp_delta_enabled_flagが0に等しいことは、PPSを参照するPHにph_cu_qp_delta_subdiv_intra_slice及びph_cu_qp_delta_subdiv_inter_sliceシンタックス要素が存在せず、変換ユニットシンタックスにcu_qp_delta_absが存在しないことを指定する。

pps_chroma_tool_offsets_present_flagが1に等しい場合、クロマツールオフセット関連のシンタックス要素がPPS RBSPシンタックス構造内に存在することを指定する。pps_chroma_tool_offsets_present_flagが0に等しい場合、クロマツールオフセット関連のシンタックス要素がPPS RBSPシンタックス構造内に存在しないことを指定する。ChromaArrayTypeが0に等しい場合、pps_chroma_tool_offsets_present_flagの値は0に等しいものとする。

pps_cb_qp_offsetとpps_cr_qp_offsetは、各々Qp_{cb}とQp_{cr}を導出するために使用されるルマ量子化パラメータQp_Yへのオフセットを指定する。pps_cb_qp_offset及びpps_cr_qp_offsetの値は、両端を含む-12～+12の範囲であるものとする。ChromaArrayTypeが0に等しい場合、pps_cb_qp_offsetとpps_cr_qp_offsetは復号処理で使用されず、デコーダはそれらの値を無視するものとする。存在しないとき、pps_cb_qp_offset及びpps_cr_qp_offsetの値は0に等しいと推定される。

pps_joint_cbr_qp_offset_present_flagが1に等しいことは、pps_joint_cbr_qp_offset_value及びjoint_cbr_qp_offset_list[i]がPPS RBSPシンタックス構造内に存在することを指定する。pps_joint_cbr_qp_offset_present_flagが0に等しいことは、pps_joint_cbr_qp_offset_value及びjoint_cbr_qp_offset_list[i]がPPS RBSPシンタックス構造内に存在しないことを指定する。ChromaArrayTypeが0に等しいか又はsps_joint_cbr_enabled_flagが0に等しい場合、pps_joint_cbr_qp_offset_present_

10

20

30

40

50

flagの値は0に等しくなる。存在しないとき、pps_joint_cbcrc_qp_offset_present_flagの値は0に等しいと推定される。

pps_joint_cbcrc_qp_offset_valueは、 Qp'_{CbCr} を導出するために使用されるルマ量子化パラメータ Qp'_Y へのオフセットを指定する。pps_joint_cbcrc_qp_offset_valueの値は、両端を含む $-12 \sim +12$ の範囲であるものとする。ChromaArrayTypeが0に等しいか又はsps_joint_cbcrc_enabled_flagが0に等しい場合、pps_joint_cbcrc_qp_offset_valueは復号処理で使用されず、デコーダはその値を無視するものとする。pps_joint_cbcrc_qp_offset_present_flagが0に等しい場合、pps_joint_cbcrc_qp_offset_valueは存在せず、0に等しいと推定される。

pps_slice_chroma_qp_offsets_present_flagが1に等しいことは、slice_cb_qp_offset及びslice_cr_qp_offsetシンタックス要素が関連付けられたスライスヘッダに存在することを指定する。pps_slice_chroma_qp_offsets_present_flagが0に等しいことは、slice_cb_qp_offset及びslice_cr_qp_offsetシンタックス要素が関連付けられたスライスヘッダに存在しないことを指定する。存在しないとき、pps_slice_chroma_qp_offsets_present_flagの値は0に等しいと推定される。

pps_cu_chroma_qp_offset_list_enabled_flagが1に等しいことは、PPSを参照するPHにph_cu_chroma_qp_offset_subdiv_intra_sliceシンタックス要素とph_cu_chroma_qp_offset_subdiv_intra_sliceシンタックス要素が存在し、変換ユニットシンタックスとパレットコーディングシンタックスにcu_chroma_qp_offset_flagが存在する可能性があることを指定する。pps_cu_chroma_qp_offset_list_enabled_flagが0に等しいことは、PPSを参照するPHにph_cu_chroma_qp_offset_subdiv_intra_sliceシンタックス要素とph_cu_chroma_qp_offset_subdiv_intra_sliceシンタックス要素が存在せず、変換ユニットシンタックスとパレットコーディングシンタックスにcu_chroma_qp_offset_flagが存在しないことを指定する。存在しないとき、pps_cu_chroma_qp_offset_list_enabled_flagの値は0に等しいと推定される。

chroma_qp_offset_list_len_minus1に1を加えたものは、PPS RBSPシンタックス構造に存在するシンタックス要素であるcb_qp_offset_list[i]、cr_qp_offset_list[i]、及びjoint_cbcrc_qp_offset_list[i]の数が指定される。chroma_qp_offset_list_len_minus1の値は、両端を含む $0 \sim 5$ の範囲であるものとする。

cb_qp_offset_list[i]、cr_qp_offset_list[i]、及びjoint_cbcrc_qp_offset_list[i]は、各々 Qp_{Cb} 、 Qp_{Cr} 、及び Qp_{CbCr} の導出で使用されるオフセットを指定する。cb_qp_offset_list[i]、cr_qp_offset_list[i]、及びjoint_cbcrc_qp_offset_list[i]の値は、両端を含む $-12 \sim +12$ の範囲であるものとする。pps_joint_cbcrc_qp_offset_present_flagが0に等しい場合、joint_cbcrc_qp_offset_list[i]は存在せず、0に等しいと推定される。

pps_weighted_pred_flagが0に等しいことは、PPSを参照するPスライスに加重予測が適用されないことを指定する。pps_weighted_pred_flagが1に等しいことは、PPSを参照するPスライスに加重予測が適用されることを指定する。sps_weighted_pred_flagが0に等しい場合、pps_weighted_pred_flagの値は0に等しくなるものとする。

pps_weighted_bipred_flagが0に等しい場合は、PPSを参照するBスライスに明示的な加重予測が適用されないことを指定する。pps_weighted_bipred_flagが1に等しい場合は、PPSを参照するBスライスに明示的な加重予測が適用されることを指定する。sps_weighted_bipred_flagが0に等しい場合、pps_weighted_pred_flagの値は0に等しくなるものとする。

deblocking_filter_control_present_flagが1に等しいことは、PPSにデブロッキングフィルタ制御シンタックス要素が存在することを指定する。deblocking_filter_control_present_flagが0に等しいことは、PPSにデブロッキングフィルタ制御シンタックス要素が存在しないことを指定する。

deblocking_filter_override_enabled_flagが1に等しいことは、PPSを参照するPHにph_deblocking_filter_override_flagが存在すること、又はPPSを参照するスライス

10

20

30

40

50

ヘッドにslice_deblocking_filter_override_flagが存在することを指定する。deblocking_filter_override_enabled_flagが0に等しいことは、PPSを参照するPHにph_deblocking_filter_override_flagが、又はPPSを参照するスライスヘッドにslice_deblocking_filter_override_flagが、存在しないことを指定する。存在しないとき、deblocking_filter_override_enabled_flagの値は0に等しいと推定される。

pps_deblocking_filter_disabled_flagが1に等しいことは、デブロッキングフィルタの操作が、slice_deblocking_filter_disabled_flagが存在しないPPSを参照するスライスに適用されないことを指定する。pps_deblocking_filter_disabled_flagが0に等しいことは、デブロッキングフィルタの操作が、slice_deblocking_filter_disabled_flagが存在しないPPSを参照するスライスに適用されないことを指定する。存在しないとき、pps_deblocking_filter_disabled_flagの値は0に等しいと推定される。

10

pps_beta_offset_div2とpps_tc_offset_div2は、PPSを参照するスライスのルマ成分に適用される tC (2で割った値) のデフォルトのデブロッキングパラメータオフセットを指定する。ただし、PPSを参照するスライスのピクチャヘッド又はスライスヘッドに存在するデブロッキングパラメータオフセットによってデフォルトのデブロッキングパラメータオフセットが上書きされる場合は除く。pps_beta_offset_div2とpps_tc_offset_div2の値は、両方とも両端を含む -12 から 12 の範囲内にあるものとする。存在しない場合、pps_beta_offset_div2とpps_tc_offset_div2の値は両方とも0に等しいと推定される。

pps_cb_beta_offset_div2とpps_cb_tc_offset_div2は、PPSを参照するスライスのCbコンポーネントに適用される tC (2で割った値) のデフォルトのデブロッキングパラメータオフセットを指定する。ただし、PPSを参照するスライスのピクチャヘッド又はスライスヘッドに存在するデブロッキングパラメータオフセットによってデフォルトのデブロッキングパラメータオフセットが上書きされる場合は除く。pps_cb_beta_offset_div2とpps_cb_tc_offset_div2の値は、両方とも両端を含む -12 から 12 の範囲内にあるものとする。存在しない場合、pps_cb_beta_offset_div2とpps_cb_tc_offset_div2の値は両方とも0に等しいと推定される。

20

pps_cr_beta_offset_div2とpps_cr_tc_offset_div2は、PPSを参照するスライスのCrコンポーネントに適用される tC (2で割った値) のデフォルトのデブロッキングパラメータオフセットを指定する。ただし、PPSを参照するスライスのピクチャヘッド又はスライスヘッドに存在するデブロッキングパラメータオフセットによってデフォルトのデブロッキングパラメータオフセットが上書きされる場合は除く。pps_cr_beta_offset_div2とpps_cr_tc_offset_div2の値は、両方とも両端を含む -12 から 12 の範囲内にあるものとする。存在しない場合、pps_cr_beta_offset_div2とpps_cr_tc_offset_div2の値は両方とも0に等しいと推定される。

30

rpl_info_in_ph_flagが1に等しいことは、参照ピクチャリスト情報が、PHシンタックス構造に存在し、PHシンタックス構造を含まないPPSを参照するスライスヘッドには存在しないことを指定する。rpl_info_in_ph_flagが0に等しいことは、参照ピクチャリスト情報が、PHシンタックス構造に存在せず、PHシンタックス構造を含まないPPSを参照するスライスヘッドには存在する可能性があることを指定する。

40

dbf_info_in_ph_flagが1に等しい場合、デブロッキングフィルタ情報が、PHシンタックス構造に存在し、PHシンタックス構造を含まないPPSを参照するスライスヘッドには存在しないことを指定する。dbf_info_in_ph_flagが0に等しいことは、デブロッキングフィルタ情報が、PHシンタックス構造に存在せず、PHシンタックス構造を含まないPPSを参照するスライスヘッドに存在する可能性があることを指定する。存在しないとき、dbf_info_in_ph_flagの値は0に等しいと推定される。

sao_info_in_ph_flagが1に等しいことは、SAOフィルタ情報が、PHシンタックス構造に存在し、PHシンタックス構造を含まないPPSを参照するスライスヘッドには存在しないことを指定する。sao_info_in_ph_flagが0に等しいことは、SAOフィルタ情報が、PHシンタックス構造に存在せず、PHシンタックス構造を含まないPPSを参照するスライス

50

ヘッダに存在する可能性があることを指定する。

alf_info_in_ph_flagが1に等しいことは、ALF情報が、PHシンタックス構造に存在し、PHシンタックス構造を含まないPPSを参照するスライスヘッダには存在しないことを指定する。alf_info_in_ph_flagが0に等しいことは、ALFフィルタ情報が、PHシンタックス構造に存在せず、PHシンタックス構造を含まないPPSを参照するスライスヘッダに存在

する可能性があることを指定する。alf_info_in_ph_flagが1に等しいことは、加重予測情報が、PHシンタックス構造に存在し、PHシンタックス構造を含まないPPSを参照するスライスヘッダには存在しないことを指定する。wp_info_in_ph_flagが0に等しいことは、加重予測情報が、PHシンタックス構造に存在せず、PHシンタックス構造を含まないPPSを参照するスライスヘッダに存在

10

する可能性があることを指定する。存在しないとき、wp_info_in_ph_flagの値は0に等しいと推定される。qp_delta_info_in_ph_flagが1に等しいことは、QP 情報が、PHシンタックス構造に存在し、PHシンタックス構造を含まないPPSを参照するスライスヘッダには存在しないことを指定する。qp_delta_info_in_ph_flagが0に等しいことは、QPデルタ情報が、PHシンタックス構造に存在せず、PHシンタックス構造を含まないPPSを参照するスライス

ヘッダに存在する可能性があることを指定する。pps_ref_wraparound_enabled_flagが1に等しい場合、インター予測で水平ラップアラウンド動き補償が適用されることを指定する。pps_ref_wraparound_enabled_flagが0に等しい場合、水平ラップアラウンド動き補償が適用されないことを指定する。CtbSizeY/MinCbSizeY + 1の値がpic_width_in_luma_samples/MinCbSizeY - 1より大きい場合、pps_ref_wraparound_enabled_flagの値は0に等しくなるものとする。sps_ref_wraparound_enabled_flagが0に等しい場合、pps_ref_wraparound_enabled_flagの値は0に等しくなるものとする。

20

pps_ref_wraparound_offsetに $(CtbSizeY/MinCbSizeY) + 2$ を加えたものは、水平ラップアラウンド位置の計算に使用するオフセットをMinCbSizeYルマサンプルの単位で指定する。pps_ref_wraparound_offsetの値は、両端を含む0から $(pic_width_in_luma_samples/MinCbSizeY) - (CtbSizeY/MinCbSizeY) - 2$ までの範囲であるものとする。

変数PpsRefWraparoundOffsetは、 $pps_ref_wraparound_offset + (CtbSizeY/MinCbSizeY) + 2$ に等しく設定されている。

30

picture_header_extension_present_flagが0に等しいことは、PPSを参照するPHにPH拡張シンタックス要素が存在しないことを指定する。picture_header_extension_present_flagが1に等しいことは、PPSを参照するPHにPH拡張シンタックス要素が存在することを指定する。picture_header_extension_present_flagは、このバージョンの仕様に準拠するビットストリームでは0に等しいものとする。

slice_header_extension_present_flagが0に等しいことは、PPSを参照するコーディングされたピクチャのスライスヘッダにスライスヘッダ拡張シンタックス要素が存在しないことを指定する。slice_header_extension_present_flagが1に等しいことは、PPSを参照するコーディングされたピクチャのスライスヘッダにスライスヘッダ拡張シンタックス要素が存在することを指定する。slice_header_extension_present_flagは、このバージョンの仕様に準拠するビットストリームでは0に等しいものとする。

40

pps_extension_flagが0に等しいことは、PPS RBSPシンタックス構造にpps_extension_data_flagシンタックス要素が存在しないことを指定する。pps_extension_flagが1に等しいことは、pps_extension_data_flagシンタックス要素がPPS RBSPシンタックス構造内に存在することを指定する。

pps_extension_data_flagは、任意の値を有してよい。その存在及び値は、この仕様のこのバージョン内で指定されるプロファイルへのデコーダ準拠性に影響しない。この仕様のこのバージョンに準拠するデコーダは、全てのpps_extension_data_flagシンタックス要素を無視するものとする。

50

【 0 0 3 6 】

3.4 APSのシンタックスとセマンティクス

最新のVVCドラフトテキストでは、APSシンタックスとセマンティクスは次の通りである。

【表 4】

adaptation_parameter_set_rbsp() {	Descriptor	10
adaptation_parameter_set_id	u(5)	10
aps_params_type	u(3)	10
if(aps_params_type == ALF_APS)		10
alf_data()		10
else if(aps_params_type == LMCS_APS)		10
lmcs_data()		10
else if(aps_params_type == SCALING_APS)		10
scaling_list_data()		10
aps_extension_flag	u(1)	20
if(aps_extension_flag)		20
while(more_rbsp_data())		20
aps_extension_data_flag	u(1)	20
rbsp_trailing_bits()		20
}		20

【 0 0 3 7 】

APS RBSPはALFシンタックス構造、すなわちalf_data()を含む。

30

40

50

【表 5】

alf_data() {	Descriptor	
alf_luma_filter_signal_flag	u(1)	
alf_chroma_filter_signal_flag	u(1)	
alf_cc_cb_filter_signal_flag	u(1)	
alf_cc_cr_filter_signal_flag	u(1)	
if(alf_luma_filter_signal_flag) {		
alf_luma_clip_flag	u(1)	10
alf_luma_num_filters_signalled_minus1	ue(v)	
if(alf_luma_num_filters_signalled_minus1 > 0)		
for(filtIdx = 0; filtIdx < NumAlfFilters; filtIdx++)		
alf_luma_coeff_delta_idx[filtIdx]	u(v)	
for(sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++)		
for(j = 0; j < 12; j++) {		
alf_luma_coeff_abs[sfIdx][j]	ue(v)	
if(alf_luma_coeff_abs[sfIdx][j])		
alf_luma_coeff_sign[sfIdx][j]	u(1)	20
}		
if(alf_luma_clip_flag)		
for(sfIdx = 0; sfIdx <= alf_luma_num_filters_signalled_minus1; sfIdx++)		
for(j = 0; j < 12; j++)		
alf_luma_clip_idx[sfIdx][j]	u(2)	
}		
if(alf_chroma_filter_signal_flag) {		
alf_chroma_clip_flag	u(1)	
alf_chroma_num_alt_filters_minus1	ue(v)	30
for(altIdx = 0; altIdx <= alf_chroma_num_alt_filters_minus1; altIdx++) {		
for(j = 0; j < 6; j++) {		
alf_chroma_coeff_abs[altIdx][j]	ue(v)	
if(alf_chroma_coeff_abs[altIdx][j] > 0)		
alf_chroma_coeff_sign[altIdx][j]	u(1)	
}		
if(alf_chroma_clip_flag)		
for(j = 0; j < 6; j++)		
alf_chroma_clip_idx[altIdx][j]	u(2)	40

}		
}		
if(alf_cc_cb_filter_signal_flag) {		
alf_cc_cb_filters_signalled_minus1	ue(v)	
for(k = 0; k < alf_cc_cb_filters_signalled_minus1 + 1; k++) {		
for(j = 0; j < 7; j++) {		
alf_cc_cb_mapped_coeff_abs[k][j]	u(3)	10
if(alf_cc_cb_mapped_coeff_abs[k][j])		
alf_cc_cb_coeff_sign[k][j]	u(1)	
}		
}		
}		
if(alf_cc_cr_filter_signal_flag) {		
alf_cc_cr_filters_signalled_minus1	ue(v)	20
for(k = 0; k < alf_cc_cr_filters_signalled_minus1 + 1; k++) {		
for(j = 0; j < 7; j++) {		
alf_cc_cr_mapped_coeff_abs[k][j]	u(3)	
if(alf_cc_cr_mapped_coeff_abs[k][j])		
alf_cc_cr_coeff_sign[k][j]	u(1)	
}		
}		30
}		
}		

【 0 0 3 8 】

APS RBSPはLMCSシンタックス構造、すなわちlmc_data () を含む。

【表 6】

	Descriptor
lmc_data() {	
lmc_min_bin_idx	ue(v)
lmc_delta_max_bin_idx	ue(v)
lmc_delta_cw_prec_minus1	ue(v)
for(i = lmc_min_bin_idx; i <= LmcMaxBinIdx; i++) {	
lmc_delta_abs_cw[i]	u(v)
if(lmc_delta_abs_cw[i] > 0)	
lmc_delta_sign_cw_flag[i]	u(1)
}	
lmc_delta_abs_crs	u(3)
if(lmc_delta_abs_crs > 0)	
lmc_delta_sign_crs_flag	u(1)
}	

10

20

【 0 0 3 9 】

APS RBSPはスケーリングリストデータシンタックス構造、すなわちscaling_list_data()を含む。

30

40

50

【表 7】

scaling_list_data() {	Descriptor	
scaling_matrix_for_lfnst_disabled_flag	u(1)	
scaling_list_chroma_present_flag	u(1)	
for(id = 0; id < 28; id ++)		
matrixSize = (id < 2) ? 2 : ((id < 8) ? 4 : 8)		
if(scaling_list_chroma_present_flag (id % 3 == 2) (id == 27)) {		10
scaling_list_copy_mode_flag[id]	u(1)	
if(!scaling_list_copy_mode_flag[id])		
scaling_list_pred_mode_flag[id]	u(1)	
if((scaling_list_copy_mode_flag[id] scaling_list_pred_mode_flag[id]) && id != 0 && id != 2 && id != 8)		
scaling_list_pred_id_delta[id]	ue(v)	
if(!scaling_list_copy_mode_flag[id]) {		
nextCoef = 0		20
if(id > 13) {		
scaling_list_dc_coef[id - 14]	se(v)	
nextCoef += scaling_list_dc_coef[id - 14]		
}		
for(i = 0; i < matrixSize * matrixSize; i++) {		
x = DiagScanOrder[3][3][i][0]		
y = DiagScanOrder[3][3][i][1]		
if(!(id > 25 && x >= 4 && y >= 4)) {		
scaling_list_delta_coef[id][i]	se(v)	30
nextCoef += scaling_list_delta_coef[id][i]		
}		
ScalingList[id][i] = nextCoef		
}		
}		
}		
}		
}		
}		40

各APS RBSPは、参照される前に、それを参照するか、又は外部手段を通じて提供される、コーディングされたスライスNALユニットのTemporalId以下のTemporalIdを持つ少なくとも1つのAUに含まれている復号処理で利用できるものとする。

特定の値のadaptation_parameter_set_id及び特定の値のaps_params_typeを持つPU内のすべてのAPS NALユニットは、プレフィックス又はサフィックスのAPS NALユニットであるかどうかに関係なく、同じ内容を有するものとする。

adaptation_parameter_set_idは、他のシンタックス要素による参照のためにAPSの識別子を提供する。

aps_params_typeがALF_APS又はSCALING_APSに等しい場合、adaptation_param

eter_set_idの値は、両端を含む0から7の範囲であるものとする。

aps_params_typeがLMCS_APSと等しい場合、adaptation_parameter_set_idの値は、両端を含む0から3までの範囲であるものとする。

apsLayerIdを特定のAPS NALユニットのnuh_layer_idの値とし、vclLayerIdを特定のVCL NALユニットのnuh_layer_idの値とする。apsLayerIdがvclLayerId以下で、nuh_layer_idがapsLayerIdと等しいレイヤが、nuh_layer_idがvclLayerIdと等しいレイヤを含む少なくとも1つのOLSに含まれていない限り、特定のVCL NALユニットは特定のAPS NALユニットを参照しないものとする。

aps_params_typeは、表6で指定されているように、APSで伝送されるAPSパラメータのタイプを指定する。

10

【表8】

表6-APSパラメータタイプコード及びAPSパラメータのタイプ

aps_params_type	aps_params_type の名称	APSパラメータのタイプ
0	ALF_APS	ALFパラメータ
1	LMCS_APS	LMCSパラメータ
2	SCALING_APS	スケーリングリストパラメータ
3..7	Reserved	予約

20

aps_params_typeの特定の値を持つすべてのAPS NALユニットは、nuh_layer_id値に関係なく、adaptation_parameter_set_idに対して同じ値空間を共有する。aps_params_typeの値が異なるAPS NALユニットでは、adaptation_parameter_set_idに個別の値空間が使用される。

注1：APS NALユニット（特定の値のadaptation_parameter_set_id及び特定の値のaps_params_typeを有する）はピクチャ間で共有でき、ピクチャ内の異なるスライスには異なるALF APSを参照できる。

30

注2：特定のVCL NALユニットに関連付けられたサフィックスAPSNALユニット（このVCL NALユニットは、復号順序でサフィックスAPS NALユニットの前にある）は、特定のVCL NALユニットでは使用されず、復号順序でサフィックスAPS NALユニットの後に続くVCL NALユニットで使用される。

aps_extension_flagが0に等しいことは、APS RBSPシンタックス構造にaps_extension_data_flagシンタックス要素が存在しないことを指定する。aps_extension_flagが1に等しいことは、APS RBSPシンタックス構造にaps_extension_data_flagシンタックス要素が存在することを指定する。

aps_extension_data_flagは、任意の値を有してよい。その存在及び値は、この仕様のこのバージョン内で指定されるプロファイルへのデコーダ準拠性に影響しない。この仕様のこのバージョンに準拠するデコーダは、全てのaps_extension_data_flagシンタックス要素を無視するものとする。

40

alf_luma_filter_signal_flagが1に等しいことは、ルマフィルタセットがシグナルされることを指定する。alf_luma_filter_signal_flagが0に等しいことは、ルマフィルタセットがシグナルされないことを指定する。

alf_chroma_filter_signal_flagが1に等しいことは、クロマフィルタがシグナルされることを指定する。alf_chroma_filter_signal_flagが0に等しいことは、クロマフィルタがシグナルされないことを指定する。ChromaArrayTypeが0の場合、alf_chroma_filter_signal_flagは0に等しいものとする。

50

alf_luma_filter_signal_flag、alf_chroma_filter_signal_flag、alf_cc_cb_filter_signal_flag、alf_cc_cr_filter_signal_flagの少なくとも1つの値は1に等しいものとする。

異なる適応ループフィルタの数を指定する変数NumAlfFiltersは25に等しく設定される。

alf_luma_clip_flagが0に等しいことは、線形適応ループフィルタリングがルマ成分に適用されることを指定する。alf_luma_clip_flagが1に等しいことは、非線形適応ループフィルタリングがルマ成分に適用される可能性があることを指定する。

alf_luma_num_filters_signalled_minus_1に1を加えたものは、ルマ係数をシグナリングできる適応ループフィルタクラスの数指定する。alf_luma_num_filters_signalled_minus_1の値は、両端を含む0からNumAlfFilters-1の範囲であるものとする。 10

alf_luma_coeff_delta_idx[filtIdx]は、0からNumAlfFilters-1までの範囲のfiltIdxで示されるフィルタクラスのシグナリングされる適応ループフィルタのルマ係数デルタのインデックスを指定する。alf_luma_coeff_delta_idx[filtIdx]が存在しないとき、それは0に等しいと推定される。alf_luma_coeff_delta_idx[filtIdx]の長さは、Ceil(Log₂(alf_luma_num_filters_signalled_minus_1 + 1))ビットである。alf_luma_coeff_delta_idx[filtIdx]の値は、両端を含む0 ~ alf_luma_num_filters_signalled_minus_1の範囲であるものとする。

alf_luma_coeff_abs[sfIdx][j]は、sfIdxで示されるシグナリングされるルマフィルタのj番目の係数の絶対値を指定する。alf_luma_coeff_abs[sfIdx][j]が存在しないとき、それは0に等しいと推定される。alf_luma_coeff_abs[sfIdx][j]の値は、両端を含む0 ~ 128の範囲であるものとする。 20

alf_luma_coeff_sign[sfIdx][j]は、sfIdxが示すフィルタのj番目のルマ係数の符号を次のように指定する：

- alf_luma_coeff_sign[sfIdx][j]が0に等しい場合、対応するルマフィルタ係数は正の値を有する、

- それ以外の場合(alf_luma_coeff_sign[sfIdx][j]が1に等しい)、対応するルマフィルタ係数は負の値を有する。

alf_luma_coeff_sign[sfIdx][j]が存在しない場合、0に等しいと推定される。

sfIdx=0 . . alf_luma_num_filters_signalled_minus_1、j=0 . . 11の変数filtCoeff[sfIdx][j]は次のように初期化される： 30

【数21】

$$\text{filtCoeff}[\text{sfIdx}][j] = \text{alf_luma_coeff_abs}[\text{sfIdx}][j] * (1 - 2 * \text{alf_luma_coeff_sign}[\text{sfIdx}][j]) \quad (93)$$

要素AlfCoeffL[adaptation_parameter_set_id][filtIdx][j]、filtIdx=0..NumAlfFilters-1及びj=0..11のルマフィルタ係数AlfCoeffL[adaptation_parameter_set_id]は、以下のように導出される： 40

【数22】

$$\text{AlfCoeffL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j] = \text{filtCoeff}[\text{alf_luma_coeff_delta_idx}[\text{filtIdx}][j]] \quad (94)$$

固定フィルタ係数AlfFixFiltCoeff[i][j]、i=0..64、j=0..11、及びフィルタマッピングAlfClassToFiltMap[m][n]、m=0..15、n=0..24は、次のように導出される：

【数23】

AlfFixFiltCoeff =

(95)

- { 0, 0, 2, -3, 1, -4, 1, 7, -1, 1, -1, 5}
- { 0, 0, 0, 0, 0, -1, 0, 1, 0, 0, -1, 2}
- { 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0}
- { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 1}
- { 2, 2, -7, -3, 0, -5, 13, 22, 12, -3, -3, 17}
- { -1, 0, 6, -8, 1, -5, 1, 23, 0, 2, -5, 10}
- { 0, 0, -1, -1, 0, -1, 2, 1, 0, 0, -1, 4}
- { 0, 0, 3, -11, 1, 0, -1, 35, 5, 2, -9, 9}
- { 0, 0, 8, -8, -2, -7, 4, 4, 2, 1, -1, 25}
- { 0, 0, 1, -1, 0, -3, 1, 3, -1, 1, -1, 3}
- { 0, 0, 3, -3, 0, -6, 5, -1, 2, 1, -4, 21}
- { -7, 1, 5, 4, -3, 5, 11, 13, 12, -8, 11, 12}
- { -5, -3, 6, -2, -3, 8, 14, 15, 2, -7, 11, 16}
- { 2, -1, -6, -5, -2, -2, 20, 14, -4, 0, -3, 25}
- { 3, 1, -8, -4, 0, -8, 22, 5, -3, 2, -10, 29}
- { 2, 1, -7, -1, 2, -11, 23, -5, 0, 2, -10, 29}
- { -6, -3, 8, 9, -4, 8, 9, 7, 14, -2, 8, 9}
- { 2, 1, -4, -7, 0, -8, 17, 22, 1, -1, -4, 23}
- { 3, 0, -5, -7, 0, -7, 15, 18, -5, 0, -5, 27}
- { 2, 0, 0, -7, 1, -10, 13, 13, -4, 2, -7, 24}
- { 3, 3, -13, 4, -2, -5, 9, 21, 25, -2, -3, 12}
- { -5, -2, 7, -3, -7, 9, 8, 9, 16, -2, 15, 12}
- { 0, -1, 0, -7, -5, 4, 11, 11, 8, -6, 12, 21}
- { 3, -2, -3, -8, -4, -1, 16, 15, -2, -3, 3, 26}
- { 2, 1, -5, -4, -1, -8, 16, 4, -2, 1, -7, 33}
- { 2, 1, -4, -2, 1, -10, 17, -2, 0, 2, -11, 33}
- { 1, -2, 7, -15, -16, 10, 8, 8, 20, 11, 14, 11}
- { 2, 2, 3, -13, -13, 4, 8, 12, 2, -3, 16, 24}
- { 1, 4, 0, -7, -8, -4, 9, 9, -2, -2, 8, 29}
- { 1, 1, 2, -4, -1, -6, 6, 3, -1, -1, -3, 30}
- { -7, 3, 2, 10, -2, 3, 7, 11, 19, -7, 8, 10}
- { 0, -2, -5, -3, -2, 4, 20, 15, -1, -3, -1, 22}
- { 3, -1, -8, -4, -1, -4, 22, 8, -4, 2, -8, 28}
- { 0, 3, -14, 3, 0, 1, 19, 17, 8, -3, -7, 20}
- { 0, 2, -1, -8, 3, -6, 5, 21, 1, 1, -9, 13}
- { -4, -2, 8, 20, -2, 2, 3, 5, 21, 4, 6, 1}
- { 2, -2, -3, -9, -4, 2, 14, 16, 3, -6, 8, 24}
- { 2, 1, 5, -16, -7, 2, 3, 11, 15, -3, 11, 22}
- { 1, 2, 3, -11, -2, -5, 4, 8, 9, -3, -2, 26}
- { 0, -1, 10, -9, -1, -8, 2, 3, 4, 0, 0, 29}
- { 1, 2, 0, -5, 1, -9, 9, 3, 0, 1, -7, 20}

10

20

30

40

- {-2, 8, -6, -4, 3, -9, -8, 45, 14, 2, -13, 7}
 - {1, -1, 16, -19, -8, -4, -3, 2, 19, 0, 4, 30}
 - {1, 1, -3, 0, 2, -11, 15, -5, 1, 2, -9, 24}
 - {0, 1, -2, 0, 1, -4, 4, 0, 0, 1, -4, 7}
 - {0, 1, 2, -5, 1, -6, 4, 10, -2, 1, -4, 10}
 - {3, 0, -3, -6, -2, -6, 14, 8, -1, -1, -3, 31}
 - {0, 1, 0, -2, 1, -6, 5, 1, 0, 1, -5, 13}
 - {3, 1, 9, -19, -21, 9, 7, 6, 13, 5, 15, 21}
 - {2, 4, 3, -12, -13, 1, 7, 8, 3, 0, 12, 26}
 - {3, 1, -8, -2, 0, -6, 18, 2, -2, 3, -10, 23}
 - {1, 1, -4, -1, 1, -5, 8, 1, -1, 2, -5, 10}
 - {0, 1, -1, 0, 0, -2, 2, 0, 0, 1, -2, 3}
 - {1, 1, -2, -7, 1, -7, 14, 18, 0, 0, -7, 21}
 - {0, 1, 0, -2, 0, -7, 8, 1, -2, 0, -3, 24}
 - {0, 1, 1, -2, 2, -10, 10, 0, -2, 1, -7, 23}
 - {0, 2, 2, -11, 2, -4, -3, 39, 7, 1, -10, 9}
 - {1, 0, 13, -16, -5, -6, -1, 8, 6, 0, 6, 29}
 - {1, 3, 1, -6, -4, -7, 9, 6, -3, -2, 3, 33}
 - {4, 0, -17, -1, -1, 5, 26, 8, -2, 3, -15, 30}
 - {0, 1, -2, 0, 2, -8, 12, -6, 1, 1, -6, 16}
 - {0, 0, 0, -1, 1, -4, 4, 0, 0, 0, -3, 11}
 - {0, 1, 2, -8, 2, -6, 5, 15, 0, 2, -7, 9}
 - {1, -1, 12, -15, -7, -2, 3, 6, 6, -1, 7, 30}
- },

【数 2 4】

10

20

30

40

50

```

AlfClassToFiltMap = (96)
{
  { 8, 2, 2, 2, 3, 4, 53, 9, 9, 52, 4, 4, 5, 9, 2, 8, 10, 9, 1, 3, 39, 39,
    10, 9, 52 }
  { 11, 12, 13, 14, 15, 30, 11, 17, 18, 19, 16, 20, 20, 4, 53, 21, 22, 23, 14, 25, 26, 26, 27,
    28, 10 }
  { 16, 12, 31, 32, 14, 16, 30, 33, 53, 34, 35, 16, 20, 4, 7, 16, 21, 36, 18, 19, 21, 26, 37,
    38, 39 }
  { 35, 11, 13, 14, 43, 35, 16, 4, 34, 62, 35, 35, 30, 56, 7, 35, 21, 38, 24, 40, 16, 21, 48,
    57, 39 }
  { 11, 31, 32, 43, 44, 16, 4, 17, 34, 45, 30, 20, 20, 7, 5, 21, 22, 46, 40, 47, 26, 48, 63,
    58, 10 }
  { 12, 13, 50, 51, 52, 11, 17, 53, 45, 9, 30, 4, 53, 19, 0, 22, 23, 25, 43, 44, 37, 27, 28,
    10, 55 }
  { 30, 33, 62, 51, 44, 20, 41, 56, 34, 45, 20, 41, 41, 56, 5, 30, 56, 38, 40, 47, 11, 37, 42,
    57, 8 }
  { 35, 11, 23, 32, 14, 35, 20, 4, 17, 18, 21, 20, 20, 20, 4, 16, 21, 36, 46, 25, 41, 26, 48,
    49, 58 }
  { 12, 31, 59, 59, 3, 33, 33, 59, 59, 52, 4, 33, 17, 59, 55, 22, 36, 59, 59, 60, 22, 36, 59,
    25, 55 }
  { 31, 25, 15, 60, 60, 22, 17, 19, 55, 55, 20, 20, 53, 19, 55, 22, 46, 25, 43, 60, 37, 28, 10,
    55, 52 }
  { 12, 31, 32, 50, 51, 11, 33, 53, 19, 45, 16, 4, 4, 53, 5, 22, 36, 18, 25, 43, 26, 27, 27,
    28, 10 }
  { 5, 2, 44, 52, 3, 4, 53, 45, 9, 3, 4, 56, 5, 0, 2, 5, 10, 47, 52, 3, 63, 39,
    10, 9, 52 }
  { 12, 34, 44, 44, 3, 56, 56, 62, 45, 9, 56, 56, 7, 5, 0, 22, 38, 40, 47, 52, 48, 57, 39,
    10, 9 }
  { 35, 11, 23, 14, 51, 35, 20, 41, 56, 62, 16, 20, 41, 56, 7, 16, 21, 38, 24, 40, 26, 26, 42,
    57, 39 }
  { 33, 34, 51, 51, 52, 41, 41, 34, 62, 0, 41, 41, 56, 7, 5, 56, 38, 38, 40, 44, 37, 42, 57,
    39, 10 }
  { 16, 31, 32, 15, 60, 30, 4, 17, 19, 25, 22, 20, 4, 53, 19, 21, 22, 46, 25, 55, 26, 48, 63,
    58, 55 }
},

```

AlfCoeffL[adaptation_parameter_set_id][filtIdx][j]の値が両端を含む -2^7 から 2^7-1 までの範囲であることがビットストリーム準拠の要件であり、ここで、 $\text{filtIdx}=0 \dots \text{NumAlfFilters}-1$ 、 $j=0 \dots 11$ である。

$\text{alf_luma_clip_idx}[\text{sflIdx}][j]$ は、 sflIdx で示されるシグナリングされるルマフィルタのj番目の係数を乗算する前に使用するクリッピング値のクリッピングインデックスを指定する。 $\text{alf_luma_clip_idx}[\text{sflIdx}][j]$ の値が両端を含む0から3までの範囲であることがビットストリーム準拠の要件であり、ここで、 $\text{sflIdx}=0 \dots \text{alf_luma_num_filters_signalle_d_minus_1}$ 及び $j=0 \dots 11$ である。

ルマフィルタクリッピング値 $\text{AlfClipL}[\text{adaptation_parameter_set_id}]$ 、要素 $\text{AlfClipL}[\text{adaptation_parameter_set_id}][\text{filtIdx}][j]$ 、 $\text{filtIdx}=0 \dots \text{NumAlfFilters}-1$ 、及

40

50

び $j=0 \dots 11$ は BitDepth と、 $\text{alf_luma_clip_idx}[\text{alf_luma_coeff_delta_idx}[\text{filtIdx}][j]]$ に等しく設定された clipIdx に応じて、表 8 で指定されているように導出される。

$\text{alf_chroma_clip_flag}$ が 0 に等しいことは、線形適応ループフィルタリングがクロマ成分に適用されることを指定する。 $\text{alf_chroma_clip_flag}$ が 1 に等しいことは、クロマ成分に非線形適応ループフィルタリングが適用されることを指定する。存在しない場合、 $\text{alf_chroma_clip_flag}$ は、0 に等しいと推定される。

$\text{alf_chroma_num_alt_filters_minus 1}$ に 1 を加えたものは、クロマ成分の代替フィルタの数値を指定する。 $\text{alf_chroma_num_alt_filters_minus 1}$ の値は、両端を含む 0 ~ 7 の範囲であるとする。

$\text{alf_chroma_coeff_abs}[\text{altIdx}][j]$ は、インデックス altIdx の代替クロマフィルタの j 番目のクロマフィルタ係数の絶対値を指定する。 $\text{alf_chroma_coeff_abs}[\text{altIdx}][j]$ が存在しないとき、それは 0 に等しいと推定される。 $\text{alf_chroma_coeff_abs}[\text{sflIdx}][j]$ の値は、両端を含む 0 ~ 128 の範囲であるものとする。

10

$\text{alf_chroma_coeff_sign}[\text{altIdx}][j]$ は、インデックス altIdx の代替クロマフィルタの j 番目のクロマフィルタ係数の符号を以下のように指定する：

- $\text{alf_chroma_coeff_sign}[\text{altIdx}][j]$ が 0 に等しい場合、対応するクロマフィルタ係数は正の値を有する、

- ($\text{alf_chroma_coeff_sign}[\text{altIdx}][j]$ が 1)、対応するクロマフィルタ係数は負の値を有する。

$\text{alf_chroma_coeff_sign}[\text{altIdx}][j]$ が存在しないとき、それは 0 に等しいと推定される。

20

クロマフィルタ係数 $\text{AlfCoeff}_c[\text{adaptation_parameter_set_id}][\text{altIdx}]$ 、要素 $\text{AlfCoeff}_c[\text{adaptation_parameter_set_id}][\text{altIdx}][j]$ 、及び $\text{altIdx}=0 \dots \text{alf_chroma_num_alt_filters_minus 1}$ 、 $j=0 \dots 5$ は、以下のように導出される：

【数 2 5】

$$\text{AlfCoeff}_c[\text{adaptation_parameter_set_id}][\text{altIdx}][j] = \text{alf_chroma_coeff_abs}[\text{altIdx}][j] \cdot (1 - 2 * \text{alf_chroma_coeff_sign}[\text{altIdx}][j]) \quad *$$

(97)

30

$\text{AlfCoeff}_c[\text{adaptation_parameter_set_id}][\text{altIdx}][j]$ の値が両端を含む -2^7 から $2^7 - 1$ までの範囲であることがビットストリーム準拠の要件であり、ここで、 $\text{altIdx}=0 \dots \text{alf_chroma_num_alt_filters_minus 1}$ 、 $j=0 \dots 5$ である。

$\text{alf_cc_cb_filter_signal_flag}$ が 1 に等しいことは、Cb色成分の成分間フィルタがシグナリングされることを指定する。 $\text{alf_cc_cb_filter_signal_flag}$ が 0 に等しいことは、Cb色成分の成分間フィルタがシグナリングされないことを指定する。 ChromaArrayType が 0 に等しい場合、 $\text{alf_cc_cb_filter_signal_flag}$ は 0 に等しいものとする。

$\text{alf_cc_cb_filters_signalled_minus 1}$ に 1 を加えたものは、現在の ALF APS でシグナリングされる Cb 色成分の成分間フィルタの数を指定する。 $\text{alf_cc_cb_filters_signalled_minus 1}$ の値は、両端を含む 0 ~ 3 の範囲であるものとする。

40

$\text{alf_cc_cb_mapped_coeff_abs}[k][j]$ は、Cb色成分のシグナリングされる k 番目の成分間フィルタの j 番目のマッピングされた係数の絶対値を指定する。 $\text{alf_cc_cb_mapped_coeff_abs}[k][j]$ が存在しないとき、それは 0 に等しいと推定される。

$\text{alf_cc_cb_coeff_sign}[k][j]$ は、Cb色成分のシグナリングされる k 番目の成分間フィルタの j 番目の係数の符号を以下の通り指定する：

- $\text{alf_cc_cb_coeff_sign}[k][j]$ が 0 に等しい場合、対応する成分間フィルタ係数は正の値を有する、

- それ以外の場合 ($\text{alf_cc_cb_sign}[k][j]$ が 1 に等しい)、対応する成分間フィルタ係数は負の値を有する。

$\text{alf_cc_cb_coeff_sign}[k][j]$ が存在しないとき、それは 0 に等しいと推定される。

50

Cb色成分CcAlfApsCoeffCb[adaptation_parameter_set_id][k][j]、 $j=0 \dots 6$ のシグナリングされるk番目の成分間フィルタ係数は次のように導出される：

- alf_cc_cb_mapped_coeff_abs[k][j]が0に等しい場合、CcAlfApsCoeffCb[adaptation_parameter_set_id][k][j]は0に等しく設定される。

- それ以外の場合、CcAlfApsCoeffCb[adaptation_parameter_set_id][k][j]は、 $(1 - 2 * \text{alf_cc_cb_coeff_sign}[k][j]) * 2^{\text{alf_cc_cb_mapped_coeff_abs}[k][j] - 1}$ に等しく設定される。

alf_cc_cr_filter_signal_flagが1に等しいことは、Cr色成分の成分間フィルタがシグナルされることを指定する。alf_cc_cr_filter_signal_flagが0に等しいことは、Cr色成分の成分間フィルタがシグナルされないことを指定する。ChromaArrayTypeが0に等しい場合、alf_cc_cr_filter_signal_flagは0に等しいものとする。 10

alf_cc_cr_filters_signalled_minus_1に1を加えたものは、現在のALF APSでシグナリングされるCr色成分の成分間フィルタの数を指定する。alf_cc_cr_filters_signalled_minus_1の値は、両端を含む0～3の範囲であるものとする。

alf_cc_cr_mappedcoeff_abs[k][j]は、Cr色成分のシグナリングされるk番目の成分間フィルタのj番目のマッピングされた係数の絶対値を指定する。alf_cc_cr_mappedcoeff_abs[k][j]が存在しないとき、それは0に等しいと推定される。

alf_cc_cr_coeff_sign[k][j]は、Cr色成分のシグナリングされるk番目の成分間フィルタのj番目の係数の符号を以下の通り指定する：

- alf_cc_cr_coeff_sign[k][j]が0に等しい場合、対応する成分間フィルタ係数は正の値を有する、 20

- それ以外の場合(alf_cc_cr_sign[k][j]が1に等しい)、対応する成分間フィルタ係数は負の値を有する。

alf_cc_cr_coeff_sign[k][j]が存在しないとき、それは0に等しいと推定される。

Cr色成分CcAlfApsCoeffCr[adaptation_parameter_set_id][k][j]、 $j=0 \dots 6$ のシグナリングされるk番目の成分間フィルタ係数は次のように導出される：

- alf_cc_cr_mapped_coeff_abs[k][j]が0に等しい場合、CcAlfApsCoeffCr[adaptation_parameter_set_id][k][j]は0に等しく設定される。

- それ以外の場合、CcAlfApsCoeffCr[adaptation_parameter_set_id][k][j]は、 $(1 - 2 * \text{alf_cc_cr_coeff_sign}[k][j]) * 2^{\text{alf_cc_cr_mapped_coeff_abs}[k][j] - 1}$ に等しく設定される。 30

alf_chroma_clip_idx[altIdx][j]は、インデックスaltIdxを有する代替クロマフィルタのj番目の係数により乗算する前に使用するクリッピング値のクリッピングインデックスを指定する。alf_chroma_clip_idx[altIdx][j]の値が両端を含む0から3までの範囲であることがビットストリーム準拠の要件であり、ここで、altIdx=0..alf_chroma_num_alt_filters_minus_1、 $j=0 \dots 5$ である。

クロマフィルタクリッピング値AlfClipc[adaptation_parameter_set_id][altIdx]、要素AlfClipc[adaptation_parameter_set_id][altIdx][j]、altIdx=0..alf_chroma_num_alt_filters_minus_1、 $j=0 \dots 5$ は、BitDepthと、alf_chroma_clip_idx[altIdx][j]に等しく設定されたclipIdxに応じて、表8で指定されているように導出される。 40

【表 9】

表 8 - BitDepth 及び clipIdx に依存する標準 AlfClip

BitDepth	clipIdx			
	0	1	2	3
8	2^8	2^5	2^3	2^1
9	2^9	2^6	2^4	2^2
10	2^{10}	2^7	2^5	2^3
11	2^{11}	2^8	2^6	2^4
12	2^{12}	2^9	2^7	2^5
13	2^{13}	2^{10}	2^8	2^6
14	2^{14}	2^{11}	2^9	2^7
15	2^{15}	2^{12}	2^{10}	2^8
16	2^{16}	2^{13}	2^{11}	2^9

10

20

Imcs_min_bin_idxは、クロマスケーリングによるルママッピング構成処理において使用される最小ピンインデックスを指定する。Imcs_min_bin_idxの値は、両端を含む 0 ~ 15 の範囲であるものとする。

Imcs_delta_max_bin_idxは、15 と、クロマスケーリングによるルママッピング構成処理において使用される最大ピンインデックスLmcsMaxBinIdxとの間のデルタ値を指定する。Imcs_delta_max_bin_idxの値は、両端を含む 0 ~ 15 の範囲であるものとする。LmcsMaxBinIdxの値は、15 - Imcs_delta_max_bin_idxに等しく設定される。LmcsMaxBinIdxの値は、Imcs_min_bin_idx以上であるものとする。

30

Imcs_delta_cw_prec_minus 1 に 1 を加えたものは、シンタックスImcs_delta_abs_cw[i]の表現のために使用されるビットの数を指定する。Imcs_delta_cw_prec_minus 1 の値は、両端を含む 0 ~ BitDepth- 2 の範囲であるものとする。

Imcs_delta_abs_cw[i]は、i番目のピンの絶対デルタコードワード値を指定する。

Imcs_delta_sign_cw_flag[i]は、以下のように変数ImcsDeltaCW[i]の符号を指定する。

Imcs_delta_sign_cw_flag[i]が 0 に等しい場合、ImcsDeltaCW[i]は正の値である。

- それ以外の場合 (Imcs_delta_sign_cw_flag[i]が 0 に等しくない)、ImcsDeltaCW[i]は負の値である。

Imcs_delta_sign_cw_flag[i]が存在しないとき、それは 0 に等しいと推定される。

40

変数OrgCWは、以下のように導出される：

【数 2 6】

$$\text{OrgCW} = (1 \ll \text{BitDepth}) / 16 \quad (98)$$

変数ImcsDeltaCW[i]、 $i = \text{Imcs_min_bin_idx} \dots \text{LmcsMaxBinIdx}$ は以下のように導出される：

【数 2 7】

$$\text{ImcsDeltaCW}[i] = (1 - 2 * \text{Imcs_delta_sign_cw_flag}[i]) * \text{Imcs_delta_abs_cw}[i] \quad (99)$$

50

変数 $lmcsCW[i]$ は以下のように導出される :

$i = 0 \dots lmcs_min_bin_idx - 1$ について、 $lmcsCW[i]$ は 0 に等しく設定される。

$i = lmcs_min_bin_idx \dots LmcsMaxBinIdx$ について、以下が適用される :

【数 2 8】

$$lmcsCW[i] = OrgCW + lmcsDeltaCW[i] \quad (100)$$

$lmcsCW[i]$ の値は)、両端を含む ($OrgCW - 3$) から ($OrgCW - 3 - 1$) の範囲であるものとする。

$i = LmcsMaxBinIdx + 1 \dots 15$ について、 $lmcsCW[i]$ は 0 に等しく設定される。

以下の条件が真であることは、ビットストリーム準拠の要件である :

【数 2 9】

$$\sum_{i=0}^{15} lmcsCW[i] \leq (1 \ll BitDepth) - 1 \quad (101)$$

変数 $InputPivot[i]$ 、 $i = 0 \dots 16$ は以下のように導出される :

【数 3 0】

$$InputPivot[i] = i * OrgCW \quad (102)$$

変数 $LmcsPivot[i]$ 、 $i = 0 \dots 16$ 、並びに、変数 $ScaleCoeff[i]$ 及び $InvScaleCoeff[i]$ 、 $i = 0 \dots 15$ は、以下のように導出される :

【数 3 1】

```

LmcsPivot[0] = 0;
for(i = 0; i <= 15; i++) {
  LmcsPivot[i + 1] = LmcsPivot[i] + lmcsCW[i]
  ScaleCoeff[i] = (lmcsCW[i] * (1 << 11) + (1 << (Log2(OrgCW) - 1))) >>
(Log2(OrgCW))
  if(lmcsCW[i] == 0)
    InvScaleCoeff[i] = 0
  else
    InvScaleCoeff[i] = OrgCW * (1 << 11) / lmcsCW[i]
}

```

$i = lmcs_min_bin_idx \dots LmcsMaxBinIdx$ について、 $LmcsPivot[i]$ の値が、 $1 \ll (BitDepth - 5)$ の倍数ではないとき、 $(LmcsPivot[i] \ll (BitDepth - 5))$ の値が $(LmcsPivot[i + 1] \ll (BitDepth - 5))$ の値と等しくないことが、ビットストリーム準拠の要件である。

$lmcs_delta_abs_crs$ は変数 $lmcsDeltaCrs$ の絶対コードワード値を指定する。 $lmcs_delta_abs_crs$ の値は、両端を含む 0 ~ 7 の範囲であるとする。存在しない場合、 $lmcs_delta_abs_crs$ は、0 に等しいと推定される。

$lmcs_delta_sign_crs_flag$ は、変数 $lmcsDeltaCrs$ の符号を指定する。存在しない場合、 $lmcs_delta_sign_crs_flag$ は、0 に等しいと推定される。

変数 $lmcsDeltaCrs$ は、以下のように導出される :

【数 3 2】

10

20

30

40

50

$$\text{lmcsDeltaCrs} = (1 - 2 * \text{lmcs_delta_sign_crs_flag}) * \text{lmcs_delta_abs_crs} \quad (104)$$

lmcsCW[i]が0に等しくないとき、(lmcsCW[i] + lmcsDeltaCrs)が両端を含む(OrgCW - 3)から((OrgCW - 3) - 1)の範囲にあることが、ビットストリーム準拠の要件である。

変数ChromaScaleCoeff[i]、i=0...15は以下のように導出される：

【数33】

```

if( lmcsCW[ i ] == 0 )
    ChromaScaleCoeff[ i ] = ( 1 << 11 )
else
    ChromaScaleCoeff[ i ] = OrgCW * ( 1 << 11 ) / ( lmcsCW[ i ] + lmcsDeltaCrs )

```

scaling_matrix_for_lfnst_disabled_flagが1に等しいことは、スケーリングマトリクスがLFNSTでコーディングされたブロックに適用されないことを指定する。scaling_matrix_for_lfnst_disabled_flagが0に等しいことは、スケーリングマトリクスがLFNSTでコーディングされたブロックに適用される可能性があることを指定する。

scaling_list_chroma_present_flagが1に等しいことは、クロマスケールリストがscaling_list_data()に存在することを指定する。scaling_list_chroma_present_flagが0に等しいことは、クロマスケールリストがscaling_list_data()に存在しないことを指定する。scaling_list_chroma_present_flagが、ChromaArrayTypeが0に等しい場合は0に等しく、ChromaArrayTypeが0に等しくない場合は1に等しいことが、ビットストリーム準拠の要件である。

scaling_list_copy_mode_flag[id]が1に等しいことは、スケーリングリストの値が参照スケーリングリストの値と同じであることを指定する。参照スケーリングリストは、scaling_list_pred_id_delta[id]で指定される。scaling_list_copy_mode_flag[id]が0に等しいことは、scaling_list_pred_mode_flagが存在することを指定する。

scaling_list_copy_mode_flag[id]が1に等しいことは、スケーリングリストの値が参照スケーリングリストから予測できることを指定する。参照スケーリングリストは、scaling_list_pred_id_delta[id]で指定される。scaling_list_copy_mode_flag[id]が0に等しいことは、スケーリングリストの値が明示的にシグナリングされることを指定する。存在しないとき、scaling_list_pred_mode_flag[id]の値は0に等しいと推定される。

scaling_list_pred_id_delta[id]は、予測スケーリングマトリクスScalingMatrixPred[id]を導出するために使用される参照スケーリングリストを指定する。存在しないとき、scaling_list_pred_id_delta[id]の値は0に等しいと推定される。scaling_list_pred_id_delta[id]の値は0からmaxIdDeltaの範囲とし、idに応じて次のようにmaxIdDeltaを導出する：

【数34】

$$\text{maxIdDelta} = (\text{id} < 2) ? \text{id} : ((\text{id} < 8) ? (\text{id} - 2) : (\text{id} - 8)) \quad (106)$$

変数refId及びmatrixSizeは、以下のように導出される：

【数35】

10

20

30

40

50

$$\text{refId} = \text{id} - \text{scaling_list_pred_id_delta}[\text{id}] \quad (107)$$

$$\text{matrixSize} = (\text{id} < 2) ? 2 : ((\text{id} < 8) ? 4 : 8) \quad (108)$$

(matrixSize) x (matrixSize) アレイであるScalingMatrixPred[x][y]、x=0 . . matrixSize-1、y=0 . . matrixSize-1、及び変数ScalingMatrixDCPredは、以下のよう
に導出される：

- scaling_list_copy_mode_flag[id]とscaling_list_pred_mode_flag[id]の両方が 10
0 の場合、ScalingMatrixPredのすべての要素は 8 に等しく設定され、ScalingMatrixDCPred
の値は 8 に等しく設定される、

- それ以外の場合、scaling_list_pred_id_delta[id]が 0 に等しい場合、ScalingMatrixPred
の全ての要素は 16 に等しく設定され、ScalingMatrixDCPredは 16 に等しく設定
される、

- それ以外の場合 (scaling_list_copy_mode_flag[id]又はscaling_list_pred_mode_flag[id]が 1
に等しく、scaling_list_pred_id_delta[id]が 0 より大きい場合)、ScalingMatrixPredはScalingMatrixRec[refId]と等しく設定され、以下がScalingMatrixDCPredに適用される：

- refIdが 13 より大きい場合、ScalingMatrixDCPredはScalingMatrixDCRec[refId-14]と等しく設定される、 20

- その他の場合 (refIdが 13 以下)、ScalingMatrixDCPredはScalingMatrixPred[0][0]と等しく設定される。

scaling_list_dc_coef[id-14]は、idが 13 より大きい場合、変数ScalingMatrixDC[id-14]の値を次のように導出するために使用される：

【数 3 6】

$$\text{ScalingMatrixDCRec}[\text{id} - 14] = (\text{ScalingMatrixDCPred} + \text{scaling_list_dc_coef}[\text{id} - 14]) \& 255 \quad (109)$$

存在しないとき、scaling_list_dc_coef[id-14]の値は 0 に等しいと推定される。scaling_list_dc_coef[id-14]の値は、両端を含む -128 ~ 127 の範囲であるものとする。ScalingMatrixDCRec[id-14]の値は 0 より大きいものとする。 30

scaling_list_delta_coef[id][i]は、scaling_list_copy_mode_flag[id]が 0 に等しい場合、現在のマトリクス係数ScalingList[id][i]と以前のマトリクス係数ScalingList[id][i-1]との間の差を指定する。scaling_list_delta_coef[id][i]の値は、両端を含む -128 ~ 127 の範囲であるものとする。scaling_list_copy_mode_flag[id]が 1 に等しい場合、ScalingList[id]のすべての要素は 0 に等しく設定される。

(matrixSize) x (matrixSize) アレイであるScalingMatrixRec[id]は次のように導出される： 40

【数 3 7】

$$\text{ScalingMatrixRec}[\text{id}][\text{x}][\text{y}] = (\text{ScalingMatrixPred}[\text{x}][\text{y}] + \text{ScalingList}[\text{id}][\text{k}]) \& 255 \quad (110)$$

with k = 0..(matrixSize * matrixSize - 1),

x = DiagScanOrder[Log2(matrixSize)][Log2(matrixSize)][k][0],

and

y = DiagScanOrder[Log2(matrixSize)][Log2(matrixSize)][k][1]

ScalingMatrixRec[id][x][y]の値は0より大きいものとする。

【 0 0 4 0 】

3.5 PHのシンタックスとセマンティクス

最新のVVCドラフトテキストでは、PHシンタックスとセマンティクスは次の通りである。

【表 1 0】

picture_header_rbsp() {	Descriptor
picture_header_structure()	
rbsp_trailing_bits()	
}	

10

PH RBSPはPHシンタックス構造、すなわちpicture_header_structure()を含む。

20

30

40

50

【表 1 1】

picture_header_structure() {	Descriptor	
gdr_or_irap_pic_flag	u(1)	
if(gdr_or_irap_pic_flag)		
gdr_pic_flag	u(1)	
ph_inter_slice_allowed_flag	u(1)	
if(ph_inter_slice_allowed_flag)		
ph_intra_slice_allowed_flag	u(1)	10
non_reference_picture_flag	u(1)	
ph_pic_parameter_set_id	ue(v)	
ph_pic_order_cnt_lsb	u(v)	
if(gdr_or_irap_pic_flag)		
no_output_of_prior_pics_flag	u(1)	
if(gdr_pic_flag)		
recovery_poc_cnt	ue(v)	
for(i = 0; i < NumExtraPhBits; i++)		
ph_extra_bit[i]	u(1)	20
if(sps_poc_msb_flag) {		
ph_poc_msb_present_flag	u(1)	
if(ph_poc_msb_present_flag)		
poc_msb_val	u(v)	
}		
if(sps_alf_enabled_flag && alf_info_in_ph_flag) {		
ph_alf_enabled_flag	u(1)	
if(ph_alf_enabled_flag) {		
ph_num_alf_aps_ids_luma	u(3)	30
for(i = 0; i < ph_num_alf_aps_ids_luma; i++)		
ph_alf_aps_id_luma[i]	u(3)	
if(ChromaArrayType != 0)		
ph_alf_chroma_idc	u(2)	
if(ph_alf_chroma_idc > 0)		
ph_alf_aps_id_chroma	u(3)	
if(sps_ccalf_enabled_flag) {		
ph_cc_alf_cb_enabled_flag	u(1)	
if(ph_cc_alf_cb_enabled_flag)		
ph_cc_alf_cb_aps_id	u(3)	40

ph_cc_alf_cr_enabled_flag	u(1)	
if(ph_cc_alf_cr_enabled_flag)		
ph_cc_alf_cr_aps_id	u(3)	
}		
}		
}		
if(sps_lmcs_enabled_flag) {		
ph_lmcs_enabled_flag	u(1)	10
if(ph_lmcs_enabled_flag) {		
ph_lmcs_aps_id	u(2)	
if(ChromaArrayType != 0)		
ph_chroma_residual_scale_flag	u(1)	
}		
}		
if(sps_scaling_list_enabled_flag) {		
ph_scaling_list_present_flag	u(1)	
if(ph_scaling_list_present_flag)		20
ph_scaling_list_aps_id	u(3)	
}		
if(sps_virtual_boundaries_enabled_flag && !sps_virtual_boundaries_present_flag) {		
ph_virtual_boundaries_present_flag	u(1)	
if(ph_virtual_boundaries_present_flag) {		
ph_num_ver_virtual_boundaries	u(2)	
for(i = 0; i < ph_num_ver_virtual_boundaries; i++)		
ph_virtual_boundaries_pos_x[i]	u(13)	30
ph_num_hor_virtual_boundaries	u(2)	
for(i = 0; i < ph_num_hor_virtual_boundaries; i++)		
ph_virtual_boundaries_pos_y[i]	u(13)	
}		
}		
if(output_flag_present_flag)		
pic_output_flag	u(1)	
if(rpl_info_in_ph_flag)		
ref_pic_lists()		
if(partition_constraints_override_enabled_flag)		40

partition_constraints_override_flag	u(1)	
if(ph_intra_slice_allowed_flag) {		
if(partition_constraints_override_flag) {		
ph_log2_diff_min_qt_min_cb_intra_slice_luma	ue(v)	
ph_max_mtt_hierarchy_depth_intra_slice_luma	ue(v)	
if(ph_max_mtt_hierarchy_depth_intra_slice_luma != 0) {		
ph_log2_diff_max_bt_min_qt_intra_slice_luma	ue(v)	
ph_log2_diff_max_tt_min_qt_intra_slice_luma	ue(v)	10
}		
if(qtbtt_dual_tree_intra_flag) {		
ph_log2_diff_min_qt_min_cb_intra_slice_chroma	ue(v)	
ph_max_mtt_hierarchy_depth_intra_slice_chroma	ue(v)	
if(ph_max_mtt_hierarchy_depth_intra_slice_chroma != 0) {		
ph_log2_diff_max_bt_min_qt_intra_slice_chroma	ue(v)	
ph_log2_diff_max_tt_min_qt_intra_slice_chroma	ue(v)	
}		
}		20
}		
if(cu_qp_delta_enabled_flag)		
ph_cu_qp_delta_subdiv_intra_slice	ue(v)	
if(pps_cu_chroma_qp_offset_list_enabled_flag)		
ph_cu_chroma_qp_offset_subdiv_intra_slice	ue(v)	
}		
if(ph_inter_slice_allowed_flag) {		
if(partition_constraints_override_flag) {		
ph_log2_diff_min_qt_min_cb_inter_slice	ue(v)	
ph_max_mtt_hierarchy_depth_inter_slice	ue(v)	30
if(ph_max_mtt_hierarchy_depth_inter_slice != 0) {		
ph_log2_diff_max_bt_min_qt_inter_slice	ue(v)	
ph_log2_diff_max_tt_min_qt_inter_slice	ue(v)	
}		
}		
if(cu_qp_delta_enabled_flag)		
ph_cu_qp_delta_subdiv_inter_slice	ue(v)	
if(pps_cu_chroma_qp_offset_list_enabled_flag)		
ph_cu_chroma_qp_offset_subdiv_inter_slice	ue(v)	40

10

20

30

40

50

if(sps_temporal_mvp_enabled_flag) {	
ph_temporal_mvp_enabled_flag	u(1)
if(ph_temporal_mvp_enabled_flag && rpl_info_in_ph_flag) {	
ph_collocated_from_l0_flag	u(1)
if((ph_collocated_from_l0_flag && num_ref_entries[0][RplsIdx[0]] > 1) (!ph_collocated_from_l0_flag && num_ref_entries[1][RplsIdx[1]] > 1))	
ph_collocated_ref_idx	ue(v)
}	
}	
mvd_l1_zero_flag	u(1)
if(sps_fpel_mmvd_enabled_flag)	
ph_fpel_mmvd_enabled_flag	u(1)
if(sps_bdof_pic_present_flag)	
ph_disable_bdof_flag	u(1)
if(sps_dmvr_pic_present_flag)	
ph_disable_dmvr_flag	u(1)
if(sps_prof_pic_present_flag)	
ph_disable_prof_flag	u(1)
if((pps_weighted_pred_flag pps_weighted_bipred_flag) && wp_info_in_ph_flag)	
pred_weight_table()	
}	
if(qp_delta_info_in_ph_flag)	
ph_qp_delta	se(v)
if(sps_joint_cbr_enabled_flag)	
ph_joint_cbr_sign_flag	u(1)
if(sps_sao_enabled_flag && sao_info_in_ph_flag) {	
ph_sao_luma_enabled_flag	u(1)
if(ChromaArrayType != 0)	
ph_sao_chroma_enabled_flag	u(1)
}	
if(sps_dep_quant_enabled_flag)	
ph_dep_quant_enabled_flag	u(1)
if(sps_sign_data_hiding_enabled_flag && !ph_dep_quant_enabled_flag)	

10

20

30

40

50

pic_sign_data_hiding_enabled_flag	u(1)	
if(deblocking_filter_override_enabled_flag && dbf_info_in_ph_flag) {		
ph_deblocking_filter_override_flag	u(1)	
if(ph_deblocking_filter_override_flag) {		
ph_deblocking_filter_disabled_flag	u(1)	
if(!ph_deblocking_filter_disabled_flag) {		10
ph_beta_offset_div2	se(v)	
ph_tc_offset_div2	se(v)	
ph_cb_beta_offset_div2	se(v)	
ph_cb_tc_offset_div2	se(v)	
ph_cr_beta_offset_div2	se(v)	
ph_cr_tc_offset_div2	se(v)	
}		20
}		
}		
if(picture_header_extension_present_flag) {		
ph_extension_length	ue(v)	
for(i = 0; i < ph_extension_length; i++)		
ph_extension_data_byte[i]	u(8)	
}		30
}		

PHシンタックス構造は、PHシンタックス構造に関連付けられたコーディングピクチャのすべてのスライスに共通する情報を含む。

gdr_or_irap_pic_flagが1に等しいことは、現在ピクチャがGDR又はIRAPピクチャであることを指定する。gdr_or_irap_pic_flagが0に等しいことは、現在ピクチャがGDR又はIRAPピクチャであってもよく又はそうではなくてもよいことを指定する。

gdr_pic_flagが1に等しいことは、PHに関連付けられたピクチャがGDRピクチャであることを指定する。gdr_pic_flagが0に等しいことは、PHに関連付けられたピクチャがGDRピクチャではないことを指定する。存在しないとき、gdr_pic_flagの値は0に等しいと推定される。gdr_enabled_flagが0に等しい場合、gdr_pic_flagの値は0に等しくなるものとする。

ph_inter_slice_allowed_flagが0に等しいことは、ピクチャのすべてのコーディングスライスのslice_typeが2に等しいことを指定する。ph_inter_slice_allowed_flagが1に等しいことは、slice_typeが0又は1に等しい1つ以上のコーディングスライスがピクチャ内に存在するかどうかを指定する。

ph_intra_slice_allowed_flagが0に等しいことは、ピクチャのすべてのコーディングスライスのslice_typeが0又は1に等しいことを指定する。ph_intra_slice_allowed_flagが1に等しいことは、slice_typeが2に等しい1つ以上のコーディングスライスがピク

10

20

30

40

50

チャ内に存在するかどうかを指定する。存在しないとき、`ph_intra_slice_allowed_flag`の値は1に等しいと推定される。

注1：PH NALユニットを変更することなくサブピクチャに基づくビットストリームマージを行うことが想定されているビットストリームの場合、エンコーダは`ph_inter_slice_allowed_flag`と`ph_intra_slice_allowed_flag`の両方の値を1に等しく設定することが期待される。

`non_reference_picture_flag`が1に等しいことは、PHに関連付けられたピクチャが参照ピクチャとして使用されないことを指定する。`non_reference_picture_flag`が0に等しいことは、PHに関連付けられたピクチャが参照ピクチャとして使用される、又は使用されないことを指定する。

`ph_pic_parameter_set_id`は、使用中のPPSの`pps_pic_parameter_set_id`の値を指定する。`ph_pic_parameter_set_id`の値は、両端を含む0～63の範囲であるものとする。

PHのTemporalIdの値が、`ph_pic_parameter_set_id`に等しい`pps_pic_parameter_set_id`を持つPPSのTemporalIdの値以上であることはビットストリーム適合性の要件である。

`ph_pic_order_cnt_lsb`は、現在ピクチャについて、`MaxPicOrderCntLsb`を法とするピクチャオーダカウントを指定する。`ph_pic_order_cnt_lsb`シンタックス要素の長さは、 $\log_2 \text{max_pic_order_cnt_lsb_minus} 4 + 4$ ビットである。`ph_pic_order_cnt_lsb`の値は、両端を含む0から`MaxPicOrderCntLsb - 1`の範囲であるものとする。

`no_output_of_prior_pics_flag`は、AnnexCで指定されたビットストリームの第1ピクチャではないCLVSSピクチャの復号後のDPB内の以前に復号されたピクチャの出力に影響を与える。

`recovery_poc_cnt`は、復号ピクチャの復元ポイントを出力順序で指定する。現在ピクチャが、PHに関連付けられたGDRピクチャであり、CLVSで現在のGDRピクチャに続き、現在のGDRピクチャの`PicOrderCntVal`に`recover_poc_cnt`の値を加えたものに等しい`PicOrderCntVal`を有するピクチャ`picA`がある場合、ピクチャ`picA`は復元ポイントピクチャと呼ばれる。その他の場合、現在ピクチャの`PicOrderCntVal`に`recover_poc_cnt`の値を加えたよりも大きい`PicOrderCntVal`を有する出力順で第1ピクチャが、復元ポイントピクチャと呼ばれる。復元ポイントピクチャは、復号順で現在のGDRピクチャに先行してはならない。`recovery_poc_cnt`の値は、両端を含む、0から`MaxPicOrderCntLsb - 1`までの範囲であるものとする。

現在ピクチャがGDRピクチャである場合、変数`RpPicOrderCntVal`は次のように導出される：

【数38】

$$\text{RpPicOrderCntVal} = \text{PicOrderCntVal} + \text{recovery_poc_cnt} \quad (82)$$

注2：`gdr_enabled_flag`が1に等しく、現在ピクチャの`PicOrderCntVal`が関連するGDRピクチャの`RpPicOrderCntVal`以上の場合、出力順で現在の及び後続の復号されたピクチャは、存在する場合には前のIRAPピクチャから復号処理を開始して生成された対応するピクチャと完全に一致する。

`ph_extra_bit[i]`は1又は0に等しくてよい。この仕様のこのバージョンに準拠するデコーダは、`ph_extra_bit[i]`の値を無視するものとする。その値は、この仕様のこのバージョン内で指定されるプロファイルへのデコーダ準拠性に影響しない。

`ph_poc_msb_present_flag`が1に等しいことは、シンタックス要素`poc_msb_val`がPHに存在することを指定する。`ph_poc_msb_present_flag`が0に等しいことは、シンタックス要素`poc_msb_val`がPHに存在しないことを指定する。`vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]`が0に等しく、現在レイヤの参照レイヤの現在A

10

20

30

40

50

Uにピクチャが存在する場合、ph_poc_msb_present_flagの値は0に等しくなるものとする。

poc_msb_valは現在ピクチャのPOC MSB値を指定する。シンタックス要素poc_msb_valの長さは、poc_msb_len_minus 1 + 1ビットである。

ph_alf_enabled_flagが1に等しいことは、PHに関連付けられたすべてのスライスに対して適応ループフィルタが有効であり、スライス内のY、Cb、又はCr色成分に適用できることを指定する。ph_alf_enabled_flagが0に等しいことは、PHに関連付けられた1つ以上又はすべてのスライスに対して適応ループフィルタを無効にできることを指定する。存在しない場合、ph_alf_enabled_flagは、0に等しいと推定される。

ph_num_alf_aps_ids_lumaは、PHに関連付けられたスライスが参照するALF APSの数を指定する。

ph_alf_aps_id_luma[i]は、PHに関連付けられたスライスのルマ成分が参照するi番目のALF APSのadaptation_parameter_set_idを指定する。

aps_params_typeがALF_APSに等しく、adaptation_parameter_set_idがph_alf_aps_id_luma[i]に等しいAPS NALユニットのalf_luma_filter_signal_flagの値は1に等しいものとする。

aps_params_typeがALF_APSと等しく、adaptation_parameter_set_idがph_alf_aps_id_luma[i]と等しいAPS NALユニットのTemporalIdは、PHに関連付けられたピクチャのTemporalId以下であるものとする。

ph_alf_chroma_idcが0に等しいことは、CbとCr色成分に適応ループフィルタが適用されないことを指定する。ph_alf_chroma_idcが1に等しいことは、Cb色成分に適応ループフィルタが適用されることを示す。ph_alf_chroma_idcが2に等しいことは、Cr色成分に適応ループフィルタが適用されることを示す。ph_alf_chroma_idcが3に等しいことは、Cb及びCr色成分に適応ループフィルタが適用されることを示す。ph_alf_chroma_idcが存在しない場合は、0に等しいと推定される。

ph_alf_aps_id_chromaは、PHに関連付けられたスライスのクロマ成分が参照するi番目のALF APSのadaptation_parameter_set_idを指定する。

aps_params_typeがALF_APSに等しく、adaptation_parameter_set_idがph_alf_aps_id_chromaに等しいAPS NALユニットのalf_chroma_filter_signal_flagの値は1に等しいものとする。

aps_params_typeがALF_APSと等しく、adaptation_parameter_set_idがph_alf_aps_id_chromaと等しいAPS NALユニットのTemporalIdは、PHに関連付けられたピクチャのTemporalId以下であるものとする。

ph_cc_alf_cb_enabled_flagが1に等しいことは、PHに関連付けられたすべてのスライスに対してCb色成分の成分間フィルタが有効になり、スライス内のCb色成分に適用できることを指定する。ph_cc_alf_cb_enabled_flagが0に等しいことは、PHに関連付けられた1つ以上又はすべてのスライスに対して、Cb色成分の成分間フィルタを無効にすることができることを指定する。存在しない場合、ph_cc_alf_cb_enabled_flagは、0に等しいと推定される。

ph_cc_alf_cb_aps_idは、PHに関連付けられたスライスのCb色成分が参照するALF APSのadaptation_parameter_set_idを指定する。

aps_params_typeがALF_APSに等しく、adaptation_parameter_set_idがph_cc_alf_cb_aps_idに等しいAPS NALユニットのalf_cc_cb_filter_signal_flagの値は1に等しいものとする。

aps_params_typeがALF_APSと等しく、adaptation_parameter_set_idがph_cc_alf_cb_aps_idと等しいAPS NALユニットのTemporalIdは、PHに関連付けられたピクチャのTemporalId以下であるものとする。

ph_cc_alf_cr_enabled_flagが1に等しいことは、PHに関連付けられたすべてのスライスに対してCr色成分の成分間フィルタが有効になり、スライス内のCr色成分に適用できることを指定する。ph_cc_alf_cr_enabled_flagが0に等しいことは、PHに関連付けら

10

20

30

40

50

れた1つ以上又はすべてのスライスに対して、Cr色成分の成分間フィルタを無効にすることができることを指定する。存在しない場合、ph_cc_alf_cr_enabled_flagは、0に等しいと推定される。

ph_cc_alf_cr_aps_idは、PHに関連付けられたスライスのCr色成分が参照するALF APSのadaptation_parameter_set_idを指定する。

aps_params_typeがALF_APSに等しく、adaptation_parameter_set_idがph_cc_alf_cr_aps_idに等しいAPS NALユニットのalf_cc_cr_filter_signal_flagの値は1に等しいものとする。

aps_params_typeがALF_APSと等しく、adaptation_parameter_set_idがph_cc_alf_cr_aps_idと等しいAPS NALユニットのTemporalIdは、PHに関連付けられたピクチャのTemporalId以下であるものとする。

10

ph_lmcs_enabled_flagが1に等しいことは、PHに関連付けられているすべてのスライスについて、クロマスケーリングによるルママッピングが有効であることを指定する。ph_lmcs_enabled_flagが0に等しいことは、PHに関連付けられた1つ以上又はすべてのスライスに対して、クロマスケーリングによるルママッピングを無効にできることを指定する。存在しないとき、ph_lmcs_enabled_flagの値は0に等しいと推定される。

ph_lmcs_aps_idは、PHに関連付けられたスライスが参照するLMCS APSのadaptation_parameter_set_idを指定する。aps_params_typeがLMCS_APSと等しく、adaptation_parameter_set_idがph_lmcs_aps_idと等しいAPS NALユニットのTemporalIdは、PHに関連付けられたピクチャのTemporalId以下であるものとする。

20

ph_chroma_residual_scale_flagが1に等しいことは、PHに関連付けられているすべてのスライスについて、クロマ残差スケーリングが有効であることを指定する。ph_chroma_residual_scale_flagが0に等しいことは、PHに関連付けられた1つ以上又はすべてのスライスに対してクロマ残差スケーリングを無効にできることを指定する。ph_chroma_residual_scale_flagが存在しないとき、それは0に等しいと推定される。

ph_scaling_list_present_flagが1に等しいことは、PHに関連付けられたスライスに使用されるスケーリングリストデータが、参照されるスケーリングリストAPSに含まれるスケーリングリストデータに基づいて導出されることを指定する。ph_scaling_list_present_flagが0に等しいことは、PHに関連付けられたスライスに使用されるスケーリングリストデータが16に等しく設定されることを指定する。存在しないとき、ph_scaling_list_present_flagの値は0に等しいと推定される。

30

ph_scaling_list_aps_idは、スケーリングリストAPSのadaptation_parameter_set_idを指定する。aps_params_typeがSCALING_APSと等しく、adaptation_parameter_set_idがph_scaling_list_aps_idと等しいAPS NALユニットのTemporalIdは、PHに関連付けられたピクチャのTemporalId以下であるものとする。

ph_virtual_boundaries_present_flagが1に等しいことは、仮想境界の情報がPHでシグナリングされることを指定する。ph_virtual_boundaries_present_flagが0に等しいことは、仮想境界の情報がPHでシグナリングされないことを指定する。PHでシグナリングされている仮想境界が1つ以上ある場合、インループフィルタリング操作は、ピクチャ内の仮想境界に渡り無効になる。インループフィルタリング操作には、デブロッキングフィルタ、サンプル適応オフセットフィルタ、及び適応ループフィルタ操作が含まれる。存在しないとき、ph_virtual_boundaries_present_flagの値は0に等しいと推定される。

40

subpic_info_present_flagの値が1に等しい場合、ph_virtual_boundaries_present_flagの値が0に等しいことが、ビットストリーム準拠の要件である。

変数VirtualBoundariesPresentFlagは、以下のように導出される：

【数39】

```

VirtualBoundariesPresentFlag = 0
if( sps_virtual_boundaries_enabled_flag )
    VirtualBoundariesPresentFlag = sps_virtual_boundaries_present_flag | |
    ph_virtual_boundaries_present_flag
(83)

```

ph_num_ver_virtual_boundariesは、PHに存在するph_virtual_boundaries_pos_x[i]シンタックス要素の数を指定する。ph_num_ver_virtual_boundariesが存在しない場合は、0に等しいと推測される。

10

変数NumVerVirtualBoundariesは、以下のように導出される：

【数 4 0】

```

NumVerVirtualBoundaries = 0
if( sps_virtual_boundaries_enabled_flag )
    NumVerVirtualBoundaries = sps_virtual_boundaries_present_flag ?
    sps_num_ver_virtual_boundaries : ph_num_ver_virtual_boundaries
(84)

```

20

ph_virtual_boundaries_pos_x[i]は、ルマサンプルを8で割った単位でi番目の垂直仮想境界の位置を指定する。ph_virtual_boundaries_pos_x[i]の値は、両端を含む $1 - \text{Ceil}(\text{pic_width_in_luma_samples} \div 8) - 1$ の範囲とする。

両端を含む0からNumVerVirtualBoundaries - 1までの範囲のiについて、ルマサンプルの単位で垂直仮想境界の位置を指定するリストVirtualBoundariesPosX[i]は、次のように導き出される：

【数 4 1】

```

for( i = 0; i < NumVerVirtualBoundaries; i++)
    VirtualBoundariesPosX[ i ] = ( sps_virtual_boundaries_present_flag ?
    sps_virtual_boundaries_pos_x[ i ] : ph_virtual_boundaries_pos_x[ i ] ) * 8(85)

```

30

2つの垂直仮想境界間の距離は、CtbSizeYルマサンプル以上であるものとする。

ph_num_hor_virtual_boundariesは、PHに存在するph_virtual_boundaries_pos_y[i]シンタックス要素の数を指定する。ph_num_hor_virtual_boundariesが存在しない場合は、0に等しいと推測される。

変数NumHorVirtualBoundariesは、以下のように導出される：

【数 4 2】

```

NumHorVirtualBoundaries = 0
if( sps_virtual_boundaries_enabled_flag )
    NumHorVirtualBoundaries = sps_virtual_boundaries_present_flag ?
    sps_num_hor_virtual_boundaries : ph_num_hor_virtual_boundaries
(86)

```

40

sps_virtual_boundaries_enabled_flagが1に等しく、ph_virtual_boundaries_present_flagが1に等しい場合、ph_num_ver_virtual_boundariesとph_num_hor_virtual_boundariesの和は0より大きいものとする。

ph_virtual_boundaries_pos_y[i]は、ルマサンプルを8で割った単位でi番目の水平

50

仮想境界の位置を指定する。ph_virtual_boundaries_pos_y[i]の値は、両端を含む $1 \sim \text{Ceil}(\text{pic_height_in_luma_samples} \div 8) - 1$ の範囲であるものとする。

両端を含む 0 から NumHorVirtualBoundaries - 1 までの範囲の i について、ルマサンプルの単位で水平仮想境界の位置を指定するリスト VirtualBoundariesPosY[i] は、次のように導き出される：

【数 4 3】

```
for( i = 0; i < NumHorVirtualBoundaries; i++)
  VirtualBoundariesPosY[ i ] = ( sps_virtual_boundaries_present_flag ?
    sps_virtual_boundaries_pos_y[ i ] : ph_virtual_boundaries_pos_y[ i ] ) * 8(87)
```

10

2 つの水平仮想境界間の距離は、CtbSizeYルマサンプル以上であるものとする。

pic_output_flag は、AnnexC に規定されているように、復号ピクチャ出力及び削除処理に影響を与える。pic_output_flag が存在しない場合、それは、1 に等しいと推定される。

partition_constraints_override_flag が 1 に等しいことは、パーティション制約パラメータが PH に存在することを指定する。partition_constraints_override_flag が 0 に等しいことは、パーティション制約パラメータが PH に存在しないことを指定する。存在しないとき、partition_constraints_override_flag の値は 0 に等しいと推定される。

20

ph_log2_diff_min_qt_min_cb_intra_slice_luma は、CTU の 4 分木分割により生じたルマリーフブロックのルマサンプルのうち最小サイズの底 2 の対数と、slice_type が PH に関連付けられている 2 (1) に等しいスライス内のルマ CU のルマサンプルのうち最小コーディングブロックサイズの底 2 の対数との間の差を指定する。ph_log2_diff_min_qt_min_cb_intra_slice_luma の値は、 $0 \sim \text{CtbLog2SizeY} - \text{MinCbLog2SizeY}$ の範囲に包含されるものとする。存在しないとき、ph_log2_diff_min_qt_min_cb_intra_slice_luma の値は sps_log2_diff_min_qt_min_cb_intra_slice_luma に等しいと推定される。

ph_max_mtt_hierarchy_depth_intra_slice_luma は、PH に関連付けられている 2 (1) に等しい slice_type を有するスライスの中の 4 分木リーフのマルチタイプ木分割から生じるコーディングユニットのデフォルトの最大階層構造深さを指定する。ph_max_mtt_hierarchy_depth_intra_slice_luma の値は、両端を含む $0 \sim 2 * (\text{CtbLog2SizeY} - \text{MinCbLog2SizeY})$ の範囲であるものとする。存在しないとき、ph_max_mtt_hierarchy_depth_intra_slice_luma の値は、sps_max_mtt_hierarchy_depth_intra_slice_luma に等しいと推定される。

30

ph_log2_diff_max_bt_min_qt_intra_slice_luma は、2 分割を使用して分割できるルマコーディングブロックのルマサンプルのうち最大サイズ (幅又は高さ) の底 2 の対数と、slice_type が PH に関連付けられている 2 (1) に等しいスライス内の CTU の 4 分木分割によって生じるルマリーフブロックのルマサンプルのうち最小サイズ (幅又は高さ) との間の差を指定する。ph_log2_diff_max_bt_min_qt_intra_slice_luma の値は、 $0 \sim \text{CtbLog2SizeY} - \text{MinQtLog2SizeIntraY}$ の範囲に包含されるものとする。存在しないとき、ph_log2_diff_max_bt_min_qt_intra_slice_luma の値は sps_log2_diff_max_bt_min_qt_intra_slice_luma に等しいと推定される。

40

ph_log2_diff_max_tt_min_qt_intra_slice_luma は、3 分割を使用して分割できるルマコーディングブロックのルマサンプルのうち最大サイズ (幅又は高さ) の底 2 の対数と、slice_type が PH に関連付けられている 2 (1) に等しいスライス内の CTU の 4 分木分割によって生じるルマリーフブロックのルマサンプルのうち最小サイズ (幅又は高さ) との間の差を指定する。ph_log2_diff_max_tt_min_qt_intra_slice_luma の値は、 $0 \sim \text{CtbLog2SizeY} - \text{MinQtLog2SizeIntraY}$ の範囲に包含されるものとする。存在しないとき、ph_log2_diff_max_tt_min_qt_intra_slice_luma の値は sps_log2_diff_max_tt_min_qt_intra_slice_luma に等しいと推定される。

50

ph_log2_diff_min_qt_min_cb_intra_slice_chromaは、treeTypeがDUAL_TREE_CHROMAに等しいクロマCTUの4分木分割により生じたクロマリーフブロックのクロマサンプルのうちの最小サイズの底2の対数と、slice_typeがPHに関連付けられている2(1)に等しいスライスのうち、treeTypeがDUAL_TREE_CHROMAに等しいクロマCUのルマサンプルのうちの最小コーディングブロックサイズの底2の対数との間の差を指定する。ph_log2_diff_min_qt_min_cb_intra_slice_chromaの値は、0 ~ CtbLog2 SizeY-MinCbLog2 SizeYの範囲に包含されるものとする。存在しないとき、ph_log2_diff_min_qt_min_cb_intra_slice_chromaの値はsps_log2_diff_min_qt_min_cb_intra_slice_chromaに等しいと推定される。

ph_max_mtt_hierarchy_depth_intra_slice_chromaは、PHに関連付けられている2(I)に等しいslice_typeを有するスライスの中のtreeTypeがDUAL_TREE_CHROMAに等しいクロマ4分木リーフのマルチタイプ木分割から生じるクロマコーディングユニットの最大階層構造深さを指定する。ph_max_mtt_hierarchy_depth_intra_slice_chromaの値は、両端を含む0 ~ 2 * (CtbLog2 SizeY-MinCbLog2 SizeY)の範囲であるものとする。存在しないとき、ph_max_mtt_hierarchy_depth_intra_slice_chromaの値は、sps_max_mtt_hierarchy_depth_intra_slice_chromaに等しいと推定される。

ph_log2_diff_max_bt_min_qt_intra_slice_chromaは、2分割を使用して分割できるクロマコーディングブロックのルマサンプルのうち最大サイズ(幅又は高さ)の底2の対数と、slice_typeがPHに関連付けられている2(1)に等しいスライス内のtreeTypeがDUAL_TREE_CHROMAに等しいクロマCTUの4分木分割によって生じるクロマリーフブロックのルマサンプルのうちの最小サイズ(幅又は高さ)との間の差を指定する。ph_log2_diff_max_bt_min_qt_intra_slice_chromaの値は、0 ~ CtbLog2 SizeY-MinQtLog2 SizeIntraCの範囲に包含されるものとする。存在しないとき、ph_log2_diff_max_bt_min_qt_intra_slice_chromaの値はsps_log2_diff_max_bt_min_qt_intra_slice_chromaに等しいと推定される。

ph_log2_diff_max_tt_min_qt_intra_slice_chromaは、3分割を使用して分割できるクロマコーディングブロックのルマサンプルのうち最大サイズ(幅又は高さ)の底2の対数と、slice_typeがPHに関連付けられている2(1)に等しいスライス内のtreeTypeがDUAL_TREE_CHROMAに等しいクロマCTUの4分木分割によって生じるクロマリーフブロックのルマサンプルのうちの最小サイズ(幅又は高さ)との間の差を指定する。ph_log2_diff_max_tt_min_qt_intra_slice_chromaの値は、0 ~ CtbLog2 SizeY-MinQtLog2 SizeIntraCの範囲に包含されるものとする。存在しないとき、ph_log2_diff_max_tt_min_qt_intra_slice_chromaの値はsps_log2_diff_max_tt_min_qt_intra_slice_chromaに等しいと推定される。

ph_cu_qp_delta_subdiv_intra_sliceは、cu_qp_delta_absとcu_qp_delta_sign_flagを伝えるイントラスライス内のコーディングユニットの最大cbSubdiv値を指定する。ph_cu_qp_delta_subdiv_intra_sliceの値は、両端を含む0から2 * (CtbLog2 SizeY-MinQtLog2 SizeIntraY + ph_max_mtt_hierarchy_depth_intra_slice_luma)までの範囲であるものとする。

存在しないとき、ph_cu_qp_delta_subdiv_intra_sliceの値は0に等しいと推定される。

ph_cu_chroma_qp_offset_subdiv_intra_sliceは、cu_chroma_qp_offset_flagを伝えるイントラスライス内のコーディングユニットの最大cbSubdiv値を指定する。ph_cu_chroma_qp_offset_subdiv_intra_sliceの値は、両端を含む0から2 * (CtbLog2 SizeY-MinQtLog2 SizeIntraY + ph_max_mtt_hierarchy_depth_intra_slice_luma)までの範囲であるものとする。

存在しないとき、ph_cu_chroma_qp_offset_subdiv_intra_sliceの値は0に等しいと推定される。

ph_log2_diff_min_qt_min_cb_inter_sliceは、CTUの4分木分割により生じたルマリーフブロックのルマサンプルのうちの最小サイズの底2の対数と、slice_typeがPHに

10

20

30

40

50

関連付けられている 0 (B) 又は 1 (P) に等しいスライスのルマCUのルマサンプルのうちの最小ルマコーディングブロックサイズの底 2 の対数との間の差を指定する。ph_log 2_diff_min_qt_min_cb_inter_sliceの値は、 $0 \sim \text{CtbLog 2 SizeY} - \text{MinCbLog 2 SizeY}$ の範囲に包含されるものとする。存在しないとき、ph_log 2_diff_min_qt_min_cb_lumaの値はsps_log 2_diff_min_qt_min_cb_inter_sliceに等しいと推定される。

ph_max_mtt_hierarchy_depth_inter_sliceは、slice_typeがPHに関連付けられている 0 (B) 又は 1 (P) に等しいスライス内の 4 分木リーフのマルチタイプツリー分割から生じるコーディングユニットの最大階層構造深さを指定する。ph_max_mtt_hierarchy_depth_inter_sliceの値は、両端を含む $0 \sim 2 * (\text{CtbLog 2 SizeY} - \text{MinCbLog 2 SizeY})$ の範囲であるものとする。存在しないとき、ph_max_mtt_hierarchy_depth_inter_sliceの値は、sps_max_mtt_hierarchy_depth_inter_sliceに等しいと推定される。

10

ph_log 2_diff_max_bt_min_qt_inter_sliceは、2 分割を使用して分割できるルマコーディングブロックのルマサンプルのうち最大サイズ (幅又は高さ) の底 2 の対数と、slice_typeがPHに関連付けられている 0 (B) 又は 1 (P) に等しいスライス内のCTUの 4 分木分割によって生じるルマリーフブロックのルマサンプルのうちの最小サイズ (幅又は高さ) との間の差を指定する。ph_log 2_diff_max_bt_min_qt_inter_sliceの値は、 $0 \sim \text{CtbLog 2 SizeY} - \text{MinQtLog 2 SizeInterY}$ の範囲に包含されるものとする。存在しないとき、ph_log 2_diff_max_bt_min_qt_inter_sliceの値はsps_log 2_diff_max_bt_min_qt_inter_sliceに等しいと推定される。

ph_log 2_diff_max_tt_min_qt_inter_sliceは、3 分割を使用して分割できるルマコーディングブロックのルマサンプルのうち最大サイズ (幅又は高さ) の底 2 の対数と、slice_typeがPHに関連付けられている 0 (B) 又は 1 (P) に等しいスライス内のCTUの 4 分木分割によって生じるルマリーフブロックのルマサンプルのうちの最小サイズ (幅又は高さ) との間の差を指定する。ph_log 2_diff_max_tt_min_qt_inter_sliceの値は、 $0 \sim \text{CtbLog 2 SizeY} - \text{MinQtLog 2 SizeInterY}$ の範囲に包含されるものとする。存在しないとき、ph_log 2_diff_max_tt_min_qt_inter_sliceの値はsps_log 2_diff_max_tt_min_qt_inter_sliceに等しいと推定される。

20

ph_cu_qp_delta_subdiv_inter_sliceは、cu_qp_delta_absとcu_qp_delta_sign_flagを伝えるインタースライス内のコーディングユニットの最大cbSubdiv値を指定する。ph_cu_qp_delta_subdiv_inter_sliceの値は、両端を含む $0 \sim 2 * (\text{CtbLog 2 SizeY} - \text{MinQtLog 2 SizeInterY} + \text{ph_max_mtt_hierarchy_depth_inter_slice})$ の範囲であるものとする。

30

存在しないとき、ph_cu_qp_delta_subdiv_inter_sliceの値は 0 に等しいと推定される。

ph_cu_chroma_qp_offset_subdiv_inter_sliceは、cu_chroma_qp_offset_flagを伝えるインタースライス内のコーディングユニットの最大cbSubdiv値を指定する。ph_cu_chroma_qp_offset_subdiv_inter_sliceの値は、両端を含む $0 \sim 2 * (\text{CtbLog 2 SizeY} - \text{MinQtLog 2 SizeInterY} + \text{ph_max_mtt_hierarchy_depth_inter_slice})$ の範囲であるものとする。

存在しないとき、ph_cu_chroma_qp_offset_subdiv_inter_sliceの値は 0 に等しいと推定される。

40

ph_temporal_mvp_enabled_flagは、時間的動きベクトル予測子をPHに関連付けられているスライスのインター予測に使用できるかどうかを指定する。ph_temporal_mvp_enabled_flagが 0 に等しい場合、PHに関連するスライスのシンタックス要素は、スライスの復号に時間的動きベクトル予測子が使用されないように制約されるものとする。その他の場合 (ph_temporal_mvp_enabled_flagが 1 に等しい)、時間的動きベクトル予測子は、PHに関連付けられたスライスの復号に使用されてよい。存在しないとき、ph_temporal_mvp_enabled_flagの値は 0 に等しいと推定される。DPB内に現在の画像と同じ空間解像度を持つ参照画像がない場合、ph_temporal_mvp_enabled_flagの値は 0 に等しいものとする。

50

サブブロックに基づくマージMVP候補の最大数MaxNumSubblockMergeCandは、次のように導出される。

【数44】

```

if( sps_affine_enabled_flag )
    MaxNumSubblockMergeCand = 5 - five_minus_max_num_subblock_merge_cand(88)
else
    MaxNumSubblockMergeCand =
    sps_sbtmvp_enabled_flag && ph_temporal_mvp_enable_flag

```

10

MaxNumSubblockMergeCandの値は、両端を含む0～5の範囲であるものとする。

ph_collocated_from_l0_flagが1に等しいことは、時間的動きベクトル予測に使用される同一位置ピクチャが参照ピクチャリスト0から導出されることを指定する。ph_collocated_from_l0_flagが0に等しいことは、時間的動きベクトル予測に使用される同一位置ピクチャが参照ピクチャリスト1から導出されることを指定する。

ph_collocated_ref_idxは、時間的動きベクトル予測に使用される同一位置ピクチャの参照インデックスを指定する。

ph_collocated_from_l0_flagが1に等しい場合、ph_collocated_ref_idxは参照ピクチャリスト0のエントリを参照し、ph_collocated_ref_idxの値は両端を含む0からnum_ref_entries[0][RplIdx[0]]-1までの範囲にあるものとする。

20

ph_collocated_from_l0_flagが0に等しい場合、ph_collocated_ref_idxは参照ピクチャリスト1のエントリを参照し、ph_collocated_ref_idxの値は両端を含む0からnum_ref_entries[1][RplIdx[1]]-1までの範囲にあるものとする。

存在しないとき、ph_collocated_ref_idxの値は0に等しいと推定される。

1に等しいmvd_l1_zero_flagは、mvd_coding(x0,y0,1)シンタックス構造がパースされないこと、compIdx=0..1及びcpIdx=0..2について、MvdL1[x0][y0][compIdx]及びMvdCpL1[x0][y0][cpIdx][compIdx]が0に設定されることを示す。mvd_l1_zero_flagが0に等しいことは、mvd_coding(x0,y0,1)シンタックス構造がパースされることを示す。

30

ph_fpel_mmvd_enabled_flagが1に等しいことは、動きベクトル差によるマージモードが、PHに関連付けられているスライス内で整数サンプル精度を使用することを指定する。ph_fpel_mmvd_enabled_flagが0に等しいことは、動きベクトル差によるマージモードが、PHに関連付けられているスライス内で分数サンプル精度を使用できることを指定する。存在しないとき、ph_fpel_mmvd_enabled_flagの値は0であると推定される。

ph_disable_bdof_flagが1に等しいことは、PHに関連付けられているスライス内で双方向オプティカルフローインター予測に基づくインター双予測が無効にされることを指定する。ph_disable_bdof_flagが0に等しいことは、PHに関連付けられているスライス内で双方向オプティカルフローインター予測に基づくインター双予測が有効にされてもされなくてもよいことを指定する。

40

ph_disable_bdof_flagが存在しない場合、以下が適用される：

- sps_bdof_enabled_flagが1に等しい場合、ph_disable_bdof_flagの値は0に等しいと推定される。

- それ以外の場合 (sps_bdof_enabled_flagが0に等しい)、ph_disable_bdof_flagの値は1に等しいと推定される。

ph_disable_dmvr_flagが1に等しいことは、PHに関連付けられているスライスで、デコーダ動きベクトル精緻化に基づくインター双予測が無効にされることを指定する。ph_disable_dmvr_flagが0に等しいことは、PHに関連付けられているスライスで、デコーダ動きベクトル精緻化に基づくインター双予測が有効にされてもされなくてもよいことを指定する。

50

ph_disable_dmvr_flagが存在しない場合、以下が適用される：

- sps_dmvr_enabled_flagが1に等しい場合、ph_disable_dmvr_flagの値は0に等しいと推定される。

- それ以外の場合 (sps_dmvr_enabled_flagが0に等しい)、ph_disable_dmvr_flagの値は1に等しいと推定される。

ph_disable_prof_flagが1に等しい場合は、PHに関連付けられたスライスでオプティカルフローによる予測精緻化が無効にされることを指定する。ph_disable_prof_flagが0に等しい場合は、PHに関連付けられたスライスでオプティカルフローによる予測精緻化が有効にされてもされなくてもよいことを指定する。

ph_disable_prof_flagが存在しない場合、以下が適用される：

- sps_affine_prof_enabled_flagが1に等しい場合、ph_disable_prof_flagの値は0に等しいと推定される、

- それ以外の場合 (sps_affine_prof_enabled_flagが0に等しい)、ph_disable_prof_flagの値は1に等しいと推定される。

ph_qp_deltaは、コードユニット層のCuQpDeltaValの値で修正されるまで、ピクチャ内のコーディングブロックに使用されるQpYの初期値を指定する。

qp_delta_info_in_ph_flagが1に等しい場合、ピクチャのすべてのスライスに対するQpY量子化パラメータの初期値SliceQpYは次のように導出される：

【数 4 5】

$$\text{SliceQpY} = 26 + \text{init_qp_minus26} + \text{ph_qp_delta}$$

(89)

SliceQpYの値は、両端を含む-QpBdOffset ~ + 6 3の範囲であるものとする。

ph_joint_cbcr_sign_flagは、tu_joint_cbcr_residual_flag[x 0][y 0]が1に等しい変換ユニットで、両方のクロマ成分の同一位置残余サンプルが反転符号を持つかどうかを指定する。変換ユニットに対してtu_joint_cbcr_residual_flag[x 0][y 0]が1に等しい場合、ph_joint_cbcr_sign_flagが0に等しいことは、Cr (又はCb) 成分の各残差サンプルの符号が同一位置Cb (又はCr) 残差サンプルの符号と同一であることを指定し、ph_joint_cbcr_sign_flagが1に等しいことは、Cr (又はCb) 成分の各残差サンプルの符号が同一位置Cb (又はCr) 残差サンプルの逆符号で与えられることを指定する。

ph_sao_luma_enabled_flagが1に等しいことは、PHに関連付けられたすべてのスライスでルマ成分についてSAOが有効にされることを指定する。ph_sao_luma_enabled_flagが0に等しいことは、PHに関連付けられた1つ以上又はすべてのスライスに対して、ルマ成分のためのSAOを無効にすることができることを指定する。ph_sao_luma_enabled_flagが存在しない場合は、0に等しいと推測される。

ph_sao_luma_enabled_flagが1に等しいことは、PHに関連付けられたすべてのスライスのクロマ成分に対してSAOが有効になる。ph_sao_chroma_enabled_flagが0に等しいことは、PHに関連付けられた1つ以上又はすべてのスライスに対して、クロマ成分のためのSAOを無効にすることができることを指定する。ph_sao_chroma_enabled_flagが存在しない場合は、0に等しいと推測される。

ph_dep_quant_enabled_flagが0に等しいことは、現在ピクチャに対して依存量子化を無効にすることを指定する。ph_dep_quant_enabled_flagが1に等しいことは、現在ピクチャに対して依存量子化を有効にすることを指定する。ph_dep_quant_enabled_flagが存在しない場合は、0に等しいと推測される。

pic_sign_data_hiding_enabled_flagが0に等しいことは、現在ピクチャに対して符号ビット非表示を無効にすることを指定する。pic_sign_data_hiding_enabled_flagが1に等しいことは、現在ピクチャに対して符号ビット非表示を有効にすることを指定する。pic_sign_data_hiding_enabled_flagが存在しない場合、0と等しいと推測される。

ph_deblocking_filter_override_flagが1に等しいことは、デブロッキングパラメー

10

20

30

40

50

タがPHに存在することを指定する。ph_deblocking_filter_override_flagが0に等しいことは、デブロッキングパラメータがPHに存在しないことを指定する。存在しないとき、ph_deblocking_filter_override_flagの値は0に等しいと推定される。

ph_deblocking_filter_disabled_flagが1に等しいことは、デブロッキングフィルタの操作がPHに関連付けられているスライスに適用されないことを指定する。ph_deblocking_filter_disabled_flagが0に等しいことは、デブロッキングフィルタの操作が、PHに関連付けられているスライスに適用されることを指定する。ph_deblocking_filter_disabled_flagが存在しない場合、それは、pps_deblocking_filter_disabled_flagに等しいと推定される。

ph_beta_offset_div2及びph_tc_offset_div2は、PHに関連付けられているスライスについて、ルマ成分に適用される及びtC(2で除算されたもの)のデブロッキングパラメータオフセットを指定する。ph_beta_offset_div2とph_tc_offset_div2の値は、両方とも両端を含む-12から12の範囲内にあるものとする。存在しない場合、ph_beta_offset_div2とph_tc_offset_div2の値は、各々pps_beta_offset_div2とpps_tc_offset_div2に等しいと推定される。

10

ph_cb_beta_offset_div2及びph_cb_tc_offset_div2は、PHに関連付けられているスライスについて、Cb成分に適用される及びtC(2で除算されたもの)のデブロッキングパラメータオフセットを指定する。ph_cb_beta_offset_div2とph_cb_tc_offset_div2の値は、両方とも両端を含む-12から12の範囲内にあるものとする。存在しない場合、ph_cb_beta_offset_div2とph_cb_tc_offset_div2の値は、各々pps_cb_beta_offset_div2とpps_cb_tc_offset_div2に等しいと推定される。

20

ph_cr_beta_offset_div2及びph_cr_tc_offset_div2は、PHに関連付けられているスライスについて、Cr成分に適用される及びtC(2で除算されたもの)のデブロッキングパラメータオフセットを指定する。ph_cr_beta_offset_div2とph_cr_tc_offset_div2の値は、両方とも両端を含む-12から12の範囲内にあるものとする。存在しない場合、ph_cr_beta_offset_div2とph_cr_tc_offset_div2の値は、各々pps_cr_beta_offset_div2とpps_cr_tc_offset_div2に等しいと推定される。

ph_extension_lengthは、PH拡張データの長さをバイト単位で指定するもので、ph_extension_length自体のシグナリングに使用されるビットは含まれない。ph_extension_lengthの値は、両端を含む0~256の範囲であるものとする。存在しないとき、ph_extension_lengthの値は0に等しいと推定される。

30

ph_extension_data_byteは、任意の値を有してよい。この仕様のこのバージョンに準拠するデコーダは、ph_extension_data_byteの値を無視するものとする。その値は、この仕様のこのバージョン内で指定されるプロファイルへのデコーダ準拠性に影響しない。

【0041】

3.6 SHのシンタックスとセマンティクス

最新のVVCドラフトテキストでは、SHシンタックスとセマンティクスは次の通りである。

。

40

50

【表 1 2】

slice_header() {	Descriptor	
picture_header_in_slice_header_flag	u(1)	
if(picture_header_in_slice_header_flag)		
picture_header_structure()		
if(subpic_info_present_flag)		
slice_subpic_id	u(v)	
if((rect_slice_flag && NumSlicesInSubpic[CurrSubpicIdx] > 1) (!rect_slice_flag && NumTilesInPic > 1))		10
slice_address	u(v)	
for(i = 0; i < NumExtraShBits; i++)		
sh_extra_bit[i]	u(1)	
if(!rect_slice_flag && NumTilesInPic > 1)		
num_tiles_in_slice_minus1	ue(v)	
if(ph_inter_slice_allowed_flag)		
slice_type	ue(v)	
if(sps_alf_enabled_flag && !alf_info_in_ph_flag) {		20
slice_alf_enabled_flag	u(1)	
if(slice_alf_enabled_flag) {		
slice_num_alf_aps_ids_luma	u(3)	
for(i = 0; i < slice_num_alf_aps_ids_luma; i++)		
slice_alf_aps_id_luma[i]	u(3)	
if(ChromaArrayType != 0)		
slice_alf_chroma_idc	u(2)	
if(slice_alf_chroma_idc)		
slice_alf_aps_id_chroma	u(3)	
if(sps_ccalf_enabled_flag) {		30
slice_cc_alf_cb_enabled_flag	u(1)	
if(slice_cc_alf_cb_enabled_flag)		
slice_cc_alf_cb_aps_id	u(3)	
slice_cc_alf_cr_enabled_flag	u(1)	
if(slice_cc_alf_cr_enabled_flag)		
slice_cc_alf_cr_aps_id	u(3)	
}		
}		
}		40

10

20

30

40

50

if(separate_colour_plane_flag == 1)		
colour_plane_id	u(2)	
if(!rpl_info_in_ph_flag && ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag))		
ref_pic_lists()		
if((rpl_info_in_ph_flag ((nal_unit_type != IDR_W_RADL && nal_unit_type != IDR_N_LP) sps_idr_rpl_present_flag)) && (slice_type != I && num_ref_entries[0][RplsIdx[0]] > 1) (slice_type == B && num_ref_entries[1][RplsIdx[1]] > 1)) {		10
num_ref_idx_active_override_flag	u(1)	
if(num_ref_idx_active_override_flag)		
for(i = 0; i < (slice_type == B ? 2 : 1); i++)		
if(num_ref_entries[i][RplsIdx[i]] > 1)		
num_ref_idx_active_minus1[i]	ue(v)	20
}		
if(slice_type != I) {		
if(cabac_init_present_flag)		
cabac_init_flag	u(1)	
if(ph_temporal_mvp_enabled_flag && !rpl_info_in_ph_flag) {		
if(slice_type == B)		
slice_collocated_from_l0_flag	u(1)	
if((slice_collocated_from_l0_flag && NumRefIdxActive[0] > 1) (! slice_collocated_from_l0_flag && NumRefIdxActive[1] > 1))		30
slice_collocated_ref_idx	ue(v)	
}		
if(!wp_info_in_ph_flag && ((pps_weighted_pred_flag && slice_type == P) (pps_weighted_bipred_flag && slice_type == B)))		
pred_weight_table()		
}		
if(!qp_delta_info_in_ph_flag)		
slice_qp_delta	se(v)	40

if(pps_slice_chroma_qp_offsets_present_flag) {		
slice_cb_qp_offset	se(v)	
slice_cr_qp_offset	se(v)	
if(sps_joint_cbr_enabled_flag)		
slice_joint_cbr_qp_offset	se(v)	
}		
if(pps_cu_chroma_qp_offset_list_enabled_flag)		10
cu_chroma_qp_offset_enabled_flag	u(1)	
if(sps_sao_enabled_flag && !sao_info_in_ph_flag) {		
slice_sao_luma_flag	u(1)	
if(ChromaArrayType != 0)		
slice_sao_chroma_flag	u(1)	
}		
if(deblocking_filter_override_enabled_flag && !dbf_info_in_ph_flag)		
slice_deblocking_filter_override_flag	u(1)	
if(slice_deblocking_filter_override_flag) {		20
slice_deblocking_filter_disabled_flag	u(1)	
if(!slice_deblocking_filter_disabled_flag) {		
slice_beta_offset_div2	se(v)	
slice_tc_offset_div2	se(v)	
slice_cb_beta_offset_div2	se(v)	
slice_cb_tc_offset_div2	se(v)	
slice_cr_beta_offset_div2	se(v)	
slice_cr_tc_offset_div2	se(v)	
}		
}		30
slice_ts_residual_coding_disabled_flag	u(1)	
if(ph_lmcs_enabled_flag)		
slice_lmcs_enabled_flag	u(1)	
if(ph_scaling_list_enabled_flag)		
slice_scaling_list_present_flag	u(1)	
if(NumEntryPoints > 0) {		
offset_len_minus1	ue(v)	
for(i = 0; i < NumEntryPoints; i++)		
entry_point_offset_minus1[i]	u(v)	40

}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte[i]	u(8)
}	
byte_alignment()	
}	

10

変数CuQpDeltaValは、cu_qp_delta_absを含むコーディングユニットのルマ量子化パラメータと、その予測との間の差を指定し、0に等しく設定される。変数CuQpOffsetCb、CuQpOffsetCr、及びCuQpOffsetCbCrは、cu_chroma_qp_offset_flagを含むコーディングユニットのQp_{cb}、Qp_{cr}、及びQp_{cbCr}量子化パラメータの各々の値を決定するときに使用されるべき値を指定し、全部0に等しく設定される。

picture_header_in_slice_header_flagが1に等しいことは、スライスヘッダにPHシンタックス構造が存在することを指定する。picture_header_in_slice_header_flagが0に等しいことは、スライスヘッダにPHシンタックス構造が存在しないことを指定する。

20

ビデオビットストリーム準拠の要件は、picture_header_in_slice_header_flagの値が、CLVS内の全てのコーディングスライスで同じであることである。

コーディングスライスのpicture_header_in_slice_header_flagが1に等しい場合、nal_unit_typeがPH_NUTと等しいVCL NALユニットがCLVSに存在しないことが、ビットストリーム準拠の要件となる。

picture_header_in_slice_header_flagが0に等しい場合、現在ピクチャ内のすべてのコーディングされたスライスは、picture_header_in_slice_header_flagが0に等しく、現在PUがPH NALユニットを持つことである。

30

slice_subpic_idは、スライスを含むサブピクチャのサブピクチャIDを指定する。slice_subpic_idが存在する場合、変数CurrSubpicIdxの値は、SubpicIdVal[CurrSubpicIdx]がslice_subpic_idと等しくなるように導出される。それ以外の場合(slice_subpic_idが存在しない)、CurrSubpicIdxは0に等しくなるよう導出される。slice_subpic_idの長さはsps_subpic_id_len_minus_1 + 1ビットである。

slice_addressはスライスのスライスアドレスを指定する。存在しないとき、slice_addressの値は0に等しいと推定される。rect_slice_flagが0に等しく、NumSlicesInSubpic[CurrSubpicIdx]が1に等しい場合、slice_addressの値は0に等しいと推定される。

rect_slice_flagが0に等しい場合、以下が適用される：

- スライスアドレスはラスタースキャンラインインデックスである、
- slice_addressの長さはCeil(Log 2 (NumTilesInPic)) ビットである、
- slice_addressの値は、両端を含む0 ~ NumTilesInPic- 1の範囲であるものとする、
- その他の場合(rect_slice_flagが1に等しい)、以下が適用される：
- スライスアドレスは、スライスのサブピクチャレベルのスライスインデックスである、
- slice_addressの長さはCeil(Log 2 (NumSlicesInSubpic[CurrSubpicIdx])) ビット)である、
- slice_addressの値は、両端を含む0 ~ NumSlicesInSubpic[CurrSubpicIdx]- 1の範囲であるものとする。

40

ビットストリーム適合性の要件は、以下の制約が適用されることである：

- rect_slice_flagが0に等しい又はsubpic_info_present_flagが0に等しい場合、sl

50

ice_addressの値は、同じコーディングピクチャの他のコーディングスライスNALユニットのslice_addressの値と等しくないものとする、

- その他の場合、slice_subpic_id値とslice_address値のペアは、同じコーディングピクチャの他のコーディングスライスNALユニットのslice_subpic_id値とslice_address値のペアと等しくないものとする、

- ピクチャのスライスの形状は、各CTUが、復号されたときに、ピクチャ境界からなるか、又は以前に復号されたCTUの境界からなる、その左側境界全体及び上側境界全体を有するものとする。

sh_extra_bit[i]は1又は0に等しくてよい。この仕様のこのバージョンに準拠するデコーダは、valueofsh_extra_bit[i]の値を無視するものとする。その値は、この仕様のこのバージョン内で指定されるプロファイルへのデコーダ準拠性に影響しない。

num_tiles_in_slice_minus_1に1を加えたものは、存在するとき、スライス内のタイル数を指定する。num_tiles_in_slice_minus_1の値は、両端を含む0 ~ NumTilesInPic-1の範囲であるものとする。

現在スライス内のCTUの数を指定する変数NumCtusInCurrSliceと、スライス内のi番目のCTBのピクチャラスタスキャンアドレスを指定するリストCtbAddrInCurrSlice[i]は、両端を含む0 ~ NumCtusInCurrSlice-1の範囲のiについて、次のように導出される：

【数46】

```

if( rect_slice_flag ) {
    picLevelSliceIdx = slice_address
    for( j = 0; j < CurrSubpicIdx; j++ )
        picLevelSliceIdx += NumSlicesInSubpic[ j ]
    NumCtusInCurrSlice = NumCtusInSlice[ picLevelSliceIdx ]
    for( i = 0; i < NumCtusInCurrSlice; i++ )
        CtbAddrInCurrSlice[ i ] = CtbAddrInSlice[ picLevelSliceIdx ][ i ]
} else {
    NumCtusInCurrSlice = 0
    for( tileIdx = slice_address; tileIdx <= slice_address + num_tiles_in_slice_minus_1;
        tileIdx++ ) {
        tileX = tileIdx % NumTileColumns
        tileY = tileIdx / NumTileColumns
        for( ctbY = tileRowBd[ tileY ]; ctbY < tileRowBd[ tileY + 1 ]; ctbY++ ) {
            for( ctbX = tileColBd[ tileX ]; ctbX < tileColBd[ tileX + 1 ]; ctbX++ ) {
                CtbAddrInCurrSlice[ NumCtusInCurrSlice ] = ctbY * PicWidthInCtb + ctbX
                NumCtusInCurrSlice++
            }
        }
    }
}

```

変数SubpicLeftBoundaryPos、SubpicTopBoundaryPos、SubpicRightBoundaryPos、及びSubpicBotBoundaryPosは、次のように導出される：

【数47】

10

20

30

40

50

```

if( subpic_treated_as_pic_flag[ CurrSubpicIdx ] ) {
  SubpicLeftBoundaryPos = subpic_ctu_top_left_x[ CurrSubpicIdx ] * CtbSizeY
  SubpicRightBoundaryPos = Min( pic_width_max_in_luma_samples - 1,
    ( subpic_ctu_top_left_x[ CurrSubpicIdx ] +
      subpic_width_minus1[ CurrSubpicIdx ] + 1 ) * CtbSizeY - 1 )
  SubpicTopBoundaryPos = subpic_ctu_top_left_y[ CurrSubpicIdx ] * CtbSizeY (118)
  SubpicBotBoundaryPos = Min( pic_height_max_in_luma_samples - 1,
    ( subpic_ctu_top_left_y[ CurrSubpicIdx ] +
      subpic_height_minus1[ CurrSubpicIdx ] + 1 ) * CtbSizeY - 1 )
}

```

10

slice_typeは、表 9 に従ってスライスのコーディングタイプを指定する。

【表 1 3】

表9 – slice_typeと名称の関連

slice_type	slice_type の名称
0	B (Bスライス)
1	P (Pスライス)
2	I (Iスライス)

20

存在しないとき、slice_type値は2に等しいと推定される。

ph_intra_slice_allowed_flagが0に等しい場合、slice_typeの値は0又は1に等しくなるものとする。nal_unit_typeが、両端を含むDR_W_RADLからCRA_NUTまでの範囲である場合、vps_independent_layer_flag[GeneralLayerIdx[nuh_layer_id]]は1に等しく、slice_typeは2に等しいものとする

30

変数MinQtLog2SizeY、MinQtLog2SizeC、MinQtSizeY、MinQtSizeC、MaxBtSizeY、MaxBtSizeC、MinBtSizeY、MaxTtSizeY、MaxTtSizeC、MinTtSizeY、MaxMttDepthY、及びMaxMttDepthCは、以下のように導出される：

- slice_typeが2 (I) に等しい場合、以下が適用される：

【数 4 8】

40

50

$$\text{MinQtLog2SizeY} = \text{MinCbLog2SizeY} + \text{ph_log2_diff_min_qt_min_cb_intra_slice_luma} \quad (119)$$

$$\text{MinQtLog2SizeC} = \text{MinCbLog2SizeY} + \text{ph_log2_diff_min_qt_min_cb_intra_slice_chroma} \quad (120)$$

$$\text{MaxBtSizeY} = 1 \ll (\text{MinQtLog2SizeY} + \text{ph_log2_diff_max_bt_min_qt_intra_slice_luma}) \quad (121)$$

$$\text{MaxBtSizeC} = 1 \ll (\text{MinQtLog2SizeC} + \text{ph_log2_diff_max_bt_min_qt_intra_slice_chroma}) \quad (122)$$

$$\text{MaxTtSizeY} = 1 \ll (\text{MinQtLog2SizeY} + \text{ph_log2_diff_max_tt_min_qt_intra_slice_luma}) \quad (123)$$

$$\text{MaxTtSizeC} = 1 \ll (\text{MinQtLog2SizeC} + \text{ph_log2_diff_max_tt_min_qt_intra_slice_chroma}) \quad (124)$$

$$\text{MaxMttDepthY} = \text{ph_max_mtt_hierarchy_depth_intra_slice_luma} \quad (125)$$

$$\text{MaxMttDepthC} = \text{ph_max_mtt_hierarchy_depth_intra_slice_chroma} \quad (126)$$

$$\text{CuQpDeltaSubdiv} = \text{ph_cu_qp_delta_subdiv_intra_slice} \quad (127)$$

$$\text{CuChromaQpOffsetSubdiv} = \text{ph_cu_chroma_qp_offset_subdiv_intra_slice} \quad (128)$$

– Otherwise (slice_type equal to 0 (B) or 1 (P)), the following applies:

$$\text{MinQtLog2SizeY} = \text{MinCbLog2SizeY} + \text{ph_log2_diff_min_qt_min_cb_inter_slice} \quad (129)$$

$$\text{MinQtLog2SizeC} = \text{MinCbLog2SizeY} + \text{ph_log2_diff_min_qt_min_cb_inter_slice} \quad (130)$$

$$\text{MaxBtSizeY} = 1 \ll (\text{MinQtLog2SizeY} + \text{ph_log2_diff_max_bt_min_qt_inter_slice}) \quad (131)$$

$$\text{MaxBtSizeC} = 1 \ll (\text{MinQtLog2SizeC} + \text{ph_log2_diff_max_bt_min_qt_inter_slice}) \quad (132)$$

$$\text{MaxTtSizeY} = 1 \ll (\text{MinQtLog2SizeY} + \text{ph_log2_diff_max_tt_min_qt_inter_slice}) \quad (133)$$

$$\text{MaxTtSizeC} = 1 \ll (\text{MinQtLog2SizeC} + \text{ph_log2_diff_max_tt_min_qt_inter_slice}) \quad (134)$$

$$\text{MaxMttDepthY} = \text{ph_max_mtt_hierarchy_depth_inter_slice} \quad (135)$$

$$\text{MaxMttDepthC} = \text{ph_max_mtt_hierarchy_depth_inter_slice} \quad (136)$$

$$\text{CuQpDeltaSubdiv} = \text{ph_cu_qp_delta_subdiv_inter_slice} \quad (137)$$

$$\text{CuChromaQpOffsetSubdiv} = \text{ph_cu_chroma_qp_offset_subdiv_inter_slice} \quad (138)$$

– The following applies:

$$\text{MinQtSizeY} = 1 \ll \text{MinQtLog2SizeY} \quad (139)$$

$$\text{MinQtSizeC} = 1 \ll \text{MinQtLog2SizeC} \quad (140)$$

$$\text{MinBtSizeY} = 1 \ll \text{MinCbLog2SizeY} \quad (141)$$

$$\text{MinTtSizeY} = 1 \ll \text{MinCbLog2SizeY} \quad (142)$$

slice_alf_enabled_flagが1に等しいことは、適応ループフィルタが有効になり、スライス内のY、Cb、又はCr色成分に適用できることを指定する。slice_alf_enabled_flagが0に等しいことは、スライス内の全ての色成分について適応ループフィルタが無効になることを指定する。存在しないとき、slice_alf_enabled_flag値は、ph_alf_enabled_flagに等しいと推定される。

slice_num_alf_aps_ids_lumaは、スライスが参照するALF APSの数を指定する。slice_alf_enabled_flagが1に等しく、slice_num_alf_aps_ids_lumaが存在しないとき、slice_num_alf_aps_ids_lumaの値はph_num_alf_aps_ids_lumaの値に等しいと推定される。slice_alf_aps_id_luma[i]は、スライスのルマ成分が参照するi番目のALF A

10

20

30

40

50

PSのadaptation_parameter_set_idを指定する。aps_params_typeがALF_APSと等しく、adaptation_parameter_set_idがslice_alf_aps_id_luma[i]と等しいAPS NALユニットのTemporalIdは、コーディングスライスNALユニットのTemporalId以下であるものとする。slice_alf_enabled_flagが1に等しく、slice_alf_aps_id_luma[i]が存在しないとき、slice_alf_aps_id_luma[i]の値はph_alf_aps_id_luma[i]の値に等しいと推定される。

aps_params_typeがALF_APSに等しく、adaptation_parameter_set_idがslice_alf_aps_id_luma[i]に等しいAPS NALユニットのalf_luma_filter_signal_flagの値は1に等しいものとする。

slice_alf_chroma_idcが0に等しいことは、CbとCr色成分に適応ループフィルタが適用されないことを指定する。slice_alf_chroma_idcが1に等しいことは、Cb色成分に適応ループフィルタが適用されることを示す。slice_alf_chroma_idcが2に等しいことは、Cr色成分に適応ループフィルタが適用されることを示す。slice_alf_chroma_idcが3に等しいことは、Cb及びCr色成分に適応ループフィルタが適用されることを示す。slice_alf_chroma_idcが存在しない場合、それは、ph_alf_chroma_idcに等しいと推定される。

10

slice_alf_aps_id_chromaは、スライスのクロマ成分が参照するi番目のALF APSのadaptation_parameter_set_idを指定する。aps_params_typeがALF_APSと等しく、adaptation_parameter_set_idがslice_alf_aps_id_chromaと等しいAPS NALユニットのTemporalIdは、コーディングスライスNALユニットのTemporalId以下であるものとする。slice_alf_enabled_flagが1に等しく、slice_alf_aps_id_chromaが存在しないとき、slice_alf_aps_id_chromaの値はph_alf_aps_id_chromaの値に等しいと推定される。

20

aps_params_typeがALF_APSに等しく、adaptation_parameter_set_idがslice_alf_aps_id_chromaに等しいAPS NALユニットのalf_chroma_filter_signal_flagの値は1に等しいものとする。

slice_cc_alf_cb_enabled_flagが0に等しいことは、成分間フィルタがCb色成分に適応されないことを指定する。slice_cc_alf_cb_enabled_flagが1に等しいことは、成分間フィルタが有効であり、Cb色成分に適応される可能性があることを指定する。slice_cc_alf_cb_enabled_flagが存在しない場合、それは、ph_cc_alf_cb_enabled_flagに等しいと推定される。

30

slice_cc_alf_cb_aps_idは、スライスのCb色成分が参照するadaptation_parameter_set_idを指定する。

aps_params_typeがALF_APSと等しく、adaptation_parameter_set_idがslice_cc_alf_cb_aps_idと等しいAPS NALユニットのTemporalIdは、コーディングスライスNALユニットのTemporalId以下であるものとする。slice_cc_alf_cb_enabled_flagが1に等しく、slice_cc_alf_cb_aps_idが存在しないとき、slice_cc_alf_cb_aps_idの値はph_cc_alf_cb_aps_idの値に等しいと推定される。

aps_params_typeがALF_APSに等しく、adaptation_parameter_set_idがslice_cc_alf_cb_aps_idに等しいAPS NALユニットのalf_cc_cb_filter_signal_flagの値は1に等しいものとする。

40

slice_cc_alf_cr_enabled_flagが0に等しいことは、成分間フィルタがCr色成分に適応されないことを指定する。slice_cc_alf_cr_enabled_flagが1に等しいことは、成分間適応ループフィルタが有効であり、Cr色成分に適応される可能性があることを指定する。slice_cc_alf_cr_enabled_flagが存在しない場合、それは、ph_cc_alf_cr_enabled_flagに等しいと推定される。

slice_cc_alf_cr_aps_idは、スライスのCr色成分が参照するadaptation_parameter_set_idを指定する。aps_params_typeがALF_APSと等しく、adaptation_parameter_set_idがslice_cc_alf_cr_aps_idと等しいAPS NALユニットのTemporalIdは、コーディングスライスNALユニットのTemporalId以下であるものとする。slice_cc_alf_cr_

50

enabled_flagが1に等しく、slice_cc_alf_cr_aps_idが存在しないとき、slice_cc_alf_cr_aps_idの値はph_cc_alf_cr_aps_idの値に等しいと推定される。

aps_params_typeがALF_APSに等しく、adaptation_parameter_set_idがslice_cc_alf_cr_aps_idに等しいAPS NALユニットのalf_cc_cr_filter_signal_flagの値は1に等しいものとする。

separate_colour_plane_flagが1に等しいとき、colour_plane_idは、現在スライスに関連付けられている色平面を識別する。colour_plane_idの値は、両端を含む0～2の範囲であるものとする。colour_plane_id値0、1、2は、各々Y、Cb、Cr平面に対応する。値3のcolour_plane_idは、ITU - T/ISO/IECによる将来の使用のために予約されている。

注1：1つのピクチャの異なる色平面の復号処理間に依存関係はない。

num_ref_idx_active_override_flagが1に等しいことは、シンタックス要素num_ref_idx_active_minus1[0]がP及びBスライスについて存在し、シンタックス要素num_ref_idx_active_minus1[1]がBスライスについて存在することを指定する。num_ref_idx_active_override_flagが0に等しいことは、シンタックス要素num_ref_idx_active_minus1[0]及びnum_ref_idx_active_minus1[1]が存在しないことを指定する。存在しないとき、num_ref_idx_active_override_flagの値は1に等しいと推定される。

num_ref_idx_active_minus1[i]は、式143で指定されている変数NumRefIdxActive[i]の導出に使用される。num_ref_idx_active_minus1[i]の値は、両端を含む0～14の範囲であるものとする。

0又は1に等しいiについて、現在スライスがBスライスであり、num_ref_idx_active_override_flagが1に等しく、num_ref_idx_active_minus1[i]が存在しないとき、num_ref_idx_active_minus1[i]は0に等しいと推定される。

現在スライスがPスライスであり、num_ref_idx_active_override_flagが1に等しく、num_ref_idx_active_minus1[i]が存在しないとき、num_ref_idx_active_minus1[i]は0に等しいと推定される。

変数NumRefIdxActive[i]は以下のように導出される：

【数49】

```

for(i = 0; i < 2; i++) {
    if( slice_type == B || (slice_type == P && i == 0) ) {
        if( num_ref_idx_active_override_flag )
            NumRefIdxActive[ i ] = num_ref_idx_active_minus1[ i ] + 1
        else {
            if(
                num_ref_entries[ i ][ RplIdx[ i ] ]
                num_ref_idx_default_active_minus1[ i ] + 1 )
                NumRefIdxActive[ i ] = num_ref_idx_default_active_minus1[ i ] + 1
            else
                NumRefIdxActive[ i ] = num_ref_entries[ i ][ RplIdx[ i ] ]
        }
    } else /* slice_type == I || (slice_type == P && i == 1) */
        NumRefIdxActive[ i ] = 0
}

```

NumRefIdxActive[i]-1の値は、スライスの復号に使用できる参照ピクチャリストiの最大参照インデックスを指定する。NumRefIdxActive[i]の値が0に等しい場合、参照ピクチャリストiの参照インデックスはスライスの復号に使用できない。

現在スライスがPスライスである場合、NumRefIdxActive[0]の値は0より大きいものとする。

現在スライスがBスライスである場合、NumRefIdxActive[0]及びNumRefIdxActive[1]の両方は0より大きいものとする。

cabac_init_flagは、コンテキスト変数の初期化処理で使用される初期化テーブルを決定する方法を指定する。cabac_init_flagが存在しない場合、0と等しいと推定される。

slice_collocated_from_l0_flagが1に等しいことは、時間的動きベクトル予測に使用される同一位置ピクチャが参照ピクチャリスト0から導出されることを指定する。slice_collocated_from_l0_flagが0に等しいことは、時間的動きベクトル予測に使用される同一位置ピクチャが参照ピクチャリスト1から導出されることを指定する。

slice_typeがB又はPに等しく、ph_temporal_mvp_enabled_flagが1に等しく、slice_collocated_from_l0_flagが存在しない場合、以下が適用される：

- rpl_info_in_ph_flagが1に等しい場合、slice_collocated_from_l0_flagはph_collocated_from_l0_flagに等しいと推定される、

- それ以外の場合（rpl_info_in_ph_flagが0に等しく、slice_typeがPに等しい）、slice_collocated_from_l0_flagの値は1に等しいと推定される。

slice_collocated_ref_idxは、時間的動きベクトル予測に使用される同一位置ピクチャの参照インデックスを指定する。

slice_typeがPに等しい場合、又はslice_typeがBに等しく、slice_collocated_from_l0_flagが1に等しい場合、slice_collocated_ref_idxは参照ピクチャリスト0のエントリを参照し、slice_collocated_ref_idxの値は、両端を含む0からNumRefIdxActive[0]-1までの範囲にあるものとする。

slice_typeがBに等しく、slice_collocated_from_l0_flagが0に等しい場合、slice_collocated_ref_idxは参照ピクチャリスト1のエントリを参照し、slice_collocated_ref_idxの値は、両端を含む0からNumRefIdxActive[1]-1までの範囲にあるものとする。

slice_collocated_ref_idxが存在しない場合、以下が適用される：

- rpl_info_in_ph_flagが1に等しい場合、slice_collocated_ref_idxの値は、ph_collocated_ref_idxに等しいと推定される、

- それ以外の場合（rpl_info_in_ph_flagが0に等しい）、slice_collocated_ref_idxの値は0に等しいと推定される。

slice_collocated_ref_idxによって参照されるピクチャが、コーディングピクチャのすべてのスライスに対して同じであることは、ビットストリーム準拠の要件である。

slice_collocated_ref_idxによって参照される参照ピクチャのpic_width_in_luma_samplesとpic_height_in_luma_samplesの値が、各々現在ピクチャのpic_width_in_luma_samplesとpic_height_in_luma_samplesの値と等しこと、及び、RprConstraintsActive[slice_collocated_from_l0_flag?0:1][slice_collocated_ref_idx]が0に等しいことは、ビットストリーム準拠の要件である。

slice_qp_deltaは、コードユニット層のCuQpDeltaValの値により変更されるまで、スライス内のコーディングブロックに使用されるQpYの初期値を指定する。

qp_delta_info_in_ph_flagが0に等しい場合、スライスに対するQpY量子化パラメータの初期値SliceQpYは次のように導出される：

【数50】

$$\text{SliceQpY} = 26 + \text{init_qp_minus26} + \text{slice_qp_delta} \quad (144)$$

SliceQpYの値は、両端を含む-QpBdOffset ~ +63の範囲であるものとする。

次の条件のいずれかが真である場合：

- wp_info_in_ph_flagの値が1に等しく、pps_weighted_pred_flagが1に等しく、slice_typeがPに等しい、

- wp_info_in_ph_flagの値が1に等しく、pps_weighted_bipred_flagが1に等しく

、slice_typeがBに等しい、

以下が適用される：

- NumRefIdxActive[0]の値は、NumWeightsL0の値以下であるものとする、
- 両端を含む0からNumRefIdxActive[0]-1までの範囲のiについて、各参照ピクチャインデックスRefPicList[0][i]について、参照ピクチャインデックスに適用されるルマ重み、Cb重み、及びCr重みは、各々LumaWeightL0[i]、ChromaWeightL0[0][i]、及びChromaWeightL0[1][i]である。

wp_info_in_ph_flagが1に等しく、pps_weighted_bipred_flagが1に等しく、slice_typeがBに等しい場合、以下が適用される：

- NumRefIdxActive[1]の値は、NumWeightsL1の値以下であるものとする、
- 両端を含む0からNumRefIdxActive[1]-1までの範囲のiについて、各参照ピクチャインデックスRefPicList[1][i]について、参照ピクチャインデックスに適用されるルマ重み、Cb重み、及びCr重みは、各々LumaWeightL1[i]、ChromaWeightL1[0][i]、及びChromaWeightL1[1][i]である。

slice_cb_qp_offsetは、Qp_{cb}量子化パラメータの値を決定するときにpps_cb_qp_offsetの値を差に加えることを指定する。slice_cb_qp_offsetの値は、両端を含む-12~+12の範囲であるものとする。slice_cb_qp_offsetが存在しない場合、0と等しいと推定される。pps_cb_qp_offset+slice_cb_qp_offsetの値は、両端を含む-12~+12の範囲であるものとする。

slice_cr_qp_offsetは、Qp_{cr}量子化パラメータの値を決定するときにpps_cr_qp_offsetの値を差に加えることを指定する。slice_cr_qp_offsetの値は、両端を含む-12~+12の範囲であるものとする。slice_cr_qp_offsetが存在しない場合、0と等しいと推定される。pps_cr_qp_offset+slice_cr_qp_offsetの値は、両端を含む-12~+12の範囲であるものとする。

slice_joint_cbcr_qp_offsetは、Qp_{cbcr}の値を決定するときにpps_joint_cbcr_qp_offset_valueの値を差に加えることを指定する。slice_joint_cbcr_qp_offsetの値は、両端を含む-12~+12の範囲であるものとする。slice_joint_cbcr_qp_offsetが存在しない場合、0と等しいと推定される。pps_joint_cbcr_qp_offset_value+slice_joint_cbcr_qp_offsetの値は、両端を含む-12~+12の範囲であるものとする。

cu_chroma_qp_offset_enabled_flagが1に等しいことは、変換ユニット及びパレットコーディングシンタックスにcu_chroma_qp_offset_flagが存在する可能性のあることを指定する。cu_chroma_qp_offset_enabled_flagが0に等しいことは、変換ユニット又はパレットコーディングシンタックスにcu_chroma_qp_offset_flagが存在しないことを指定する。存在しないとき、cu_chroma_qp_offset_enabled_flagの値は0に等しいと推定される。

slice_sao_luma_flagが1に等しいことは、現在スライスでルマ成分についてSAOが有効にされることを指定する。slice_sao_luma_flagが0に等しいことは、現在スライスでルマ成分についてSAOが無効にされることを指定する。slice_sao_luma_flagが存在しない場合、それは、ph_sao_luma_enabled_flagに等しいと推定される。

slice_sao_chroma_flagが1に等しいことは、現在スライスでクロマ成分についてSAOが有効にされることを指定する。slice_sao_chroma_flagが0に等しいことは、現在スライスでクロマ成分についてSAOが無効にされることを指定する。slice_sao_luma_flagが存在しない場合、それは、ph_sao_luma_enabled_flagに等しいと推定される。

slice_deblocking_filter_override_flagが1に等しいことは、デブロッキングパラメータがスライスヘッダに存在することを指定する。slice_deblocking_filter_override_flagが0に等しいことは、デブロッキングパラメータがスライスヘッダに存在しないことを指定する。存在しないとき、slice_deblocking_filter_override_flagの値は、ph_deblocking_filter_override_flagに等しいと推定される。

slice_deblocking_filter_disabled_flagが1に等しいことはデブロッキングフィルタの操作が、現在スライスに適用されないことを指定する。slice_deblocking_filter_disa

10

20

30

40

50

bled_flagが0に等しいことはデブロッキングフィルタの操作が、現在スライスに適用されることを指定する。slice_deblocking_filter_disabled_flagが存在しない場合、それは、ph_deblocking_filter_disabled_flagに等しいと推定される。

slice_beta_offset_div2及びslice_tc_offset_div2は、現在スライスについて、ルマ成分に適用される及びtC(2で除算されたもの)のデブロッキングパラメータオフセットを指定する。slice_beta_offset_div2とslice_tc_offset_div2の値は、両方とも両端を含む-12から12の範囲内にあるものとする。存在しない場合、slice_beta_offset_div2とslice_tc_offset_div2の値は、各々ph_beta_offset_div2とph_tc_offset_div2に等しいと推定される。

slice_cb_beta_offset_div2及びslice_cb_tc_offset_div2は、現在スライスについて、Cb成分に適用される及びtC(2で除算されたもの)のデブロッキングパラメータオフセットを指定する。slice_cb_beta_offset_div2とslice_cb_tc_offset_div2の値は、両方とも両端を含む-12から12の範囲内にあるものとする。存在しない場合、slice_cb_beta_offset_div2とslice_cb_tc_offset_div2の値は、各々ph_cb_beta_offset_div2とph_cb_tc_offset_div2に等しいと推定される。

10

slice_cr_beta_offset_div2及びslice_cr_tc_offset_div2は、現在スライスについて、Cr成分に適用される及びtC(2で除算されたもの)のデブロッキングパラメータオフセットを指定する。slice_cr_beta_offset_div2とslice_cr_tc_offset_div2の値は、両方とも両端を含む-12から12の範囲内にあるものとする。存在しない場合、slice_cr_beta_offset_div2とslice_cr_tc_offset_div2の値は、各々ph_cr_beta_offset_div2とph_cr_tc_offset_div2に等しいと推定される。

20

slice_ts_residual_coding_disabled_flagが1に等しいことは、現在スライスの変換スキップブロックの残差サンプルを解析するために、residual_coding()シンタックス構造が使用されることを指定する。slice_ts_residual_coding_disabled_flagが0に等しいことは、現在スライスの変換スキップブロックの残差サンプルを解析するために、residual_ts_coding()シンタックス構造が使用されることを指定する。slice_ts_residual_coding_disabled_flagが存在しない場合、0と等しいと推定される。

slice_lmcs_enabled_flagが1に等しいことは、現在スライスに対してクロマスケールによるルママッピングが有効であることを指定する。slice_lmcs_enabled_flagが0に等しいことは、現在スライスに対してクロマスケールによるルママッピングが有効ではないことを指定する。slice_lmcs_enabled_flagが存在しない場合、0と等しいと推定される。

30

slice_scaling_list_present_flagが1に等しいことは、現在スライスに使用されるスケーリングリストデータが、aps_params_typeがSCALING_APSに等しくadaptation_parameter_set_idがph_scaling_list_aps_idに等しい参照スケーリングリストAPSに含まれるスケーリングリストデータに基づいて導出されることを指定する。slice_scaling_list_present_flagが0に等しいことは、現在ピクチャに使用されるスケーリングリストデータが7.4.3.2.1項で指定された導出されたデフォルトのスケーリングリストデータであることを指定する。存在しないとき、slice_scaling_list_present_flagの値は0に等しいと推定される。

40

現在スライス内のエントリポイントの数を指定する変数NumEntryPointsは、次のように導出される：

【数51】

```

NumEntryPoints = 0
for( i = 1; i < NumCtUsInCurrSlice; i++ ) {
    ctbAddrX = CtbAddrInCurrSlice[ i ] % PicWidthInCtbsY
    ctbAddrY = CtbAddrInCurrSlice[ i ] / PicWidthInCtbsY           (145)
    prevCtbAddrX = CtbAddrInCurrSlice[ i - 1 ] % PicWidthInCtbsY
    prevCtbAddrY = CtbAddrInCurrSlice[ i - 1 ] / PicWidthInCtbsY
    if( CtbToTileRowBd[ ctbAddrY ] != CtbToTileRowBd[ prevCtbAddrY ] ||
        CtbToTileColBd[ ctbAddrX ] != CtbToTileColBd[ prevCtbAddrX ] ||
        ( ctbAddrY != prevCtbAddrY &&
sps_wpp_entry_point_offsets_present_flag ))
        NumEntryPoints++
}

```

offset_len_minus 1 に 1 を加えたものは、entry_point_offset_minus 1 [i] シンタックス要素の長さをビット単位で指定する。offset_len_minus 1 の値は、両端を含む 0 ~ 31 の範囲であるものとする。

entry_point_offset_minus 1 [i] に 1 を加えたものは、i 番目のエントリポイントのオフセットをバイト単位で指定し、offset_len_minus 1 + 1 ビットで表される。スライスヘッダに続くスライスデータは、NumEntryPoints + 1 個のサブセットで構成され、サブセットインデックス値の範囲は、両端を含む 0 から NumEntryPoints までである。スライスデータの第 1 バイトはバイト 0 と見なされる。存在する場合、コーディングスライスNALユニットのスライスデータ部分に出現するエミュレーション防止バイトは、サブセット識別のためにスライスデータの一部としてカウントされる。サブセット 0 は、コーディングスライスデータの両端を含むバイト 0 から entry_point_offset_minus 1 [0] までで構成される。サブセット k は、両端を含む 1 から NumEntryPoints - 1 までの範囲の k により、コーディングスライスの両端を含むバイト firstByte[k] から lastByte[k] までのバイトで構成され、firstByte[k] と lastByte[k] は以下のように定義される：

【数 5 2】

$$\text{firstByte}[k] = \sum_{n=1}^k (\text{entry_point_offset_minus1}[n-1] + 1) \quad (146)$$

$$\text{lastByte}[k] = \text{firstByte}[k] + \text{entry_point_offset_minus1}[k] \quad (147)$$

最後のサブセット（サブセットインデックスが NumEntryPoints と等しい）は、コーディングスライスデータの残りのバイトで構成される。

sps_entropy_coding_sync_enabled_flag が 0 に等しく、スライスに 1 つ以上の完全なタイルが含まれている場合、各サブセットは、同じタイル内にあるスライス内のすべての CTU のすべてのコーディングされたビットで構成され、サブセットの数（つまり、NumEntryPoints の値 + 1）は、スライス内のタイルの数と等しいものとする。

sps_entropy_coding_sync_enabled_flag が 0 に等しく、スライスに 1 つのタイルからの CTU 行のサブセットが含まれている場合、NumEntryPoints は 0 になり、サブセットの数は 1 になるものとする。サブセットは、スライス内のすべての CTU のすべてのコーディングビットで構成されるものとする。

sps_entropy_coding_sync_enabled_flag が 1 に等しい場合、各サブセット k は、両端を含む 0 から NumEntryPoints までの範囲の k について、タイル内にある CTU 行のすべての CTU のコーディングビットで構成され、サブセットの数（つまり、NumEntryPoints の値 + 1）は、スライス内のタイル固有 CTU 行の合計数と等しいものとする。

slice_header_extension_lengthは、スライスヘッダ拡張データの長さをバイト単位で指定し、slice_header_extension_length自体のシグナリングに使用されるビットは含まれない。slice_header_extension_lengthの値は、両端を含む0 ~ 256の範囲であるものとする。存在しないとき、slice_header_extension_lengthの値は0に等しいと推定される。

slice_header_extension_data_byte[i]は任意の値を持つことができる。このバージョンの仕様に準拠するデコーダは、全てのgci_reserved_constraint_byte[i]シンタックス要素を無視する必要がある。その値は、この仕様のこのバージョン内で指定されるプロファイルへのデコーダ準拠性に影響しない。

【0042】

3.7 変換ユニットシンタックス (スライスデータ)

最新のVVCドラフトテキストでは、変換ユニットシンタックスとセマンティクスは次の通りである：

【表14】

transform_unit(x0, y0, tbWidth, tbHeight, treeType, subTuIndex, chType) {	Descriptor
if(IntraSubPartitionsSplitType != ISP_NO_SPLIT && treeType == SINGLE_TREE && subTuIndex == NumIntraSubPartitions - 1) {	
xC = CbPosX[chType][x0][y0]	
yC = CbPosY[chType][x0][y0]	
wC = CbWidth[chType][x0][y0] / SubWidthC	
hC = CbHeight[chType][x0][y0] / SubHeightC	
} else {	
xC = x0	
yC = y0	
wC = tbWidth / SubWidthC	
hC = tbHeight / SubHeightC	
}	
chromaAvailable = treeType != DUAL_TREE_LUMA && ChromaArrayType != 0 && (IntraSubPartitionsSplitType == ISP_NO_SPLIT (IntraSubPartitionsSplitType != ISP_NO_SPLIT && subTuIndex == NumIntraSubPartitions - 1))	
if((treeType == SINGLE_TREE treeType == DUAL_TREE_CHROMA) && ChromaArrayType != 0 && (IntraSubPartitionsSplitType == ISP_NO_SPLIT && ((subTuIndex == 0 && cu_sbt_pos_flag) (subTuIndex == 1 && !cu_sbt_pos_flag))) (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex == NumIntraSubPartitions - 1))) {	

10

20

30

40

50

tu_cbf_cb [xC][yC]	ae(v)
tu_cbf_cr [xC][yC]	ae(v)
}	
if(treeType == SINGLE_TREE treeType == DUAL_TREE_LUMA) {	
if((IntraSubPartitionsSplitType == ISP_NO_SPLIT && !(cu_sbt_flag && ((subTuIndex == 0 && cu_sbt_pos_flag) (subTuIndex == 1 && !cu_sbt_pos_flag))) && (CuPredMode[chType][x0][y0] == MODE_INTRA (chromaAvailable && (tu_cbf_cb[xC][yC] tu_cbf_cr[xC][yC]))) CbWidth[chType][x0][y0] > MaxTbSizeY CbHeight[chType][x0][y0] > MaxTbSizeY)) (IntraSubPartitionsSplitType != ISP_NO_SPLIT && (subTuIndex < NumIntraSubPartitions - 1 !InferTuCbfLuma)))	10
tu_cbf_luma [x0][y0]	ae(v)
if(IntraSubPartitionsSplitType != ISP_NO_SPLIT)	
InferTuCbfLuma = InferTuCbfLuma && !tu_cbf_luma[x0][y0]	
}	20
if((CbWidth[chType][x0][y0] > 64 CbHeight[chType][x0][y0] > 64 tu_cbf_luma[x0][y0] (chromaAvailable && (tu_cbf_cb[xC][yC] tu_cbf_cr[xC][yC]))) && treeType != DUAL_TREE_CHROMA && cu_qp_delta_enabled_flag && !IsCuQpDeltaCoded) {	
cu_qp_delta_abs	ae(v)
if(cu_qp_delta_abs)	
cu_qp_delta_sign_flag	ae(v)
}	30
if((CbWidth[chType][x0][y0] > 64 CbHeight[chType][x0][y0] > 64 (chromaAvailable && (tu_cbf_cb[xC][yC] tu_cbf_cr[xC][yC]))) && treeType != DUAL_TREE_LUMA && cu_chroma_qp_offset_enabled_flag && !IsCuChromaQpOffsetCoded) {	
cu_chroma_qp_offset_flag	ae(v)
if(cu_chroma_qp_offset_flag && chroma_qp_offset_list_len_minus1 > 0)	
cu_chroma_qp_offset_idx	ae(v)
}	40

10

20

30

40

50

<pre> if(sps_joint_cbr_enabled_flag && ((CuPredMode[chType][x0][y0] == MODE_INTRA && (tu_cbf_cb[xC][yC] tu_cbf_cr[xC][yC])) (tu_cbf_cb[xC][yC] && tu_cbf_cr[xC][yC])) && chromaAvailable) </pre>		
tu_joint_cbr_residual_flag[xC][yC]		ae(v)
<pre> if(tu_cbf_luma[x0][y0] && treeType != DUAL_TREE_CHROMA) { </pre>		
<pre> if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][0] && tbWidth <= MaxTsSize && tbHeight <= MaxTsSize && (IntraSubPartitionsSplitType == ISP_NO_SPLIT) && !cu_sbt_flag) </pre>		10
transform_skip_flag[x0][y0][0]		ae(v)
<pre> if(!transform_skip_flag[x0][y0][0] slice_ts_residual_coding_disabled_flag) </pre>		
<pre> residual_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0) </pre>		
else		
<pre> residual_ts_coding(x0, y0, Log2(tbWidth), Log2(tbHeight), 0) </pre>		
}		
<pre> if(tu_cbf_cb[xC][yC] && treeType != DUAL_TREE_LUMA) { </pre>		
<pre> if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][1] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag) </pre>		20
transform_skip_flag[xC][yC][1]		ae(v)
<pre> if(!transform_skip_flag[xC][yC][1] slice_ts_residual_coding_disabled_flag) </pre>		
<pre> residual_coding(xC, yC, Log2(wC), Log2(hC), 1) </pre>		
else		
<pre> residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 1) </pre>		
}		
<pre> if(tu_cbf_cr[xC][yC] && treeType != DUAL_TREE_LUMA && !(tu_cbf_cb[xC][yC] && tu_joint_cbr_residual_flag[xC][yC])) { </pre>		30
<pre> if(sps_transform_skip_enabled_flag && !BdpcmFlag[x0][y0][2] && wC <= MaxTsSize && hC <= MaxTsSize && !cu_sbt_flag) </pre>		
transform_skip_flag[xC][yC][2]		ae(v)
<pre> if(!transform_skip_flag[xC][yC][2] slice_ts_residual_coding_disabled_flag) </pre>		
<pre> residual_coding(xC, yC, Log2(wC), Log2(hC), 2) </pre>		
else		
<pre> residual_ts_coding(xC, yC, Log2(wC), Log2(hC), 2) </pre>		
}		
}		40

変換係数レベルは、アレイTransCoeffLevel[x0][y0][cldx][xC][yC]により表される。アレイインデックスx0,y0は、ピクチャの左上ルマサンプルに対する検討される変換ブロックの左上サンプルの位置(x0,y0)を指定する。アレイインデックスcldxは色成分の指示子を指定し、Yの場合は0、Cbの場合は1、Crの場合は2に等しい。アレイインデックスxCとyCは、現在の変換ブロック内の変換係数の位置(xC,yC)を指定する。TransCoeffLevel[x0][y0][cldx][xC][yC]の値が7.3.10.11項で指定されていない場合、0と等しいと推定される。

tu_cbf_cr[x0][y0]が1に等しいことは、Cb変換ブロックが0に等しくない変換係数レベルが1つ以上含まれていることを指定する。アレイインデックスx0,y0は、対象と

なる変換ブロックの左上に位置する。

tu_cbf_cb[x0][y0]が存在しない場合、0と等しいと推定される。

tu_cbf_cr[x0][y0]が1に等しいことは、Cr変換ブロックが0に等しくない変換係数レベルが1つ以上含まれていることを指定する。アレイインデックスx0、y0は、対象となる変換ブロックの左上の位置(x0,y0)を指定する。

tu_cbf_cr[x0][y0]が存在しない場合、0と等しいと推定される。

tu_cbf_luma[x0][y0]が1に等しいことは、ルマ変換ブロックが0に等しくない1つ以上の変換係数レベルを含むことを指定する。アレイインデックスx0,y0は、ピクチャの左上ルマサンプルに対する検討される変換ブロックの左上サンプルの位置(x0,y0)を指定する。

10

tu_cbf_luma[x0][y0]が存在しない場合、それは、以下のように推定される：

cu_sbt_flagが1に等しく、次の条件のうちの1つが真である場合、tu_cbf_luma[x0][y0]は0に等しいと推定される：

- subTuIndexが0に等しく、cu_sbt_pos_flagが1に等しい、
- subTuIndexが1に等しく、cu_sbt_pos_flagが0に等しい、
- それ以外の場合、treeTypeがDUAL_TREE_CHROMAに等しい場合、tu_cbf_luma[x0][y0]は0に等しいと推定される、

- それ以外の場合、tu_cbf_luma[x0][y0]は1に等しいと推定される。

tu_joint_cbcr_residual_flag[x0][y0]は、クロマ成分CbとCrの両方の残差サンプルが1つの変換ブロックとしてコーディングされるかどうかを指定する。アレイインデックスx0,y0は、ピクチャの左上ルマサンプルに対する検討される変換ブロックの左上サンプルの位置(x0,y0)を指定する。

20

tu_joint_cbcr_residual_flag[x0][y0]が1に等しいことは、変換ユニットシンタックスが、CbとCrの両方の残差サンプルが導出される単一の変換ブロックの変換係数レベルを含むことを指定する。tu_joint_cbcr_residual_flag[x0][y0]が0に等しいことは、クロマ成分の変換係数レベルがシンタックス要素tu_cbf_cb[x0][y0]及びtu_cbf_cr[x0][y0]で示されるようにコーディングされることを指定する。

tu_joint_cbcr_residual_flag[x0][y0]が存在しない場合、0と等しいと推定される。

tu_joint_cbcr_residual_flag[x0][y0]、tu_cbf_cb[x0][y0]、及びtu_cbf_cr[x0][y0]に応じて、変数TuCResMode[x0][y0]は次のように導出される：

30

- tu_joint_cbcr_residual_flag[x0][y0]が0に等しい場合、変数TuCResMode[x0][y0]は0に等しく設定される、

- その他の場合、[x0任意で、y2が1に等しい[x0][y0]が0に等しい場合、変数TuCResMode[x0][y0]は0に等しく設定される、

- それ以外の場合、tu_cbf_cb[x0][y0]が1に等しい場合、変数TuCResMode[x0][y0]は2に等しく設定される、

- その他の場合、変数TuCResMode[x0][y0]は3に等しく設定される。

cu_qp_delta_absは、現在コーディングユニットの量子化パラメータとその予測との差CuQpDeltaValの絶対値を指定する。

40

cu_qp_delta_sign_flagは、次のようにCuQpDeltaValの符号を指定する：

- cu_qp_delta_sign_flagが0に等しい場合、対応するCuQpDeltaValは正の値を持つ、

- その他の場合(cu_qp_delta_sign_flagが1に等しい)、対応するCuQpDeltaVal負の値を持つ。

cu_qp_delta_sign_flagが存在しない場合、0と等しいと推定される。

cu_qp_delta_absが存在する場合、変数IsCuQpDeltaCodedとCuQpDeltaValは次のように導出される：

【数53】

50

$$\text{IsCuQpDeltaCoded} = 1 \quad (187)$$

$$\text{CuQpDeltaVal} = \text{cu_qp_delta_abs} * (1 - 2 * \text{cu_qp_delta_sign_flag}) \quad (188)$$

CuQpDeltaValの値は、両端を含む $-(3 2 + \text{QpBdOffset} / 2)$ から $+(3 1 + \text{QpBdOffset} / 2)$ の範囲であるものとする。

cu_chroma_qp_offset_flagが存在し、1に等しい場合、cb_qp_offset_list[]内のエントリがCuQpOffset_{cb}の値を決定するために使用され、cr_qp_offset_list[]内の対応するエントリがCuQpOffset_{cr}の値を決定するために使用され、joint_cbr_qp_offset_list[]内の対応するエントリがCuQpOffset_{cbcr}の値を決定するために使用されることを指定する。cu_chroma_qp_offset_flagが0に等しいことは、これらのリストがCuQpOffset_{cb}、CuQpOffset_{cr}、及びCuQpOffset_{cbcr}の値を決定するために使用されないことを指定する。

cu_chroma_qp_offset_idxが存在する場合は、CuQpOffset_{cb}、CuQpOffset_{cr}、及びCuQpOffset_{cbcr}の値を決定するために使用されるcb_qp_offset_list[]、cr_qp_offset_list[]、及びjoint_cbr_qp_offset_list[]へのインデックスを指定する。存在する場合、cu_chroma_qp_offset_idxの値は、両端を含む0からchroma_qp_offset_list_len_minus_1までの範囲にあるものとする。存在しないとき、cu_chroma_qp_offset_idxの値は0に等しいと推定される。

cu_chroma_qp_offset_flagが存在する場合、以下が適用される：

- 変数IsCuChromaQpOffsetCodedは、1に等しく設定される、
- 変数CuQpOffset_{cb}、CuQpOffset_{cr}、及びCuQpOffset_{cbcr}は、次のように導出される：

- cu_chroma_qp_offset_flagが1に等しい場合、以下が適用される：

【数54】

$$\text{CuQpOffset}_{cb} = \text{cb_qp_offset_list}[\text{cu_chroma_qp_offset_idx}] \quad (189)$$

$$\text{CuQpOffset}_{cr} = \text{cr_qp_offset_list}[\text{cu_chroma_qp_offset_idx}] \quad (190)$$

$$\text{CuQpOffset}_{cbcr} = \text{joint_cbr_qp_offset_list}[\text{cu_chroma_qp_offset_idx}] \quad (191)$$

- それ以外の場合 (cu_chroma_qp_offset_flagが0に等しい)、CuQpOffset_{cb}、CuQpOffset_{cr}、及びCuQpOffset_{cbcr}はすべて0に設定される。

transform_skip_flag[x0][y0][cldx]は、変換が関連付けられた変換ブロックに適用されるかどうかを指定する。アレイインデックスx0,y0は、ピクチャの左上マサンプルに対する検討される変換ブロックの左上サンプルの位置(x0,y0)を指定する。アレイインデックスcldxは色成分の指示子を指定し、Yの場合は0、Cbの場合は1、Crの場合は2に等しい。transform_skip_flag[x0][y0][cldx]が1に等しいことは、変換が関連付けられた変換ブロックに適用されないことを指定する。transform_skip_flag[x0][y0][cldx]が0に等しいことは、変換が関連付けられた変換ブロックに適用されるかどうかの決定が他のシンタックス要素に依存することを指定する。

transform_skip_flag[x0][y0][cldx]が存在しない場合、それは、以下のように推定される：

- BdpcmFlag[x0][y0][cldx]が1に等しい場合、transform_skip_flag[x0][y0][cldx]は1に等しいと推定される、

- それ以外の場合 (BdpcmFlag[x0][y0][cldx]が0に等しい)、transform_skip_flag[x0][y0][cldx]は0に等しいと推定される。

【 0 0 4 3 】

4 . 開示の実施形態による技術的問題の例

SH、PPS、APSシンタックス要素（SE）の既存の設計には、次の問題がある：

1) 最新のVVCドラフトテキストでは、SCALINGAPSでシグナルされるスケーリング/量子化マトリクス（QM）の数を制御するために、APSシンタックス要素scaling_list_chroma_present_flagがシグナリングされる。つまり、scaling_list_chroma_present_flagが1に等しい場合、28個のQMがルマとクロマの両方に対してシグナルされる。それ以外の場合（scaling_list_chroma_present_flagが0に等しい）、10個のQMがルマ専用でシグナリングされる。現在、scaling_list_chroma_present_flagの値は、（SPSシンタックス要素から導出された）ChromaArrayTypeに基づいて制限されている。つまり、ChromaArrayTypeが0に等しくない場合はscaling_list_chroma_present_flagの値が1に等しい必要があり、一方、ChromaArrayTypeが0に等しい場合はscaling_list_chroma_present_flagの値が0に等しい必要がある。セマンティクスのこのような制約は、APSのSPSへの依存性をもたらすが、これは発生しないはずである。なぜなら、APSは異なるSPSを参照するピクチャ（又はピクチャのスライス）に適用される可能性があり、それは異なるChromaArrayType値に関連付けられる可能性があるからである。

a . 更に、現在のところ、ブロックがユーザ定義のスケーリングリストでコーディングされると、ルマとクロマの両方（使用可能な場合）がユーザ定義のスケーリングリストを適用する必要がある。つまり、ルマとクロマのユーザ定義のスケーリングリストを個別にオン/オフすることはできない。このような設計は効率的/柔軟性に欠ける場合がある。

2) 最新のVVCドラフトテキストでは、LMCS APSシンタックス構造がシグナリングされるとき、ChromaArrayTypeが0（例えば、ビデオコンテンツにクロマ成分がない）に等しいかどうかに関係なく、クロマ残差スケーリングに関連するAPSシンタックス要素が常にシグナリングされる。これにより、ビデオコンテンツに処理されるクロマが存在しないのに、クロマ関連のシンタックス要素が不要に送信される可能性がある。

3) 最新のVVCドラフトテキストでは、SHシンタックス要素slice_ts_residual_coding_disabled_flagを使用して、変換スキップに基づく残差コーディング（transform skip based residual coding : TSRC）と通常残差コーディング（regular residual coding : RRC）のどちらを変換スキップブロックに使用するかを指定する。ただし、現時点では、SPSレベル変換スキップの無効化に関係なく、SHでは常にslice_ts_residual_coding_disabled_flagがシグナリングされる。sps_transform_skip_enabled_flagの場合、transform_skip_flagは常に0であり、TSRCとRRCを切り換える条件（!transform_skip_flag[xC][yC][1] || slice_ts_residual_coding_disabled_flag）が常に真であるため、このような場合、slice_ts_residual_coding_disabled_flagは無意味になる。

a . 更に、現在、非TSブロックはRRCしか使用できず、TSRCとRRCを切り換えることができないため、非TSブロック圧縮には効率的ではない可能性がある。

4) 最新のVVCドラフトテキストでは、ルマブロックのcu_qp_deltaを有効にするためにマルチレベル制御が使用されている。つまり、最初にPPSオン/オフ制御フラグcu_qp_delta_enabled_flagをシグナリングし、次にPHで量子化グループ（QG）サイズを指定し、最後に各QGでcu_qp_delta_absの値をシグナリングする。このように設計することで、ピクチャが複数のスライスで構成されている場合、一部のスライスではcu_qp_deltaが使用されているが、他のスライスではまったく使用されていない場合でも、依然としてすべてのQGに対してブロックレベルcu_qp_delta_absをシグナリングする必要がある。そのため、ブロックレベルのビットの無駄が発生するが、これは回避できる。

5) 最新のVVCドラフトテキストでは、PPSシンタックス要素single_slice_per_subpic_flagが0に等しい場合、PPSを参照するピクチャの各サブピクチャは1つ以上の長方形スライスで構成される場合がある。single_slice_per_subpic_flagが0の場合、そのようなPPSを参照するピクチャでは、以下のようなケースが発生する可能性がある：

a . 冗長なケース : sps_num_subpics_minus_1 が0より大きい、各サブピクチャに1つのスライスしかない場合。このような場合、各ピクチャには複数のサブピクチャと複

10

20

30

40

50

数の長方形スライスが含まれるが、single_slice_per_subpic_flagは0に等しいため、num_slices_in_pic_minus_1をシグナリングする必要がある。ただし、このようなケースは、single_slice_per_subpic_flagが1に等しいことと概念的に同じであり、このSEを通知する必要はまったくないため、冗長にシグナリングされる。

b. 冗長なケース：sps_num_subpics_minus_1が0に等しく、PPSを参照する各ピクチャに1つのスライスしかない場合。このような場合、各ピクチャには1つのスライスのみで構成される1つのサブピクチャが含まれるが、single_slice_per_subpic_flagは0に等しいことが引き続き許可されるため、num_slices_in_pic_minus_1をシグナリングする必要がある。ただし、このようなケースは、single_slice_per_subpic_flagが1に等しいことと概念的に同じであり、このSEを通知する必要はまったくないため、冗長にシグナリングされる。

10

c. 更に、上記の冗長なケースをすべて禁止/回避した場合、ピクチャが複数のサブピクチャを有する場合（各サブピクチャに1つのスライス又は複数のスライスが含まれる）、又はピクチャが複数のスライスを有する場合（各ピクチャに1つのサブピクチャ又は複数のサブピクチャが含まれる）には、常に0と等しいsingle_slice_per_subpic_flagが使用されることが分かる。どちらの場合も、num_slices_in_pic_minus_1の値は常に1より大きくなる。また、条件付きでPPSシンタックス要素tile_idx_delta_present_flagをシグナリングする必要がないことも判明した。

6) 最新のVVCドラフトテキストでは、タイルの幅や高さなどのタイルレイアウトは、暗黙の推定に関連する明示的なシグナリングの方法で設計されている。ピクチャが同じ高さのタイルを持つ複数のタイル行に分割されている場合、現在の設計では、第1タイル行の高さをシグナリングするだけで、残りのタイル行の高さを推定できる。それ以外の場合、ピクチャが高さの異なるタイルを持つ複数のタイル行に分割されている場合、各タイル行の高さが明示的にシグナリングされる。そうしないと、ピクチャが異なる高さの最初の幾つかのタイル行と同じ高さの最後の幾つかのタイル行を持つ複数のタイル行に分割された場合、最初の幾つかのタイル行の高さと最後の幾つかのタイル行のうちの一つだけを明示的にシグナリングし、その後、同じ高さの残りのタイル行の高さをシグナリングせずに推定する。現在の設計では、明示的なシグナリングと暗黙的な推定を組み合わせることで、これら3つのケースについて、うまく機能する。しかし、同じ高さの最初の幾つかのタイル行と異なる高さの最後に幾つかのタイル行がある複数のタイル行にピクチャを分割する別のケースもある。このような場合、暗黙の推定を適用することはできず、すべてのタイル行の高さを明示的にシグナリングする必要があるため、現在の設計はそれほど効率的ではないように思われる。同様に、タイル列シグナリングと長方形スライスレイアウトシグナリング、つまりスライスがタイルより小さい場合のスライス高さシグナリングにも同じ状況がある。ここで修正を適用して改善できる。

20

30

7) 現在、GCIシンタックス要素no_aps_constraint_flagは、nuh_unit_typeがPREFIX_APS_NUT又はSUFFIX_APS_NUTと等しいNALユニットを無効にするために使用される。ALF_APSが存在しないかどうかについては、ドラフトテキストで更に多くの制約が対処される予定である。

8) 現在、適合ウィンドウパラメータは、ピクチャの幅と高さが、PPSによって参照されるSPSでシグナリングされる最大ピクチャ幅と高さと同じ場合を含め、常にPPSでシグナリングされるが、一方で、最大ピクチャ幅と高さを持つピクチャの適合ウィンドウパラメータもSPSでシグナリングされる。PPS内の最大ピクチャ幅と高さを持つピクチャの適合ウィンドウパラメータのシグナリングは冗長である。

40

9) no_APS_constraint_flagが1に等しい場合、その意図は、ビットストリームに存在するか（帯域内（in-band）としても知られる）又は外部手段を通じて（例えば、実装ではAPIを通じて、帯域外（out-of-band）を通じて、等）デコーダに提供されるAPS_NALユニットの使用を無効にすることであった。ただし、現在のVVC仕様では、no_APS_constraint_flagのセマンティクスは、ビットストリームにAPS_NALユニットが存在する場合にのみ適用される。したがって、no_APS_constraint_flagが1に等しい場合でも

50

、APSは外部手段を通じて依然として使用できる可能性がある。

10) 現在、ほとんどのコーディングツール/機能はGCIフラグによって制御/非活性化することができる。しかし、幾つかの機能/コーディングツールはまだGCIフラグ制御を持っていない。可能な限り機能/コーディングツールの非活性化制御が望ましいかもしれない。

11) 現在、ローカルデュアルツリーを使用する場合、MODE_TYPE_INTRAに使用できるのはイントラ及びIBCコーディングモードのみであると定義されている。更に、intra (MODE_INTRA)、IBC (MODE_IBC) を定義し、インターコーディングモードをMODE_TYPE_ALLに使用することができる。ただし、MODE_TYPE_INTRA及びMODE_TYPE_ALLにはパレットモード (MODE_PLT) も使用される必要がある。現在の定義は不完全であると主張されている。

10

12) コーディングツリーノードの場合、コーディングツリーノード内のCUに許可される予測モードは、導出されたmodeTypeConditionと、modeTypeConditionが2に等しい場合にはシグナリングされたmode_constraint_flagによって定義される。ただし、modeTypeConditionが2に等しく、mode_constraint_flagが1に等しい場合、INTRAとIBCに等しい予測モードのみが許可され、一方、modeTypeConditionが2に等しく、mode_constraint_flagが0に等しい場合、INTERと等しい予測モードのみが許可されることに注意する。そのため、この場合は常にPALETTE MODEは許可されず、望ましくない。

20

30

40

50

【表 1 5】

coding_tree(x0, y0, cbWidth, cbHeight, qgOnY, qgOnC, cbSubdiv, cqtDepth, mttDepth, depthOffset, partIdx, treeTypeCurr, modeTypeCurr) {	Descriptor
...	
if(modeTypeCondition == 1)	
modeType = MODE_TYPE_INTRA	
else if(modeTypeCondition == 2) {	
mode_constraint_flag	ae(v)
modeType = mode_constraint_flag ? MODE_TYPE_INTRA : MODE_TYPE_INTER	
} else	
modeType = modeTypeCurr	
treeType = (modeType == MODE_TYPE_INTRA) ? DUAL_TREE_LUMA : treeTypeCurr	

10

変数 modeTypeCondition は、以下のように導出される：

以下の条件のうちの1つ以上が真である場合、modeTypeConditionは0に等しく設定される：

- sh_slice_typeがIに等しく、sps_qtbt_dual_tree_intra_flagが1に等しい、
- modeTypeCurrはMODE_TYPE_ALLに等しくない、
- sps_chroma_format_idcが0に等しい、
- sps_chroma_format_idcが3に等しい、

20

-その他の場合、以下の条件のうちのいずれか1つが真である場合、modeTypeConditionは1に等しく設定される：

- cbWidth*cbHeightは64に等しく、split_qt_flagは1に等しい、
- cbWidth*cbHeightは64に等しく、MttSplitMode[x 0][y 0][mttDepth]はSPLIT_TT_HOR又はSPLIT_TT_VERに等しい、
- cbWidth*cbHeightは32に等しく、MttSplitMode[x 0][y 0][mttDepth]はSPLIT_BT_HOR又はSPLIT_BT_VERに等しい、

30

その他の場合、以下の条件のうちのいずれかがTUREである場合、modeTypeConditionは1 + (sh_slice_type!=I?1:0) に等しく設定される：

- cbWidth*cbHeightは64に等しく、MttSplitMode[x 0][y 0][mttDepth]はSPLIT_BT_HOR又はSPLIT_BT_VERに等しく、sps_chroma_format_idcは1に等しい、
- cbWidth*cbHeightは128に等しく、MttSplitMode[x 0][y 0][mttDepth]はSPLIT_TT_HOR又はSPLIT_TT_VERに等しく、sps_chroma_format_idcは1に等しい、
- cbWidthは8に等しく、MttSplitMode[x 0][y 0][mttDepth]はSPLIT_BT_VERに等しい、
- cbWidthは16に等しく、MttSplitMode[x 0][y 0][mttDepth]はSPLIT_TT_VERに等しい、

40

-その他の場合、modeTypeConditionが0に等しく設定される。

mode_constraint_flag が0の場合、現在のコーディングツリーノード内のコーディングユニットはインター予測コーディングモードのみを使用できることを指定する。mode_constraint_flag が1の場合、現在のコーディングツリーノード内のコーディングユニットはインター予測コーディングモードを使用できないことを指定する。

5 . 実施形態及び技術の例示的な列挙

上記の問題点や、言及されない幾つかの他の問題点を解決するために、以下に要約する方法が開示される。発明は、一般的な概念を説明するための例として考えられるべきであり、狭義に解釈するべきではない。更に、これらの発明は任意の方法で個別に適用でき又は結合できる。

【 0 0 4 5 】

以下の説明では、最新のワーキングドラフトJVET - Q 2 0 0 1 - vDに基づく保護的なテキストの変更について、削除された部分は左と右の二重角括弧（例えば、[[[]]]）で強調表示され、二重角括弧の間に削除されたテキストが表示され、追加された部分は太字の斜体で表示される。

10

1 . 第 1 の問題を解決するためにSPSシンタックス要素に依存するAPSシンタックス要素scaling_list_chroma_present_flagの値に関して、次のアプローチの 1 つ以上が開示されている。

1) 一例では、VPS ID及び / 又はSPS ID及び / 又はPPS IDをAPSシンタックス構造、すなわち、adaptation_parameter_set_rbsp () に追加することができる。例えば、adaptation_parameter_set_rbsp () のシンタックス構造を次のように変更できる：

【表 1 6】

adaptation_parameter_set_rbsp () {	Descriptor
adaptation_parameter_set_id	<i>u(5)</i>
<i>aps_seq_parameter_set_id</i>	<i>u(4)</i>
aps_params_type	<i>u(3)</i>
if(aps_params_type == ALF_APS)	
alf_data ()	

20

aps_seq_parameter_set_id は、APS の *sps_seq_parameter_set_id* の値を指定する。*aps_seq_parameter_set_id* の値は、両端を含む 0 ~ 1 5 の範囲であるものとする。*aps_seq_parameter_set_id* の値は、CLVS 内のコーディングされたピクチャによって参照されるすべての APS で同じであるものとする。

30

a . 更に、代替として、クロマスケールリングリストのシグナリングは、ChromaArrayType の値に基づいて明示的に条件付けされる場合がある。例えば、scaling_list_data () のシンタックステーブルは、次のように変更できる：

【表 1 7】

scaling_list_data () {	Descriptor
scaling_matrix_for_lfnst_disabled_flag	<i>u(1)</i>
[[scaling_list_chroma_present_flag]]	<i>u(1)</i>
for(id = 0; id < 28; id ++)	
matrixSize = (id < 2) ? 2 : ((id < 8) ? 4 : 8)	
if([[scaling_list_chroma_present_flag]] (<i>ChromaArrayType</i> != 0) (id % 3 == 2) (id == 27)) {	
scaling_list_copy_mode_flag[id]	<i>u(1)</i>
...	

40

50

また、`scaling_list_chroma_present_flag`のセマンティクスは次のように変更される：

[[`scaling_list_chroma_present_flag`が1に等しいことは、クロマスケールリングリストが`scaling_list_data()`に存在することを指定する。`scaling_list_chroma_present_flag`が0に等しいことは、クロマスケールリングリストが`scaling_list_data()`に存在しないことを指定する。`scaling_list_chroma_present_flag`が、`ChromaArrayType`が0に等しい場合は0に等しく、`ChromaArrayType`が0に等しくない場合は1に等しいことが、ビットストリーム準拠の要件である。]]

2) 一例では、APSに関連付けられたVPS及び/又はSPS及び/又はPPSが暗黙的に導出される場合がある。

a. 例えば、APSがビデオユニット(ピクチャヘッダやスライスヘッダなど)によって参照され、ビデオユニットがVPS及び/又はSPS及び/又はPPSに依存している場合、APSは暗黙的にVPS及び/又はSPS及び/又はPPSに関連付けられる。

3) 一例では、ユーザ定義のスケールリングリスト(明示的なスケールリングリストとも呼ばれる)を使用する代わりに、明示的なスケールリングリストがルマブロックに適用されている場合でも、フラット量子化(デフォルトスケールリングリスト)をクロマブロックに使用できる。

a. 又は、更に、ルマブロックの明示的なスケールリングリストがビットストリームでシグナリングされる場合でも、クロマブロックの明示的なスケールリングリストがシグナリングされない場合がある。

4) 又は、`scaling_list_chroma_present_flag`の値を`ChromaArrayType`の値と分離することもできる。

a. 異なる色成分(例えば、ルマブロックとクロマブロック)に明示的なスケールリングリスト又はデフォルトのスケールリングリストを使用するかどうかの指示は、個別にシグナリング/制御される場合がある。

i. 一例では、シンタックス要素(例えば、1つ以上のフラグ)をSPS/PPS/PH/SHに追加して、ルマ成分及び/又はクロマ成分に対してユーザ定義のスケールリングリスト(明示的なスケールリングリストとも呼ばれる)を有効にするかどうかを指定できる。

ii. 例えば、SPSにフラグを追加して、ルマ変換係数をフラット量子化(デフォルトのスケールリングリスト)とユーザ定義のスケールリングリストの間で切り換え可能にすることができる。

ii. 例えば、SPSに1つ以上のフラグを追加して、クロマ-U及び/又はルマ-V変換係数をフラット量子化(デフォルトのスケールリングリスト)とユーザ定義のスケールリングリストの間で切り換え可能にすることができる。

b. 例えば、`ChromaArrayType`が0に等しい場合、`scaling_list_chroma_present_flag`は1に等しくなる。

i. 一例では、4:0:0クロマフォーマットのコーディングピクチャの場合、スケールリングマトリクスのN(N=28など)個のセットがAPSでシグナリングできる。

ii. 一例では、`separate_colour_plane_flag`が1に等しい4:4:4クロマフォーマットのコーディングピクチャの場合、スケールリングマトリクスのM(M=28など)個のセットがAPSでシグナリングできる。

a) 例えば、APSで信号が送られた`separate_colour_plane_flag`が1であり、スケールリングマトリクスのM(M=28など)個のセットの場合、Y(Luma)、U(Cb)、及びV(Cr)チャンネル変換係数の各々は、luma-Yチャンネルとして扱うことができ、Y、U、及びV変換係数のスケールリングマトリクス識別子変数idは、Y成分(例えば、0の値)と等しい色成分に関して導出される。

b) あるいは、`separate_colour_plane_flag`が1に等しく、スケールリングマトリクスのM(M=28など)個のセットがAPSでシグナリングされた場合、ルマ-Y変換係数のスケールリングマトリクス識別子変数idは、色成分がY成分に等しい(例えば、0の値)として導出され、一方、クロマ-Uのスケールリングマトリクス識別子変数idは、色成分がU成分

10

20

30

40

50

に等しい（例えば、1の値）として導出され、クロマ - Vのスケーリングマトリクス識別子変数idは、色成分がV成分に等しい（例えば、2の値）として導出される。

c.例えば、ChromaArrayTypeが1に等しい場合、scaling_list_chroma_present_flagは0であってよい。

i.一例では、クロマ変換係数がユーザ定義のスケーリングリストを使用できるかどうかは、scaling_list_chroma_present_flagの値に依存してよい。

a)例えば、scaling_list_chroma_present_flagが0に等しい場合、sps_scaling_list_enabled_flag、ph_scaling_list_enabled_flag、及びslice_scaling_list_enabled_flagの値に関係なく、ユーザ定義のスケーリングリストをクロマ変換係数に使用することはできない（例えば、クロマのユーザ定義のスケーリングリストの使用を指定するために使用される追加フラグの値は、0や1などの特定の数値と等しい必要がある）。 10

b)例えば、scaling_list_chroma_present_flagが1に等しい場合、ユーザ定義のスケーリングリストをクロマ変換係数に使用できる場合がある。

ii.一例では、separate_colour_plane_flagが0に等しい4:2:0及び/又は4:2:2クロマフォーマット、及び/又は4:4:4クロマフォーマットのコーディングピクチャの場合、スケーリングマトリクスのN（N=10など）個のセットがAPSでシグナリングされてよい。

a)例えば、ChromaArrayTypeが0より大きく、スケーリングマトリクスのN（N=10など）個のセットがAPSでシグナリングされる場合、U及び/又はV変換係数のスケーリングマトリクスは、Y変換係数のスケーリングマトリクスのシグナリングされたN個のセットから導出されてよい。 20

b)又は、ChromaArrayTypeが0より大きく、スケーリングマトリクスのN（N=10など）個のセットがAPSでシグナリングされる場合、U及び/又はV変換係数は、ユーザ定義のスケーリングリストを使用しない（代わりに、U及び/又はV変換係数は、デフォルトのスケーリング係数を使用してフラット量子化を使用してよい）。

d.例えば、ChromaArrayTypeに基づくscaling_list_chroma_present_flagに関するセマンティクス制約は、次のように、シンタックス要素scaling_list_chroma_present_flagに関連付けられなくてよい：

scaling_list_chroma_present_flagが1に等しいことは、クロマスケーリングリストがscaling_list_data()に存在することを指定する。scaling_list_chroma_present_flagが0に等しいことは、クロマスケーリングリストがscaling_list_data()に存在しないことを指定する。[[scaling_list_chroma_present_flagが、ChromaArrayTypeが0に等しい場合は0に等しく、ChromaArrayTypeが0に等しくない場合は1に等しいことが、ビットストリーム準拠の要件である。]] 30

e.例えば、ChromaArrayTypeに基づくscaling_list_chroma_present_flagに関するセマンティクス制約は、次のように変更できる：

scaling_list_chroma_present_flagが1に等しいことは、クロマスケーリングリストがscaling_list_data()に存在することを指定する。scaling_list_chroma_present_flagが0に等しいことは、クロマスケーリングリストがscaling_list_data()に存在しないことを指定する。ChromaArrayTypeが0に等しい場合、scaling_list_chroma_present_flagが0に等しいことが、ビットストリーム準拠の要件である。 40

【数55】

[[ChromaArrayTypeが0に等しくない場合は1に等しくなるものとする]]

5)又は、PH/SHシンタックス要素に関連付けられた制約を追加して、PH/SHシンタックス要素によって導出されたChromaArrayTypeに従ってscaling_list_chroma_present_flagの値を特定の値（0や1など）に制約することもできる。例えば、次のようになる：

10

20

30

40

50

一例では、`ph_scaling_list_aps_id`のセマンティクスは次のように変更される：

`ph_scaling_list_aps_id`は、スケーリングリストAPSの`adaptation_parameter_set_id`を指定する。`aps_params_type`が`SCALING_APS`と等しく、`adaptation_parameter_set_id`が`ph_scaling_list_aps_id`と等しいAPS NALユニットの`TemporalId`は、PHに関連付けられたピクチャの`TemporalId`以下であるものとする。

【数56】

aps_params_type が *SCALING_APS* に等しく *adaptation_parameter_set_id* が *ph_scaling_list_aps_id* に等しい APS NAL の *scaling_list_chroma_present_flag* の値は *ChromaArrayType* == 0?0:1 に等しいものとする。

10

又は、`ph_scaling_list_aps_id`のセマンティクスは次のように変更される：

`ph_scaling_list_aps_id`は、スケーリングリストAPSの`adaptation_parameter_set_id`を指定する。`aps_params_type`が`SCALING_APS`と等しく、`adaptation_parameter_set_id`が`ph_scaling_list_aps_id`と等しいAPS NALユニットの`TemporalId`は、PHに関連付けられたピクチャの`TemporalId`以下であるものとする。

【数57】

ChromaArrayType が 1 に等しい場合、*aps_params_type* が *SCALING_APS* に等しく *adaptation_parameter_set_id* が *ph_scaling_list_aps_id* に等しい APS NAL ユニットの *scaling_list_chroma_present_flag* の値は、1 に等しいものとする。

20

又は、`ph_scaling_list_aps_id`のセマンティクスは次のように変更される：

`ph_scaling_list_aps_id`は、スケーリングリストAPSの`adaptation_parameter_set_id`を指定する。`aps_params_type`が`SCALING_APS`と等しく、`adaptation_parameter_set_id`が`ph_scaling_list_aps_id`と等しいAPS NALユニットの`TemporalId`は、PHに関連付けられたピクチャの`TemporalId`以下であるものとする。

【数58】

ChromaArrayType が 0 に等しい場合、*aps_params_type* が *SCALING_APS* に等しく *adaptation_parameter_set_id* が *ph_scaling_list_aps_id* に等しい APS NAL ユニットの *scaling_list_chroma_present_flag* の値は、0 に等しいものとする。

30

また、APS SEセマンティクスは次のように変更される：

`scaling_list_chroma_present_flag`が1に等しいことは、クロマスケーリングリストが`scaling_list_data()`に存在することを指定する。`scaling_list_chroma_present_flag`が0に等しいことは、クロマスケーリングリストが`scaling_list_data()`に存在しないことを指定する。[[`scaling_list_chroma_present_flag`が、`ChromaArrayType`が0に等しい場合は0に等しく、`ChromaArrayType`が0に等しくない場合は1に等しいことが、ビットストリーム準拠の要件である。]]

40

2.`ChromaArrayType`が0に等しい場合の不要なクロマ関連APSシンタックス要素のシグナルについて、第2の問題を解決するために、次のアプローチの1つ以上が開示される：

1) 一例では、シンタックス要素(例えばフラグ)をAPSシンタックス構造`Imcs_data()`に追加して、クロマ残差スケーリング関連のAPSシンタックス要素(例えば、`Imcs_delta_abs_crs`、`Imcs_delta_sign_crs_flag`など)の存在を制御することができる。

a.例えば、`ChromaArrayType`が0に等しい場合、クロマ残差スケーリング関連のAPSシンタックス要素(例えば、`Imcs_delta_abs_crs`、`Imcs_delta_sign_crs_flag`など)はシグナリングできないようにする必要がある。例えば、追加されたフラグは0や1などの特定の値と等しい必要がある。

50

b.例えば、ChromaArrayTypeが0に等しくない場合、クロマ残差スケーリング関連のAPSシンタックス要素（例えば、Imcs_delta_abs_crs、Imcs_delta_sign_crs_flagなど）はシグナリングされる必要がある。例えば、追加されたフラグは0や1などの特定の値と等しい必要がある。

c.例えば、現在スライスでクロマ残差スケーリングの使用が許可されるかどうかは、追加されたフラグに依存してよい。例えば、追加フラグが、クロマ残差スケーリング関連のAPSシンタックス要素がシグナルされないことを示す場合、sps_lmcs_enabled_flag、ph_lmcs_enabled_flag、ph_chroma_residual_scale_flag、及びsh_lmcs_enabled_flagの値に関係なく、クロマ残差スケーリングは使用されない。

2) PH/SH/APSシンタックス要素のセマンティクスの下にビットストリーム制約を追加して、ChromaArrayTypeの値に関してImcs_delta_abs_crsの値を制約することができる。例えば、次のようになる：

ph_lmcs_aps_idは、PHに関連付けられたスライスが参照するLMCS APSのadaptation_parameter_set_idを指定する。aps_params_typeがLMCS_APSと等しく、adaptation_parameter_set_idがph_lmcs_aps_idと等しいAPS NALユニットのTemporalIdは、PHに関連付けられたピクチャのTemporalId以下であるものとする。

【数59】

ChromaArrayType が 0 に等しい場合、*aps_params_type* が *LMCS_APS* に等しく *adaptation_parameter_set_id* が *ph_lmcs_aps_id* に等しい *APS NAL* ユニットの *Imcs_delta_abs_crs* の値は、0に等しいものとする

また、scaling_list_chroma_present_flagのセマンティクスは次のように変更される：

ph_lmcs_aps_idは、PHに関連付けられたスライスが参照するLMCS APSのadaptation_parameter_set_idを指定する。aps_params_typeがLMCS_APSと等しく、adaptation_parameter_set_idがph_lmcs_aps_idと等しいAPS NALユニットのTemporalIdは、PHに関連付けられたピクチャのTemporalId以下であるものとする。

【数60】

ChromaArrayType が 1 に等しい場合、*aps_params_type* が *LMCS_APS* に等しく *adaptation_parameter_set_id* が *ph_lmcs_aps_id* に等しい *APS NAL* ユニットの *Imcs_delta_abs_crs* の値は、0に等しいものとする。

また、scaling_list_chroma_present_flagのセマンティクスは次のように変更される：

ph_lmcs_aps_idは、PHに関連付けられたスライスが参照するLMCS APSのadaptation_parameter_set_idを指定する。aps_params_typeがLMCS_APSと等しく、adaptation_parameter_set_idがph_lmcs_aps_idと等しいAPS NALユニットのTemporalIdは、PHに関連付けられたピクチャのTemporalId以下であるものとする。

【数61】

ChromaArrayType が 0 に等しい場合、*aps_params_type* が *LMCS_APS* に等しく *adaptation_parameter_set_id* が *ph_lmcs_aps_id* に等しい *APS NAL* ユニットの *Imcs_delta_abs_crs* の値は、0に等しいものとする。その他の場合、*aps_params_type* が *LMCS_APS* に等しく *adaptation_parameter_set_id* が *ph_lmcs_aps_id* に等しい *APS NAL* ユニットの *Imcs_delta_abs_crs* の値は、0より大きいものとする。

3.上記の例では、「ChromaArrayType」という用語は「色フォーマットが4:0:0と

等しいことを確認する」に置き換えられてよい。

4. 第3の問題を解決するためのRRCとTSRCの使用について、次のアプローチの1つ以上が開示されている：

1) TSRCの有効化/無効化フラグ（例えば、`slice_ts_residual_coding_disabled_flag`）のシグナリングは、変換スキップが有効になっているかどうか（例えば、SPSの`sps_transform_skip_enabled_flag`）を条件としてよい。

a. 一例では、次が適用されてよい：

【表18】

<i>if(!sps_transform_skip_enabled_flag)</i>	
<code>slice_ts_residual_coding_disabled_flag</code>	u(1)

10

代替として、`slice_ts_residual_coding_disabled_flag`が存在しない場合、1と等しいと推定される。

2) 又は、TSRC有効化フラグ（例えば、`slice_ts_residual_coding_disabled_flag`）の値は、SPSの`sps_transform_skip_enabled_flag`によって制約されてよい。例えば、`slice_ts_residual_coding_disabled_flag`のセマンティクスは次のように変更できる：

`slice_ts_residual_coding_disabled_flag`が1に等しいことは、現在スライスの変換スキップブロックの残差サンプルを解析するために、`residual_coding()` シンタックス構造が使用されることを指定する。`slice_ts_residual_coding_disabled_flag`が0に等しいことは、現在スライスの変換スキップブロックの残差サンプルを解析するために、`residual_ts_coding()` シンタックス構造が使用されることを指定する。

20

【数62】

[[`slice_ts_residual_coding_disabled_flag`]] *sps_transform_skip_enabled_flag* [[が存在しない]] が0に等しい場合、*slice_ts_residual_coding_disabled_flag* の値は[[0]] 1に等しい[[と推定される]] ものとする。

30

3) 又は、TSRC有効化フラグ（例えば、`slice_ts_residual_coding_disabled_flag`）のシグナリングは他のシンタックス要素を条件としない場合、常に存在してよい。例えば、`slice_ts_residual_coding_disabled_flag`のセマンティクスは次のように変更できる：

`slice_ts_residual_coding_disabled_flag`が1に等しいことは、現在スライスの変換スキップブロックの残差サンプルを解析するために、`residual_coding()` シンタックス構造が使用されることを指定する。`slice_ts_residual_coding_disabled_flag`が0に等しいことは、現在スライスの変換スキップブロックの残差サンプルを解析するために、`residual_ts_coding()` シンタックス構造が使用されることを指定する。[[`slice_ts_residual_coding_disabled_flag`が存在しない場合、0と等しいと推定される。]]

40

4) 更に、TSRCは非変換スキップ（非TS）コーディングブロックにも適用できる。

a. 一例では、1つ以上のシンタックスフラグを追加して、非TSブロックに対してTSRC又はRRCを有効にするかどうかを指定できる。

i. 一例では、1つ以上のブロックレベル（CTU/CU/TU）シンタックスフラグを追加して、現在ビデオユニットがTSRC又はRRCのどちらを使用しているかを指定できる。

ii. また、代替として、ビデオユニットに対してTSRCを許可するかどうかを指定するために、1つ以上の高レベル（SPS/PPS/PH/SH）シンタックスフラグを追加することもできる。

b. 一例では、非TSコーディングブロックにTSRCを使用するかどうか、又は非TSコーディングブロックにTSRCを許可するかどうかは、ブロックのQP値などのコーディング情報

50

に依存してよい。

i. 一例では、QPがXと等しいかXより大きくない (X=4 など) 非TSブロックの場合、残差コーディングにTSRC又はRRCのいずれかを使用することが許可されてよい。

5. 第4の問題を解決するためのルマブロックに対するcu_qpデルタのオン/オフ制御に関して、以下のアプローチの1つ以上が開示されている：

a. SHレベルのシンタックス要素 (例えば、slice_cu_qp_delta_enabled_flagで表されるフラグ) を追加して、特定のスライスに対するcu_qpデルタの有効化及び/又は無効化を制御できる。

i. 一例では、提案されたslice_cu_qp_delta_enabled_flagの存在は、PPSのcu_qp_delta_enabled_flagを条件とする。例えば、PPSのcu_qp_delta_enabled_flagが1に等しい場合のみ、提案されたslice_cu_qp_delta_enabled_flagがシグナリングされ、それ以外の場合 (PPSのcu_qp_delta_enabled_flagが0に等しい)、提案されたslice_cu_qp_delta_enabled_flagはシグナリングされず、0に等しいと推定される。

10

a) 又は、提案されたslice_cu_qp_delta_enabled_flagの値は、PPSのcu_qp_delta_enabled_flagの値に制限される。つまり、PPSのcu_qp_delta_enabled_flagが0に等しい場合、提案されたslice_cu_qp_delta_enabled_flagの値は0に等しくなる。

ii. 一例では、PPSのcu_qp_delta_enabled_flagを使用して、SH内のSHレベルのcu_qpデルタ有効化フラグの存在、及び/又は変換ユニットシンタックスとパレットコーディングシンタックス内のcu_qp_delta_abs及び/又はcu_qp_delta_sign_flagの存在を制御できる。

20

一例では、シンタックス構造は次のように変更できる：

PPSシンタックス構造は次のように変更される：

【表 19】

pic_parameter_set_rbsp() {	Descriptor
pps_pic_parameter_set_id	ue(v)
...	
init_qp_minus26	se(v)
pps_cu_qp_delta_enabled_flag	u(1)
pps_chroma_tool_offsets_present_flag	u(1)

30

【数 63】

pps_cu_qp_delta_enabled_flag が 1 に等しいことは、ph_cu_qp_delta_subdiv_intra_slice 及び ph_cu_qp_delta_subdiv_inter_slice シンタックス要素が PPS を参照する PH に存在し、cu_qp_delta_abs が変換ユニットシンタックス及びパレットコーディングシンタックスに存在する可能性があることを指定する。 pps_cu_qp_delta_enabled_flag が 0 に等しいことは、ph_cu_qp_delta_subdiv_intra_slice 及び ph_cu_qp_delta_subdiv_inter_slice シンタックス構造が PPS を参照する PH に存在しないこと、及び cu_qp_delta_abs が変換ユニットシンタックス及びパレットコーディングシンタックスに存在しないことを指定する。

40

また、PHシンタックス構造は次のように変更される：

50

【表 2 0】

picture_header_structure() {	Descriptor
gdr_or_irap_pic_flag	u(1)
...	
if(ph_intra_slice_allowed_flag) {	
...	
if(pps_cu_qp_delta_enabled_flag)	
ph_cu_qp_delta_subdiv_intra_slice	ue(v)
...	
if(ph_inter_slice_allowed_flag) {	
...	
if(pps_cu_qp_delta_enabled_flag)	
ph_cu_qp_delta_subdiv_inter_slice	ue(v)
...	

10

20

また、SHシンタックス構造は次のように変更される：

【表 2 1】

slice_header() {	Descriptor
picture_header_in_slice_header_flag	u(1)
...	
if(pps_cu_qp_delta_enabled_flag)	
cu_qp_delta_enabled_flag	u(1)
if(pps_cu_chroma_qp_offset_list_enabled_flag)	
cu_chroma_qp_offset_enabled_flag	u(1)

30

【数 6 4】

cu_qp_delta_enabled_flag が 1 に等しいことは、*cu_qp_delta_abs* が変換ユニット及びパレットコーディングシンタックスに存在する可能性があることを指定する。*cu_qp_delta_enabled_flag* が 0 に等しいことは、*cu_qp_delta_abs* が変換ユニット又はパレットコーディングシンタックスに存在しないことを指定する。存在しない場合、*cu_qp_delta_enabled_flag* の値は 0 に等しいと推定される。

40

b.更に、シンタックス構造palette_coding()及び/又はシンタックス構造transform_unit()における*cu_qp_delta_abs*の存在は、提案された*slice_cu_qp_delta_enabled_flag*を条件とする。例えば、提案された*slice_cu_qp_delta_enabled_flag*の値が 1 に等しい場合にのみ、*cu_qp_delta_abs*がシグナリングされ、それ以外の場合(提案された*slice_cu_qp_delta_enabled_flag*が 0 に等しい)、*cu_qp_delta_abs*の値はシグナリングされず、0 に等しいと推定される。

50

あるいは、クロマcu qpオフセットがスライスレベルのオン/オフフラグによって制御されなくてもよい。例えば、クロマcu qpオフセットが現在スライスに適用されるかどうかは、PH/PPS/SPSレベルフラグに依存してよい。

c.又は、PHレベルのシンタックス要素（例えば、ph_cu_qp_delta_enabled_flagで表されるフラグ）を追加して、特定のスライスに対するcu qpデルタの有効化及び/又は無効化を制御することもできる。

6.PPS SE、single_slice_per_subpic_flag、num_slices_in_pic_minus1、tile_idx_delta_present_flagの設計に関して、及び第5の問題を解決するために、次のアプローチの1つ以上が開示されている：

a.一例では、PH/SH/PPSシンタックス要素のセマンティクスに制約が追加されてよい。例えば、次のようになる：

ph_pic_parameter_set_idは、使用中のPPSのpps_pic_parameter_set_idの値を指定する。ph_pic_parameter_set_idの値は、両端を含む0～63の範囲であるものとする。

PHのTemporalIdの値が、ph_pic_parameter_set_idに等しいpps_pic_parameter_set_idを持つPPSのTemporalIdの値以上であることはビットストリーム適合性の要件である。

【数65】

sps_num_subpics_minus1 が0に等しい場合、*rect_slice_flag* は1に等しく、ピクチャは1つのみのスライスを含み、別のピクチャは1つより多くのピクチャを含み、2つのピクチャは異なるPPSを参照するものとする。

b.一例では、single_slice_per_subpic_flagのセマンティクスが次のように変更されてよい：

【数66】

single_slice_per_subpic_flag が1に等しいことは、各サブピクチャが1つのみの長方形スライスで構成されることを指定する。*single_slice_per_subpic_flag* が0に等しいことは、*sps_num_subpics_minus1* が0より大きいならば(この場合には、各ピクチャは1つより多くのスライスを含む)、各サブピクチャは1つ以上の長方形スライスを含んでよいが、全部のサブピクチャの各々が1つのみの長方形スライスを含むのではない。その他の場合 (*sps_num_subpics_minus1* が0に等しく、この場合、各ピクチャは1つのみのサブピクチャを含む)、各ピクチャは1つより多くスライスを含む。存在しない場合、*single_slice_per_subpic_flag* の値は0に等しいと推定される。

c.一例では、single_slice_per_subpic_flagのセマンティクスに次のように制約を追加できる：

single_slice_per_subpic_flag が1に等しいことは、各サブピクチャが1つの長方形スライスだけで構成されることを指定する。*single_slice_per_subpic_flag* が0に等しいことは、各サブピクチャが1つ以上の長方形スライスで構成されてよいことを指定する。存在しないとき、*single_slice_per_subpic_flag* の値は0に等しいと推定される。

【数67】

sps_num_subpics_minus1 が0に等しく、PPSを参照する各ピクチャに1つのみのスライスが存在する場合、*single_slice_per_subpic_flag* の値は1に等しいものとする。

d.更に、single_slice_per_subpic_flagのセマンティクスに次のような制約を追加することもできる：

single_slice_per_subpic_flag が1に等しいことは、各サブピクチャが1つの長方形

スライスだけで構成されることを指定する。single_slice_per_subpic_flagが0に等しいことは、各サブピクチャが1つ以上の長方形スライスで構成されてよいことを指定する。存在しないとき、single_slice_per_subpic_flagの値は0に等しいと推定される。

【数68】

sps_num_subpics_minus1が0より大きく、各サブピクチャに1つのみのスライスが存在する場合、single_slice_per_subpic_flagの値は1に等しいものとする。

e.一例では、各サブピクチャが1つの長方形スライスだけで構成されている場合、single_slice_per_subpic_flagは1に等しくなければならないという制約がある。

10

f.また、PPSシンタックス要素tile_idx_delta_present_flagの存在は、num_slices_in_pic_minus1に基づいて条件付けされないこともある。例えば次の通りである：

【表22】

[[if(num_slices_in_pic_minus1 > 0)]]	
tile_idx_delta_present_flag	u(1)

g.一例では、single_slice_per_subpic_flagが1に等しい場合、num_slices_in_pic_minus1がsps_num_subpics_minus1に等しくなるように制約されている。

20

h.また、代替として、PPSシンタックス要素num_slices_in_pic_minus1をnum_slices_in_pic_minus2に変更することもできる。

i.更に、num_slices_in_pic_minus2に基づいてtile_idx_delta_present_flagの存在を次のように条件付けする：

【表23】

num_slices_in_pic_minus[[1]]2	ue(v)
[[if(num_slices_in_pic_minus1 > 0)]]	
tile_idx_delta_present_flag	u(1)

30

【数69】

num_slices_in_pic_minus[[1]]2に[[1]]2を加えたものは、PPSを参照する各ピクチャ内の長方形スライスの数を指定する。num_slices_in_pic_minus[[1]]2の値は、両端を含む0からMaxSlicesPerPicture - [[1]]2の範囲であるものとする。ここで、MaxSlicesPerPictureはAnnex Aで指定される。変数NumSlicesInPicの値は、num_slices_in_pic_minus2 + 2に等しく導出される。

no_pic_partition_flagが1に等しい場合、変数NumSlicesInPicの値は、1に等しく導出される[[num_slices_in_pic_minus1は0に等しいと推定される]]。single_slice_per_subpic_flagが1に等しい場合、NumSlicesInPicは[[num_slices_in_pic_minus1は]]sps_num_subpics_minus1 - 1に等しく[[推定される]]導出される。

40

更に、VVCドラフトテキストの他のすべての場所の「num_slices_in_pic_minus1」は「NumSlicesInPic-1」に置き換えられる。

7.第6の問題を解決するためのスライスとタイルレイアウトのシグナリングに関して、次のアプローチの1つ以上が開示されている：

a.PPSにシンタックス要素（例えば、1つ以上のフラグ）を追加して、ピクチャを同じ

50

高さの最初の幾つかのタイル行/列と、異なる高さ/幅の最後の幾つかのタイル行/列とを持つ複数のタイル行/列に分割するかどうかを指定することができる。

i. 例えば、提案されたシンタックスフラグはno_pic_partition_flag及び/又は明示的なタイルの行/列の数 (num_exp_tile_columns_minus1 及び/又は num_exp_tile_rows_minus1 など) に依存し、例えば以下ようになる :

【表 2 4】

no_pic_partition_flag	u(1)	
if(!no_pic_partition_flag) {		
pps_log2_ctu_size_minus5	u(2)	10
num_exp_tile_columns_minus1	ue(v)	
num_exp_tile_rows_minus1	ue(v)	
<i>if(num_exp_tile_columns_minus1 > 1)</i>		
<i>exp_tile_columns_in_reverse_order</i>	u(1)	
<i>if(num_exp_tile_rows_minus1 > 1)</i>		
<i>exp_tile_rows_in_reverse_order</i>	u(1)	
for(i = 0; i <= num_exp_tile_columns_minus1; i++)		20
tile_column_width_minus1[i]	ue(v)	
for(i = 0; i <= num_exp_tile_rows_minus1; i++)		
tile_row_height_minus1[i]	ue(v)	

【数 7 0】

*exp_tile_columns_in_reverse_order*が1に等しいことは、ピクチャが、同じ幅の最初の幾つかのタイル列と異なる幅の最初の幾つかのタイル列とを有する2つより多くのタイル列に分割されることを指定する。

*exp_tile_columns_in_reverse_order*が0に等しいことは、そのようなタイルレイアウトを課さない。
*exp_tile_rows_in_reverse_order*が1に等しいことは、ピクチャが、同じ高さの最初の幾つかのタイル行と異なる高さの最後の幾つかのタイル行とを有する2つより多くのタイル行に分割されることを指定する。
*exp_tile_rows_in_reverse_order*が0に等しいことは、そのようなタイルレイアウトを課さない。

更に、提案されたシンタックスフラグが1に等しい場合、タイルの列の幅及び/又はタイルの行の高さは、提案されたシンタックスフラグの値に従って次のように導出される :

タイル列の数を指定する変数NumTileColumnsと、両端を含む0からNumTileColumns-1までの範囲のiについて、i番目のタイル列の幅をCTB単位で指定するリストcolWidth[i]は、次のように導出される :

【数 7 1】

10

20

30

40

50

```

remainingWidthInCtbsY = PicWidthInCtbsY
for( i = 0; i < num_exp_tile_columns_minus1; i++ ) {
    ccolWidth[ i ] = tile_column_width_minus1[ i ] + 1
    remainingWidthInCtbsY -= ccolWidth[ i ]
}
uniformTileColWidth = tile_column_width_minus1[ num_exp_tile_columns_minus1 ] +
1 (23)
while( remainingWidthInCtbsY >= uniformTileColWidth ) {
    ccolWidth[ i++ ] = uniformTileColWidth
    remainingWidthInCtbsY -= uniformTileColWidth
}
if( remainingWidthInCtbsY > 0 )
    ccolWidth[ i++ ] = remainingWidthInCtbsY
NumTileColumns = i
if( exp_tile_cols_in_reverse_order )
    for( k = 0; k < NumTileColumns; k++ ) {
        colWidth[ k ] = ccolWidth[ NumTileColumns - 1 - k]
    }

```

タイル行の数を指定する変数NumTileRowsと、両端を含む0からNumTileRows-1までの範囲のjについて、j番目のタイル行の高さをCTB単位で指定するリストRowHeight[j]は、次のように導出される：

【数72】

```

remainingHeightInCtbsY = PicHeightInCtbsY
for( j = 0; j < num_exp_tile_rows_minus1; j++ ) {
    cRowHeight[ j ] = tile_row_height_minus1[ j ] + 1
    remainingHeightInCtbsY -= cRowHeight[ j ]
}
uniformTileRowHeight = tile_row_height_minus1[ num_exp_tile_rows_minus1 ] + 1
(24)
while( remainingHeightInCtbsY >= uniformTileRowHeight ) {
    cRowHeight[ j++ ] = uniformTileRowHeight
    remainingHeightInCtbsY -= uniformTileRowHeight
}
if( remainingHeightInCtbsY > 0 )
    cRowHeight[ j++ ] = remainingHeightInCtbsY
NumTileRows = j
if( exp_tile_rows_in_reverse_order )
    for( k = 0; k < NumTileRows; k++ ) {
        RowHeight[ k ] = cRowHeight[ NumTileRows - 1 - k]
    }

```

b.更に、同様に、タイルが複数のスライスによって分割されている場合（この場合、スライスサイズはタイルサイズより小さい）、タイルを同じ高さの最初の幾つかのスライス

行と異なる高さの最後の幾つかのスライス行で複数のスライス行に分割するかどうかを指定するシンタックス要素（例えば、1つ以上のフラグ）がPPSに追加されることがある。

i. 更に、提案されたシンタックスフラグが1に等しい場合は、提案されたシンタックスフラグの値に従ってスライスの高さ（SliceHeightInCtusMinus 1 など）が導出される：

8. 第7の問題を解くためのALF APSが存在しない場合については、以下のアプローチのうち1つ以上が開示される：

a. 一例では、ALF APSがない場合（例えば、no_aps_constraint_flagが1に等しい、又は必要なAPS IDを持つAPSが使用できない）、ALFが許可されないことがある（この場合、sps_alf_enabled_flagとsps_ccalf_enabled_flagは0に等しい必要がある）。

b. 一例では、ALF APSがない場合でも（例えば、no_aps_constraint_flagが1に等しい、又は必要なAPS IDを持つAPSが使用できない）、ALFが許可されてよい（この場合、sps_alf_enabled_flagは0又は1に等しいことが許可される）。

i. 例えば、ALF APSが存在しない場合（例えば、no_aps_constraint_flagは1に等しい）、ph_alf_enabled_flag及び/又はslice_alf_enabled_flagは0又は1に等しいことが許可される。

ii. 例えば、ALF APSがない場合（例えば、no_aps_constraint_flagは1に等しい）、クロマALFとCC - ALFは許可されないが、フィルタが固定されたルマALFを使用することができる。

iii. 例えば、ALF APSが存在しない場合（例えば、no_aps_constraint_flagは1に等しい）、ph_num_alf_aps_ids_luma、ph_alf_chroma_idc、slice_num_alf_aps_ids_luma、slice_alf_chroma_idc、sps_ccalf_enabled_flagの値は0である必要がある。

c. 一例では、GCIシンタックス要素no_alf_constraint_flagが1に等しい場合、ALF及び/又はCCALFが許可されないことがある（この場合、sps_alf_enabled_flag及び/又はsps_ccalf_enabled_flagは0に等しい必要がある）。

d. あるいは、使用するALF APS数（例えば、ph_num_alf_aps_ids_luma）及び/又はALF/CC - ALF APSインデックス（例えば、ph_alf_aps_id_luma、ph_alf_aps_id_chroma、ph_cc_alf_cb_aps_id、ph_cc_alf_cr_aps_id）をシグナリングするかどうかは、ALF APSが許可されているかどうか（例えば、no_aps_constraint_flag）に依存してよい。

i. 一例では、ALF APSが適用されていない場合、その情報はシグナリングされなくてよい。

e. 一例では、ALF、CCALF、及び/又はLMCS、及び/又はユーザ定義のスケーリングリストを無効にするために、新しいシンタックス要素がSPS/PPS/PH/SH/GCIでシグナリングされてよい。

f. 一例では、外部手段によって提供されるAPS/VPSは、GCIシンタックス要素（例えば、APS/VPSの制約フラグ）に従って非活性化/無効化/制限/オフ/不許可にされてよい。

i. 例えば、ビットストリームでシグナリングされたAPS/VPSと、外部手段によって提供されたAPS/VPSの使用は、非活性化/無効化/制限/オフ/不許可にされてよい。

ii. 例えば、GCIシンタックス要素（例えば、no_aps_constraint_flagという名前のフラグ）は、GCIシンタックス構造でシグナリングされ、復号処理でのVPSの使用の非活性化/無効化/オフを指定する（例えば、そのような場合、ビットストリーム内のAPSと外部手段によって提供されるAPSのどちらも許可されない）。

iii. 例えば、特定の値（例えば1）に等しいGCIシンタックス要素no_aps_constraint_flagは、復号処理で使用可能なnuh_unit_typeがPREFIX_APS_NUT又はSUFFIX_APS_NUTと等しいNALユニットが存在しないことを指定する。例えば、JVET - R2001 - vAのテキストは次のように変更される（追加された部分は太字の斜体のテキストで強調表示され、削除された部分は右及び左の二重角括弧（例えば、[[[]]]）で強調表示され、二重角括弧の間に削除されたテキストが表示される）。

【数73】

10

20

30

40

50

no_aps_constraint_flag が 1 に等しいことは、ビットストリーム内に存在するか又は外部手段により提供される[[present in OlsInScopeに存在する]]*OlsInScope* のビットストリームを復号するときにはデコーダに利用可能な、**nuh_unit_type** が **PREFIX_APS_NUT** 又は **SUFFIX_APS_NUT** に等しい NAL ユニットが存在しないことを指定する。**sps_lmcs_enabled_flag** 及び **sps_scaling_list_enabled_flag** は 0 に等しいものとする。**no_aps_constraint_flag** が 0 に等しいことは、そのような制約を課さない。

或いは、

no_aps_constraint_flag が 1 に等しいことは、ビットストリーム内に存在する又は外部手段により提供された[[OlsInScope内に存在する]]*OlsInScope* のビットストリームを復号するときにはデコーダに利用可能な、**nuh_unit_type** が **PREFIX_APS_NUT** 又は **SUFFIX_APS_NUT** に等しい NAL ユニットが存在しないことを指定する。**sps_lmcs_enabled_flag**、[[及び]]**sps_scaling_list_enabled_flag**、**sps_ccalf_enabled_flag**、**ph_num_alf_aps_ids_luma**、**sh_num_alf_aps_ids_luma**、**ph_alf_cb_flag**、**ph_alf_cr_flag**、**sh_alf_cb_flag**、及び **sh_alf_cr_flag** は[[両方とも]] 全て 0 に等しいものとする。**no_aps_constraint_flag** が 0 に等しいことは、そのような制約を課さない。

iii. 例えば、特定の値（例えば 1）に等しい GCI シンタックス要素 **no_aps_constraint_flag** は、*OlsInScope* 内の存在を指定せずに、APS で必要なコーディングツール（例えば、LMCS、スケーリングリスト、CCALF、クロマ ALF、APS によるルマ ALF など）の非活性化を指定する。例えば、JVET-R2001-vA のテキストは次のように変更される（追加された部分は太字の斜体のテキストで強調表示され、削除された部分は右及び左の二重角括弧（例えば、[[]]）で強調表示され、二重角括弧の間に削除されたテキストが表示される）。

no_aps_constraint_flag が 1 に等しいことは、[[**nuh_unit_type** が **PREFIX_APS_NUT** 又は **SUFFIX_APS_NUT** と等しい NAL ユニットが *OlsInScope* に存在しないこと]] **sps_lmcs_enabled_flag** と **sps_scaling_list_enabled_flag** の両方が 0 に等しいことを指定する。**no_aps_constraint_flag** が 0 に等しい場合、このような制約は課されない。

或いは、

【数 7 4】

no_aps_constraint_flag が 1 に等しいことは、[[NAL unit with **nuh_unit_type** が **PREFIX_APS_NUT** 又は **SUFFIX_APS_NUT** に等しい NAL ユニットが *OlsInScope* に存在しないこと、及び]] **sps_lmcs_enabled_flag**、[[及び]] **sps_scaling_list_enabled_flag**、**sps_ccalf_enabled_flag**、**ph_num_alf_aps_ids_luma**、**sh_num_alf_aps_ids_luma**、**ph_alf_cb_flag**、**ph_alf_cr_flag**、**sh_alf_cb_flag**、及び **sh_alf_cr_flag** が[[両方とも]] 全て 0 に等しいことを指定する。**no_aps_constraint_flag** が 0 に等しいことは、そのような制約を課さない。

v. 上記のシンタックス要素（**no_aps_constraint_flag** など）は、使用される / 使用可能な / 存在する APS があるかどうかを指定する例であり、必要に応じて名前を変更できる。

9. 第 8 の問題を解決するための適合ウィンドウパラメータのシグナリングに関して：

a. 一例では、**pic_width_in_luma_samples** が **pic_width_max_in_luma_samples** と等しく、**pic_height_in_luma_samples** が **pic_height_max_in_luma_samples** と等しい場合、PPS の適合ウィンドウパラメータ（すなわち、**pps_conformance_window_flag**、**pps_conf_win_left_offset**、**pps_conf_win_right_offset**、**pps_conf_win_top_offset**、及び **pps_conf_win_bottom_offset**）のシグナリングがスキップされてよい。

i. 一例では、PPS シンタックスにフラグを追加することができ、このフラグの値が X (0

10

20

30

40

50

又は 1) に等しい場合は、pic_width_in_luma_samplesがpic_width_max_in_luma_samplesと等しく、pic_height_in_luma_samplesがpic_height_max_in_luma_samplesと等しいことを指定し、フラグが 1 - Xに等しいことは、pic_width_in_luma_samplesがpic_width_max_in_luma_samplesより小さいか、又はpic_height_in_luma_samplesがpic_height_max_in_luma_samplesより小さいことを指定する。ただし、pic_width_in_luma_samplesがpic_width_max_in_luma_samplesと等しく、かつpic_height_in_luma_samplesがpic_height_max_in_luma_samplesと等しい場合でも、PPSがSPSに解析依存しないように、pic_width_in_luma_samplesとpic_height_in_luma_samplesは依然としてPPSでシグナリングされる必要がある。

更に、上記のフラグがXに等しい場合、PPSの適合ウィンドウパラメータのシグナリング(すなわち、pps_conformance_window_flag、pps_conf_win_left_offset、pps_conf_win_right_offset、pps_conf_win_top_offset、及びpps_conf_win_bottom_offset)はスキップされ、更に、パラメータの値はSPSのパラメータの値と等しいと推定される(すなわち、sps_conformance_window_flag、sps_conf_win_left_offset、sps_conf_win_right_offset、sps_conf_win_top_offset、及びsps_conf_win_bottom_offset)。

10. 第 10 の問題を解決するためのGCI SEによる非活性化制御について：

a. 一例では、SPS/PPS/APS/PH/SHシンタックス要素の値及び/又はそれらの存在に対する制限を表すGCIシンタックス構造に制約フラグ/指示子が追加/シグナリングされてよい。

i. 例えば、特定の値(例えば 1)に等しい制約フラグ/指示子は、Xが示すコーディングツール/機能を制限することを指定する。

ii. 例えば、制約フラグはu(1)コーディングされてよい。

iii. 例えば、制約指示子は、Nが正の定数であるようにu(N)コーディングされてよい。

例えば、Xは、以下のうちの 1 つ以上を含んでよい：

a) sps_ptl_dpb_hrd_params_present_flag(例えば、ptl、dpb、hrdなどの存在の制限)、

b) sps_subpic_info_present_flag(例えば、サブピクチャ、サブピクチャ抽出などの制限)、

c) sps_independent_subpics_flag(例えば、独立/従属サブピクチャ、サブピクチャ抽出などの制限)、

d) sps_subpic_id_mapping_explicitly_signalled_flag(例えば、subpic idマッピングの制限など)、

e) sps_subpic_id_mapping_present_flag(例えば、subpic idマッピングなどの制限)、

f) sps_entropy_coding_sync_enabled_flag(例えば、エントロピーコーディング同期、wppなどの制限)、

g) sps_entry_point_offsets_present_flag(例えば、エントロピーコーディング同期、wppなどの制限)、

h) sps_log2_max_pic_order_cnt_lsb_minus_4(例えば、POC特性などの制限)、

i) sps_poc_msb_cycle_flag(例えば、POC特性などの制限)、

j) sps_sublayer_dpb_params_flag(例えば、サブレイヤhrdなどの制限)、

k) sps_partition_constraints_override_enabled_flag(例えば、SPS以外のSHなどのシンタックス構造におけるパーティション制約付きオーバーライドの制限など)、

l) sps_log2_min_luma_coding_block_size_minus_2(例えば、最小CTBサイズなどの制限)、

m) sps_log2_diff_min_qt_min_cb_intra_slice_luma(例えば、イントラスライスの最小QTサイズなどの制限)、

n) sps_max_mtt_hierarchy_depth_intra_slice_luma(例えば、イントラスライスのMTT深さなどの制限)、

10

20

30

40

50

- o) `sps_log2_diff_min_qt_min_cb_inter_slice` (例えば、インタースライスの最小QTサイズなどの制限)、
- p) `sps_max_mtt_hierarchy_depth_inter_slices` (例えば、インタースライスのMTT深さなどの制限)、
- q) `sps_max_luma_transform_size_64_flag` (例えば、 $64 \times N/N \times 64$ 変換サイズなどの制限)、
- r) `sps_same_qp_table_for_chroma_flag` (例えば、Cb/Crの同一QPテーブルなどに対する制限)、
- s) `sps_log2_transform_skip_max_size_minus2` (例えば、変換スキップの最大ブロックサイズなどの制限)、
- t) `sps_idr_rpl_present_flag` (例えば、IDRピクチャのRPL、ビットストリームのマージ/BEAMingなどの制限)、
- u) `sps_rpl1_same_as_rpl0_flag` (例えば、list 0 と list 1 の同じRPLなどに対する制限)、
- v) `sps_num_ref_pic_lists[i]` (例えば、i番目のRPLエントリの参照ピクチャリストの数の制限)、
- w) `sps_dmvr_control_present_in_ph_flag` (例えば、SPS以外のPHにおけるDMVR制御の制限)、
- x) `sps_mmvd_fullpel_only_flag` (例えば、フルペル/分数ペルMMVDの制限)、
- y) `sps_six_minus_max_num_merge_cand` (例えば、マージの制限)、
- z) `sps_affine_type_flag` (例えば、4パラメータ/6パラメータモデルなどのアフィンタイプの制限)、
- aa) `sps_affine_amvr_enabled_flag` (例えば、AMVRによるアフィンの制限)、
- bb) `sps_prof_control_present_in_ph_flag` (例えば、SPS以外のPHでのPROF制御の制限)、
- cc) `sps_log2_parallel_merge_level_minus2` (例えば、MERの制限)、
- dd) `sps_chroma_horizontal_collocated_flag` (例えば、クロマサブサンプリング方法の制限)、
- ee) `sps_chroma_vertical_collocated_flag` (例えば、クロマサブサンプリング方法の制限)、
- ff) `sps_internal_bit_depth_minus_input_bit_depth` (例えば、内部ビット深度増加の制限)、
- gg) `sps_scaling_matrix_for_lfnst_disabled_flag` (例えば、LFNSTのスケーリングマトリクスの制限)、
- hh) `sps_scaling_matrix_for_alternative_colour_space_disabled_flag` (例えば、代替色空間のためのスケーリングマトリクスの制限)、
- ii) `sps_scaling_matrix_designated_colour_space_flag` (例えば、指定された色空間以外の色空間の拡大縮小マトリクスの制限)、
- jj) `sps_general_hrd_params_present_flag` (例えば、HRDの制限)
- kk) `sps_sublayer_cpb_params_present_flag` (例えば、CPB/HRDの制限)、
- ll) `sps_vui_parameters_present_flag` (例えば、VUIの制限)、
- v. 以上に列挙された名前 (例えば、X) は、特定のコーディングツール/機能/機能に対応するSPS/PPS/PH/SHシンタックス構造内のsignalled/presentシンタックス要素を指定する例であり、必要に応じて名前を変更することができる。

11. 第11の問題を解決するためのローカルデュアルツリーの不完全な記述に関しては、コーディングツリーノードに対して、モードタイプという名前の変数が復号された情報に従って導出されてよい (例えば、VVCのmodeTypeCondition及び/又はmode_constraint_flag/non_inter_flagに従う)。更に、以下が適用されてよい:

- a. 一例として、コーディングツリーノードが特定のモードタイプ (例えば、VVC仕様書のMODE_TYPE_INTRA及び/又はMODE_TYPE_ALL) に属している場合、コーディン

10

20

30

40

50

グツリーノード内のコーディングユニットに対してパレットモード（例えば、MODE_PLT、及び/又は1に等しいpred_mode_plt_flag）を有効にすることができる。

i. 例えば、特定のモードタイプ（例えば、MODE_TYPE_INTRA）を定義して、イントラ（MODE_INTRA）とIBC（MODE_IBC）及びパレット（MODE_PLT）コーディングモードのみをコーディングツリーノード内のコーディングユニットに使用できるかどうかを指定できる。

ii. 例えば、特定のモードタイプ（例えば、MODE_TYPE_INTER）を定義して、コーディングツリーノード内のコーディングユニットにインター（MODE_INTER）コーディングモードのみを使用できるかどうかを指定できる。

iii. 例えば、特定のモードタイプ（例えば、MODE_TYPE_ALL）を定義して、イントラ、IBC、パレット、インターコーディングモードのすべてをコーディングツリーノード内のコーディングユニットに使用できるかどうかを指定できる。

b. 一例では、コーディングツリーノードが特定のモードタイプ（例えば、MODE_TYPE_INTRA及び/又はMODE_TYPE_ALL）に属している場合、コーディングツリーノード内のルマコーディングブロックに対してのみパレットモード（例えば、MODE_PLT及び/又は1に等しいpred_mode_plt_flag）を有効にすることができる。

c. 例えば、コーディングツリーノードにコーディングモードの制限が適用されていない場合（例えば、このコーディングツリーノードのモードタイプがMODE_TYPE_ALLに等しい）、コーディングツリーノード内のコーディングユニットには以下のモードのいずれかを使用することができる：

i. イントラモード（例えば、MODE_INTRAと等しい予測モード）

ii. IBCモード（例えば、MODE_IBCと等しい予測モード）

iii. パレットモード（例えば、MODE_PLTに等しい予測モード、及び/又は1に等しいpred_mode_plt_flag）

iv. インターモード（例えば、MODE_INTERと等しい予測モード）

d. 一例では、コーディングツリーノードにローカルデュアルツリーを適用する場合（例えば、このコーディングツリーノードのモードタイプがMODE_TYPE_INTRAに等しい）、コーディングツリーノード内のコーディングユニットに次のいずれかのモードを使用できる：

i. イントラモード（例えば、MODE_INTRAと等しい予測モード）

ii. IBCモード（例えば、MODE_IBCと等しい予測モード）

iii. パレットモード（例えば、MODE_PLTに等しい予測モード、及び/又は1に等しいpred_mode_plt_flag）

e. 一例では、コーディングツリーノード内のコーディングユニットにローカルデュアルツリーを適用する場合（例えば、このコーディングツリーノードのモードタイプがMODE_TYPE_INTRAに等しい）、このコーディングツリーノード内のルマブロックは、イントラモード及び/又はIBCモード及び/又はパレットモードでコーディングされてよい。

i. 更に、このコーディングツリーノード内のクロマブロックはイントラモード（MODE_INTRA）でコーディングされている。

ii. 例えば、このコーディングツリーノード内のすべてのルマブロックは、イントラモード（MODE_INTRA）でコーディングされてよい。

iii. 例えば、このコーディングツリーノード内のすべてのルマブロックは、IBCモード（MODE_IBC）でコーディングされてよい。

iv. 例えば、このコーディングツリーノード内のすべてのルマブロックはパレットモード（MODE_PLT）でコーディングされてよい。

v. 例えば、このコーディングツリーノード内のルマブロックは、イントラモード（MODE_INTRA）、IBCモード（MODE_IBC）、パレットモード（MODE_PLT）のいずれかでコーディングされたルマブロックと混合コーディングできる。

vi. 例えば、インターモード（MODE_INTER）は、ローカルデュアルツリーコーディングツリーノード（つまり、モードタイプがMODE_TYPE_INTRAと等しいコーディングツ

10

20

30

40

50

リーノード)内のコーディングユニットには使用できない。

f.あるいは、更に、モードタイプが特定の値(例えば、MODE_TYPE_INTRA及び/又はMODE_TYPE_ALL)と等しいコーディングツリーノード内のCUでは、パレットモードが適用されているかどうかの指示が更にシグナリングされてよい。

g.一例では、JVET-R2001-vAのパーティション利用可能性チェックのテキストは次のように変更される(追加されたものは太字の斜体のテキストで強調表示される)：

6.4 利用可能性処理

6.4.1 許可される4分割処理

この処理への入力は：

- ルマサンプル数でのコーディングブロックサイズcbSize；

- マルチタイプツリー深さmttDepth；

- コーディングツリーノードのパーティションにシングルツリー(SINGLE_TREE)又はデュアルツリーのどちらを使用するか、及びデュアルツリーを使用する場合はルマ成分(DUAL_TREE_LUMA)又はクロマ成分(DUAL_TREE_CHROMA)のどちらが現在処理されているかを指定する変数treeType；

【数75】

変数 modeType は、コーディングツリーノードの中のコーディングユニットについて、イントラ(MODE_INTRA)、IBC(MODE_IBC)、パレット(MODE_PLT)及びインターコーディングモードが使用できるかどうか(MODE_TYPE_ALL)、又はイントラ、[[及び]]IBC及びパレットコーディングモードが使用できるかどうか(MODE_TYPE_INTRA)、又はインターコーディングモードのみが使用できるかどうか(MODE_TYPE_INTER)を指定する。

...

6.4.2 許可される2分割処理

この処理への入力は：

...

- コーディングツリーノードのパーティションにシングルツリー(SINGLE_TREE)又はデュアルツリーのどちらを使用するか、及びデュアルツリーを使用する場合はルマ成分(DUAL_TREE_LUMA)又はクロマ成分(DUAL_TREE_CHROMA)のどちらが現在処理されているかを指定する変数treeType；

【数76】

- 変数modeTypeは、コーディングツリーノードの中のコーディングユニットについて、イントラ(MODE_INTRA)、IBC(MODE_IBC)、パレット(MODE_PLT)及びインターコーディングモードが使用できるかどうか(MODE_TYPE_ALL)、又はイントラ、[[及び]]IBC及びパレットコーディングモードが使用できるかどうか(MODE_TYPE_INTRA)、又はインターコーディングモードのみが使用できるかどうか(MODE_TYPE_INTER)を指定する。

....

6.4.3 許可される3分割処理

この処理への入力は：

...

- コーディングツリーノードのパーティションにシングルツリー(SINGLE_TREE)又はデュアルツリーのどちらを使用するか、及びデュアルツリーを使用する場合はルマ成分(DUAL_TREE_LUMA)又はクロマ成分(DUAL_TREE_CHROMA)のどちらが現在処理されているかを指定する変数treeType；

【数77】

10

20

30

40

50

- 変数modeTypeは、コーディングツリーノードの中のコーディングユニットについて、イントラ(MODE_INTRA)、IBC (MODE_IBC)、パレット(MODE_PLT) 及びインターコーディングモードが使用できるかどうか(MODE_TYPE_ALL)、又はイントラ、[[及び]] IBC 及びパレットコーディングモードが使用できるかどうか (MODE_TYPE_INTRA)、又はインターコーディングモードのみが使用できるかどうか (MODE_TYPE_INTER) を指定する。

...

1 2.modeTypeCondition (例えば、VVCで定義されている) が1に等しいときにシンタックス要素がシグナルされてよく、このシンタックス要素を使用して、コーディングツリーノード内のすべてのCUに対して3つの予測モード(例えば、MODE_INTRA、MODE_IBC、MODE_PALETTE)の1つを許可するかどうかを指定する。

10

a.一例では、シンタックス要素はフラグである(例えば、mode_constraint_flag又はnon_inter_flag又はinter_mode_only_flagとして示される)。

b.一例では、シンタックス要素のセマンティクスは次のように定義できる:

non_inter_flagが0に等しいことは、現在のコーディングツリーノード内のコーディングユニットは、インター予測コーディングモードのみを使用できることを指定する。non_inter_flagが1に等しいことは、現在のコーディングツリーノード内のコーディングユニットは、イントラ又はIBC又はパレット予測コーディングモードを使用できることを指定する。

20

或いは、

inter_mode_only_flagが1に等しいことは、現在のコーディングツリーノード内のコーディングユニットは、インター予測コーディングモードのみを使用できることを指定する。inter_mode_only_flagが0に等しいことは、現在のコーディングツリーノード内のコーディングユニットは、イントラ又はIBC又はパレット予測コーディングモードを使用できることを指定する。

c.また、更に、CUに使用される予測モードは、上記のシンタックス要素に加えて、他のシンタックス要素/復号された情報に更に依存してよい。

【0046】

30

図1は、ここに開示される種々の技術が実施され得る例示的なビデオ処理システム1900を示すブロック図である。種々の実装は、システム1900のコンポーネントの一部又は全部を含んでよい。システム1900は、ビデオコンテンツを受信する入力1902を含んでよい。ビデオコンテンツは、生(raw)又は非圧縮フォーマット、例えば8又は10ビット複数成分ピクセル値で受信されてよく、或いは圧縮又は符号化フォーマットであってよい。入力1902は、ネットワークインタフェース、周辺機器バスインタフェース、又はストレージインタフェースを表してよい。ネットワークインタフェースの例は、イーサネット(登録商標)、受動光ネットワーク(passive optical network (PON))等のような有線インタフェース、及びWi-Fi又はセルラインタフェースのような無線インタフェースを含む。

40

【0047】

システム1900は、本願明細書に記載された種々のコーディング又は符号化方法を実施し得るコーディングコンポーネント1904を含んでよい。コーディングコンポーネント1904は、入力1902からコーディングコンポーネント1904の出力へのビデオの平均ビットレートを低減して、ビデオのコーディング表現を生成してよい。コーディング技術は、従って、時に、ビデオ圧縮又はビデオトランスコーディング技術と呼ばれる。コーディングコンポーネント1904の出力は、コンポーネント1906により表されるように、格納されるか、又は通信接続を介して送信されてよい。入力1902で受信された、格納され又は通信されたビットストリーム(又はコーディングされた)表現は、コンポーネント1908により、ディスプレイインタフェース1910へ送信されるピクセル

50

値又は表示可能なビデオを生成するために、使用されてよい。ビットストリーム表現からユーザに閲覧可能なビデオを生成する処理は、時に、ビデオ伸長と呼ばれる。更に、特定のビデオ処理動作は「コーディング」動作又はツールと呼ばれるが、コーディングツール又は動作は、エンコーダにおいて使用され、コーディングの結果を逆にする対応する復号ツール又は動作がデコーダにより実行されることが理解される。

【0048】

周辺機器バスインタフェース又はディスプレイインタフェースの例は、ユニバーサルシリアルバス (universal serial bus (USB)) 又は高解像度マルチメディアインタフェース (high definition multimedia interface (HDMI (登録商標))) 又はディスプレイポート (Displayport)、等を含んでよい。ストレージインタフェースの例は、SATA (serial advanced technology attachment)、PCI、IDE インタフェース、等を含む。本願明細書に記載した技術は、移動電話機、ラップトップ、スマートフォン、又はデジタルデータ処理を実行可能な他の装置、及び/又はビデオディスプレイのような種々の電子装置に実装されてよい。

10

【0049】

図2は、ビデオ処理機器3600のブロック図である。機器3600は、ここに記載した方法のうちの一つ以上を実施するために使用されてよい。機器3600は、スマートフォン、タブレット、コンピュータ、モノのインターネット (Internet of Things (IoT)) 受信機、等において実施されてよい。機器3600は、一つ以上のプロセッサ3602、一つ以上のメモリ3604、及びビデオ処理ハードウェア3606を含んでよい。プロセッサ3602は、本願明細書に記載した一つ以上の方法を実施するよう構成されてよい。メモリ (複数のメモリ) 3604は、本願明細書に記載の方法及び技術を実施するために使用されるデータ及びコードを格納するために使用されてよい。ビデオ処理ハードウェア3606は、ハードウェア回路で、本願明細書に記載した幾つかの技術を実施するために使用されてよい。

20

【0050】

図4は、本開示の技術を利用し得る例示的なビデオコーディングシステム100を示すブロック図である。

【0051】

図4に示されるように、ビデオコーディングシステム100は、ソース装置110と宛先装置120とを含んでよい。ソース装置110は、ビデオ符号化装置と呼ばれてよく、符号化ビデオデータを生成する。宛先装置120は、ビデオ復号装置と呼ばれてよく、ソース装置110により生成された符号化ビデオデータを復号してよい。

30

【0052】

ソース装置110は、ビデオソース112、ビデオエンコーダ114、及び入力/出力 (I/O) インタフェース116を含んでよい。

【0053】

ビデオソース112は、ビデオキャプチャ装置のようなソース、ビデオコンテンツプロバイダからビデオデータを受信するインタフェース、及び/又はビデオデータを生成するコンピュータグラフィックシステム、又はそのようなソースの組合せを含んでよい。ビデオデータは、一つ以上のピクチャを含んでよい。ビデオエンコーダ114は、ビデオソース112からのビデオデータを符号化して、ビットストリームを生成する。ビットストリームは、ビデオデータのコーディング表現を形成するビットのシーケンスを含んでよい。ビットストリームは、コーディングピクチャ及び関連データを含んでよい。コーディングピクチャは、ピクチャのコーディング表現である。関連データは、シーケンスパラメータセット、ピクチャパラメータセット、及び他のシンタックス構造を含んでよい。I/Oインタフェース116は、変調器/復調器 (モデム) 及び/又は送信機を含んでよい。符号化ビデオデータは、I/Oインタフェース116を介してネットワーク130aを通じて、宛先装置120へ直接送信されてよい。符号化ビデオデータは、宛先装置120によるアクセスのために、記憶媒体/サーバ130bに格納されてもよい。

40

50

及びインター予測 (combination of intra and inter predication (CIIP)) モードを選択してよい。CIIPモードでは、予測はインター予測信号及びイントラ予測信号に基づく。モード選択ユニット203は、インター予測の場合に、ブロックについて動きベクトルの解像度 (例えば、サブピクセル又は整数ピクセル精度) を選択してもよい。

【0064】

現在ビデオブロックに対してインター予測を実行するために、動き推定ユニット204は、バッファ213からの1つ以上の参照フレームを現在ビデオブロックと比較することにより、現在ビデオブロックについて動き情報を生成してよい。動き補償ユニット205は、動き情報及び現在ビデオブロックに関連するピクチャ以外のバッファ213からのピクチャの復号サンプルに基づき、現在ビデオブロックについて予測ビデオブロックを決定してよい。

10

【0065】

動き推定ユニット204及び動き補償ユニット205は、例えば現在ビデオブロックがIスライス、Pスライス、又はBスライスかに依存して、現在ビデオブロックについて異なる動作を実行してよい。

【0066】

幾つかの例では、動き推定ユニット204は、現在ビデオブロックについて片方向予測を実行してよく、動き推定ユニット204は、現在ビデオブロックの参照ビデオブロックについて、リスト0又はリスト1の参照ピクチャを検索してよい。動き推定ユニット204は、次に、参照ビデオブロックを含むリスト0又はリスト1内の参照ピクチャを示す参照インデックス、及び現在ビデオブロックと参照ビデオブロックとの間の空間変位を示す動きベクトルを生成してよい。動き推定ユニット204は、参照インデックス、予測方向指示子、及び動きベクトルを、現在ビデオブロックの動き情報として出力してよい。動き補償ユニット205は、現在ビデオブロックの動き情報により示される参照ビデオブロックに基づき、現在ブロックの予測ビデオブロックを生成してよい。

20

【0067】

他の例では、動き推定ユニット204は、現在ビデオブロックについて双方向予測を実行してよく、動き推定ユニット204は、現在ビデオブロックの参照ビデオブロックについてリスト0内の参照ピクチャを検索してよく、現在ビデオブロックの別の参照ビデオブロックについてリスト1内の参照ピクチャを検索してよい。動き推定ユニット204は、次に、参照ビデオブロックを含むリスト0又はリスト1内の参照ピクチャを示す参照インデックス、及び参照ビデオブロックと現在ビデオブロックとの間の空間変位を示す動きベクトルを生成してよい。動き推定ユニット204は、現在ビデオブロックの動き情報として、参照インデックス及び現在ビデオブロックの動きベクトルを出力してよい。動き補償ユニット205は、現在ビデオブロックの動き情報により示される参照ビデオブロックに基づき、現在ビデオブロックの予測ビデオブロックを生成してよい。

30

【0068】

幾つかの例では、動き推定ユニット204は、デコーダの復号処理のために動き情報の完全なセットを出力してよい。

【0069】

幾つかの例では、動き推定ユニット204は、現在ビデオ動き情報の完全なセットを出力しなくてよい。むしろ、動き推定ユニット204は、別のビデオブロックの動き情報を参照して、現在ビデオブロックの動き情報をシグナリングしてよい。例えば、動き推定ユニット204は、現在ビデオブロックの動き情報が、近隣ビデオブロックの動き情報と十分に類似していることを決定してよい。

40

【0070】

一例では、動き推定ユニット204は、現在ビデオブロックに関連付けられたシンタックス構造の中で、現在ビデオブロックが別のビデオブロックと同じ動き情報を有することをビデオデコーダ300に示す値を示してよい。

【0071】

50

別の例では、動き推定ユニット 204 は、現在ビデオブロックに関連付けられたシンタックス構造の中で、別のビデオブロック及び動きベクトル差 (motion vector difference (MVD)) を識別してよい。動きベクトル差は、現在ビデオブロックの動きベクトルと示されたビデオブロックの動きベクトルとの間の差を示す。ビデオデコーダ 300 は、示されたビデオブロックの動きベクトル及び動きベクトル差を使用して、現在ビデオブロックの動きベクトルを決定してよい。

【0072】

上述のように、ビデオエンコーダ 200 は、動きベクトルを予測的にシグナリングしてよい。ビデオエンコーダ 200 により実施され得る予測的シグナリング技術の 2 つの例は、高度動きベクトル予測 (advanced motion vector predication (AMVP)) 及びマージモードシグナリングを含む。

10

【0073】

イントラ予測ユニット 206 は、現在ビデオブロックに対してイントラ予測を実行してよい。イントラ予測ユニット 206 が現在ビデオブロックに対してイントラ予測を実行するとき、イントラ予測ユニット 206 は、同じピクチャ内の他のビデオブロックの復号サンプルに基づき、現在ビデオブロックの予測データを生成してよい。現在ビデオブロックの予測データは、予測ビデオブロック及び種々のシンタックス要素を含んでよい。

【0074】

残差生成ユニット 207 は、現在ビデオブロックの予測ビデオブロックを現在ビデオブロックから減算することにより (例えば、マイナス符号により示される)、現在ビデオブロックの残差データを生成してよい。現在ビデオブロックの残差データは、現在ビデオブロック内のサンプルの異なるサンプル成分に対応する残差ビデオブロックを含んでよい。

20

【0075】

他の例では、現在ビデオブロックについて、例えばスキップモードでは現在ビデオブロックの残差データが存在しなくてよく、残差生成ユニット 207 は減算動作を実行しなくてよい。

【0076】

変換処理ユニット 208 は、現在ビデオブロックに関連付けられた残差ビデオブロックに 1 つ以上の変換を適用することにより、現在ビデオブロックについて 1 つ以上の変換係数ビデオブロックを生成してよい。

30

【0077】

変換処理ユニット 208 が現在ビデオブロックに関連付けられた変換係数ビデオブロックを生成した後に、量子化ユニット 209 は、現在ビデオブロックに関連付けられた 1 つ以上の量子化パラメータ (quantization parameter (QP)) に基づき、現在ビデオブロックに関連付けられた変換係数ビデオブロックを量子化してよい。

【0078】

逆量子化ユニット 210 及び逆変換ユニット 211 は、各々変換係数ビデオブロックに逆量子化及び逆変換を適用して、変換係数ビデオブロックから残差ビデオブロックを再構成してよい。再構成ユニット 212 は、再構成残差ビデオブロックを、予測ユニット 202 により生成された 1 つ以上の予測ビデオブロックからの対応するサンプルに加算して、バッファ 213 に格納するために現在ビデオブロックに関連付けられた再構成ビデオブロックを生成してよい。

40

【0079】

再構成ユニット 212 がビデオブロックを再構成した後に、ループフィルタリング動作が実行されて、ビデオブロック内のビデオブロッキングアーチファクトを低減してよい。

【0080】

エントロピー符号化ユニット 214 は、ビデオエンコーダ 200 の他の機能コンポーネントからデータを受信してよい。エントロピー符号化ユニット 214 がデータを受信すると、エントロピー符号化ユニット 214 は、1 つ以上のエントロピー符号化動作を実行して、エントロピー符号化データを生成し、エントロピー符号化データを含むビットストリ

50

ームを出力してよい。

【 0 0 8 1 】

開示した技術の幾つかの実施形態は、ビデオ処理ツール又はモードを有効にすることの決定又は判断を行うことを含む。例では、ビデオ処理ツール又はモードが有効にされると、エンコーダは、ビデオのブロックの処理において該ツール又はモードを使用又は実施するが、必ずしもツール又はモードの使用に基づき結果として生じるビットストリームを変更する必要はない。つまり、ビデオのブロックからビデオのビットストリーム（又はビットストリーム表現）への変換は、決定又は判断に基づき有効にされるとき、ビデオ処理ツール又はモードを使用する。別の例では、ビデオ処理ツール又はモードが有効にされると、デコーダは、ビットストリームがビデオ処理ツール又はモードに基づき変更されているという知識により、ビットストリームを処理する。つまり、ビデオのビットストリームからビデオのブロックへの変換は、決定又は判断に基づき有効にされたビデオ処理ツール又はモードを使用して実行される。

10

【 0 0 8 2 】

図 6 は図 4 に示したシステム 1 0 0 の中のビデオデコーダ 1 1 4 であってよいビデオデコーダ 3 0 0 の例を示すブロック図である。

【 0 0 8 3 】

ビデオデコーダ 3 0 0 は、本開示の技術のうちのいずれか又は全部を実行するよう構成されてよい。図 6 の例では、ビデオデコーダ 3 0 0 は複数の機能コンポーネントを含む。本開示に記載した技術は、ビデオデコーダ 3 0 0 の種々のコンポーネントの間で共有されてよい。幾つかの例では、プロセッサは、本開示に記載した技術のうちのいずれか又は全部を実行するよう構成されてよい。

20

【 0 0 8 4 】

図 6 の例では、ビデオデコーダ 3 0 0 は、エントロピー復号ユニット 3 0 1、動き補償ユニット 3 0 2、イントラ予測ユニット 3 0 3、逆量子化ユニット 3 0 4、逆変換ユニット 3 0 5、及び再構成ユニット 3 0 6、及びパツファ 3 0 7 を含む。ビデオデコーダ 3 0 0 は、幾つかの例では、ビデオエンコーダ 2 0 0（図 5）に関して説明した符号化経路に対して通常相互的な復号経路を実行してよい。

【 0 0 8 5 】

エントロピー復号ユニット 3 0 1 は、符号化ビットストリームを読み出してよい。符号化ビットストリームは、エントロピー符号化ビデオデータ（例えば、ビデオデータの符号化ブロック）を含んでよい。エントロピー復号ユニット 3 0 1 は、エントロピー符号化ビデオデータを復号し、エントロピー復号ビデオデータから、動き補償ユニット 3 0 2 が、動きベクトル、動きベクトル制度、参照ピクチャリストインデックス、及び他の動き情報を含む動き情報を決定してよい。動き補償ユニット 3 0 2 は、例えば、AMVP 及びマージモードを実行することにより、このような情報を決定してよい。

30

【 0 0 8 6 】

動き補償ユニット 3 0 2 は、場合によっては補間フィルタに基づき補間を実行することにより、動き補償ブロックを生成してよい。サブピクセル制度で使用されるべき補間フィルタの識別子は、シンタックス要素に含まれてよい。

40

【 0 0 8 7 】

動き補償ユニット 3 0 2 は、参照ブロックのサブ整数ピクセルの補間値を計算するためにビデオブロックの符号化中にビデオエンコーダ 2 0 0 により使用されるような補間フィルタを使用してよい。動き補償ユニット 3 0 2 は、受信したシンタックス情報に従い、ビデオエンコーダ 2 0 0 により使用される補間フィルタを決定し、補間フィルタを使用して予測ブロックを生成してよい。

【 0 0 8 8 】

動き補償ユニット 3 0 2 は、シンタックス情報の一部を使用して、符号化ビデオシーケンスのフレーム及び/又はスライスを符号化するために使用されるブロックのサイズ、符号化ビデオシーケンスのピクチャの各マクロブロックがどのようにパーティションされる

50

かを記述するパーティション情報、各パーティションがどのように符号化されるかを示すモード、インター符号化ブロック毎の1つ以上の参照フレーム（及び参照フレームリスト）、及び符号化ビデオシーケンスを復号するための他の情報を決定してよい。

【0089】

イントラ予測ユニット303は、例えばビットストリーム内で受信したイントラ予測モードを使用して、空間的に隣接するブロックから予測ブロックを形成してよい。逆量子化ユニット303は、ビットストリーム内で提供され、エントロピー復号ユニット301により復号された量子化されたビデオブロック係数を逆量子化、つまり量子化解除する。逆変換ユニット303は、逆変換を適用する。

【0090】

再構成ユニット306は、残差ブロックを、動き補償ユニット202又はイントラ予測ユニット303により生成された対応する予測ブロックと加算して、復号ブロックを形成してよい。望ましい場合には、ブロックアーチファクトを除去するために復号ブロックをフィルタリングするデブロッキングフィルタも適用されてよい。復号ビデオブロックは、次に、バッファ307に格納されて、後の動き補償/イントラ予測のために参照ブロックを提供し、更にディスプレイ装置上で提示するために復号ビデオを生成する。

【0091】

幾つかの実施形態による好ましいソリューションのリストが次に提供される。

【0092】

以下のソリューションは、前の章で議論した技術の例示的な実施形態（例えば、項目1）を示す。

【0093】

（ソリューション1）ビデオ処理方法300（例えば図3に示された方法300）であって、ビデオのビデオ領域と前記ビデオのコーディング表現との間の変換を実行するステップ（302）を含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、前記ビデオ内の色成分のスケーリングリストが、シーケンスパラメータセットのシンタックスフィールド値とは独立して適応パラメータセットに含まれるかどうかをフラグが示すことを指定する、方法。

【0094】

（ソリューション2）前記フォーマットルールは、シーケンスパラメータセットを識別するフィールドが前記適応パラメータセットに含まれることを指定する、ソリューション1に記載の方法。

【0095】

（ソリューション3）前記フォーマットルールは、前記適応パラメータセットと、前記コーディング表現の前記スケーリングリストへの包含を制御するシーケンスパラメータセットのビデオパラメータセット又はピクチャパラメータセットと、の間の暗黙的關係を指定する、ソリューション1に記載の方法。

【0096】

（ソリューション4）前記フォーマットルールは、前記変換の際に使用されるユーザ定義された又は明示的なスケーリングリストの包含のためのフォーマットを指定する、ソリューション1～3のいずれかに記載の方法。

【0097】

（ソリューション5）前記フォーマットルールは、前記コーディング表現への前記フラグの包含が、クロマ成分のアレイタイプを示すシンタックス要素の包含と独立であることを指定する、ソリューション1～4のいずれかに記載の方法。

【0098】

（ソリューション6）前記フラグは、前記スケーリングリストが含まれることを示し、前記クロマ成分の前記アレイタイプを示す前記シンタックス要素は0に設定される、ソリューション5に記載の方法。

【0099】

10

20

30

40

50

(ソリューション7) 前記フラグは、前記スケーリングリストが含まれないことを示し、前記クロマ成分の前記アレイタイプを示す前記シンタックス要素は1に設定される、ソリューション5に記載の方法。

【0100】

(項8) 前記フォーマットルールは、前記フラグは、ピクチャヘッダ又はスライスヘッダに依存する制約ルールにより制約されることを指定する、ソリューション1に記載の方法。

【0101】

以下のソリューションは、前の章で議論した技術の例示的な実施形態(例えば、項目2)を示す。

【0102】

(ソリューション9) ビデオ処理の方法であって、ビデオのビデオ領域と前記ビデオ領域のコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、各適応パラメータセットについて、前記ビデオのクロマ制約によりクロマ関連シンタックス要素が省略されるように、1つ以上の適応パラメータセットが前記コーディング表現に含まれることを指定する、方法。

【0103】

(ソリューション10) 適応パラメータセット毎に、シンタックス要素は、クロマ関連シンタックス要素が前記適応パラメータセットに含まれるかどうかをシグナリングする、ソリューション9に記載の方法。

【0104】

(ソリューション11) 前記フォーマットルールは、前記クロマ制約が前記ビデオのコーディング表現の中のクロマの存在を示す場合にのみ、ピクチャヘッダ又はスライスヘッダ又は適応パラメータセットにクロマ関連フィールドが条件付きで含まれることを指定する、ソリューション9に記載の方法。

【0105】

以下のソリューションは、前の章で議論した技術の例示的な実施形態(例えば、項目3)を示す。

【0106】

(ソリューション12) 前記クロマ制約は、クロマアレイタイプが0に等しいことである、ソリューション9~11のいずれかに記載の方法。

【0107】

(ソリューション13) 前記クロマ制約は、前記ビデオのフォーマットが4:0:0に等しいことである、ソリューション9~11のいずれかに記載の方法。

【0108】

以下のソリューションは、前の章で議論した技術の例示的な実施形態(例えば、項目4)を示す。

【0109】

(ソリューション14) ビデオ処理の方法であって、1つ以上のビデオユニットを含む1つ以上のビデオ領域を含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、第1変換コーディングシンタックスフィールドが、ビデオ領域のビデオユニットのレベルの前記コーディング表現に含まれるかどうか、及び/又はその値が、前記ビデオ領域のレベルの第2変換コーディングシンタックスフィールドの値に依存するかどうかを指定する、方法。

【0110】

(ソリューション15) 前記第1変換コーディングシンタックスフィールドは、`slice_ts_residual_coding_disabled_flag`であり、前記第2変換コーディングシンタックスフィールドは、`sps_transform_skip_enabled_flag`である、ソリューション14に記載の方法。

10

20

30

40

50

【 0 1 1 1 】

以下のソリューションは、前の章で議論した技術の例示的な実施形態（例えば、項目 5）を示す。

【 0 1 1 2 】

（ソリューション 16）ビデオ処理方法であって、1つ以上のビデオ領域を含むビデオであって、各ビデオ領域が1つ以上のビデオユニットを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、ビデオユニットレベルのフラグが、前記変換に対して量子化パラメータの差分シグナリングが有効かどうかを制御することを指定する、方法。

10

【 0 1 1 3 】

（ソリューション 17）前記ビデオユニットレベルのフラグは、コーディングユニット又は変換ユニットレベルの第2フラグが、差分量子化パラメータシグナリングの使用をシグナリングするために包含されるかどうかを制御する、ソリューション 16に記載の方法。

【 0 1 1 4 】

以下のソリューションは、前の章で議論した技術の例示的な実施形態（例えば、項目 6）を示す。

【 0 1 1 5 】

（ソリューション 18）ビデオ処理方法であって、1つ以上のビデオ領域を含むビデオであって、各ビデオ領域が1つ以上のビデオユニットを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを、を含む通信フォーマットルールに従い、前記フォーマットルールは、サブピクチャの数を示すピクチャレベルの第1フラグと、サブピクチャ内のスライスを示すサブピクチャレベルの第2フラグの解釈を指定する、方法。

20

【 0 1 1 6 】

（ソリューション 19）前記フォーマットルールは、前記第1フラグが1に設定され、前記第1フラグが1に設定される場合に、前記ピクチャ内の少なくとも1つのサブピクチャが1より多くのスライスを含むことを指定する、ソリューション 18に記載の方法。

【 0 1 1 7 】

（ソリューション 20）前記フォーマットルールは、前記第1フラグが0であること及び各ピクチャ内に単一のスライスが存在することに起因して、前記第2フラグが1に設定されなければならないことを指定する、ソリューション 18に記載の方法。

30

【 0 1 1 8 】

以下のソリューションは、前の章で議論した技術の例示的な実施形態（例えば、項目 7）を示す。

【 0 1 1 9 】

（ソリューション 21）ビデオ処理の方法であって、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、ビデオピクチャに関連するピクチャパラメータセット内のフィールドが、異なる高さ又は幅の複数のタイル行又は列に分割されることを指定する、方法。

40

【 0 1 2 0 】

（ソリューション 22）前記コーディング表現の中の第2フィールドは、前記ビデオピクチャのタイルが異なる高さを有する複数のスライス行に分割されるかどうかを示す、ソリューション 21に記載の方法。

【 0 1 2 1 】

（ソリューション 23）ぜんき 第2フィールドは、前記複数のスライス行のスライス高さを示す、ソリューション 22に記載の方法。

【 0 1 2 2 】

50

以下のソリューションは、前の章で議論した技術の例示的な実施形態（例えば、項目 8）を示す。

【0123】

（ソリューション 24）ビデオ処理の方法であって、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、適応パラメータセットが適応ループフィルタリングの表示を除外する場合のビデオ領域への適応ループフィルタリングの適用可能性が第2ルールに基づくことを指定する、方法。

【0124】

（ソリューション 25）前記第2ルールは、前記ビデオ領域について適応ループフィルタリングが無効にされることを指定する、ソリューション 24に記載の方法。

【0125】

（ソリューション 26）前記第2ルールは、シーケンスパラメータセットレベルのフラグの値に基づき、適応ループフィルタリングが条件付きで許可されることを指定する、ソリューション 24に記載の方法。

【0126】

以下のソリューションは、前の章で議論した技術の例示的な実施形態（例えば、項目 9）を示す。

【0127】

（ソリューション 27）ビデオ処理の方法であって、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、幅と高さがビデオの最大幅と最大高さであるピクチャについて、ピクチャパラメータセット内の適合ウィンドウパラメータの明示的なシグナリングをスキップすることを指定する、方法。

【0128】

（ソリューション 28）前記フォーマットルールは、前記明示的なシグナリングがスキップされる場合に、前記幅及び前記高さが最大幅及び最大高さに等しいかどうかを示すフラグを含むことを更に指定する、ソリューション 27に記載の方法。

【0129】

以下のソリューションは、前の章で議論した技術の例示的な実施形態（例えば、項目 10）を示す。

【0130】

（ソリューション 29）ビデオ処理の方法であって、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、一般制約情報（GCI）シンタックス構造内のシンタックスフィールドが、前記コーディング表現内のパラメータセット又はヘッダ内の1つ以上のシンタックス要素の値に対する1つ以上の制約を制御することを指定する、方法。

【0131】

（ソリューション 30）前記パラメータセットは、シーケンスパラメータセット又はピクチャパラメータセット又は適応パラメータセットに対応する、ソリューション 29に記載の方法。

【0132】

（ソリューション 31）前記ヘッダは、ピクチャヘッダ又はスライスヘッダに対応する、ソリューション 29～30のいずれかに記載の方法。

【0133】

以下のソリューションは、前の章で議論した技術の例示的な実施形態（例えば、項目 1

10

20

30

40

50

1)を示す。

【0134】

(ソリューション32)ビデオ処理の方法であって、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、前記ビデオのコーディングツリーノードについて、ビデオユニットに対応する復号情報がルールに従って導出可能な変数の値を制御することを指定する、方法。

【0135】

(ソリューション33)前記変数は、前記ビデオユニットのパレットコーディングモードに関連する係数である、ソリューション32に記載の方法。

10

【0136】

以下のソリューションは、前の章で議論した技術の例示的な実施形態(例えば、項目12)を示す。

【0137】

(ソリューション34)ビデオ処理の方法であって、1つ以上のビデオピクチャを含むビデオであって、各ビデオピクチャが1つ以上のスライス及び/又は1つ以上のタイルを含むビデオと、前記ビデオのコーディング表現との間の変換を実行するステップを含み、前記コーディング表現はフォーマットルールに従い、前記フォーマットルールは、前記コーディング表現が、前記コーディングツリーノード内のビデオユニットの表現に対して3つの予測モードのいずれかが許可されるかどうかを示す値を有するコーディングツリーノードのシンタックス要素を含むことを指定する、方法。

20

【0138】

(ソリューション35)前記3つの予測モードは、イントラコーディングモード、パレットコーディングモード、及びイントラブロックコピーモードを含む、ソリューション34に記載の方法。

【0139】

(ソリューション36)前記ビデオ領域はビデオピクチャを含む、ソリューション1~35のいずれかに記載の方法。

【0140】

(ソリューション37)前記ビデオユニットは、ビデオスライス又はビデオコーディングツリーユニットを含む、ソリューション1~6のいずれかに記載の方法。

30

【0141】

(ソリューション38)前記変換は、前記ビデオを前記コーディング表現に符号化することを含む、ソリューション1~37のいずれかに記載の方法。

【0142】

(ソリューション39)前記変換は、前記コーディング表現を復号して前記ビデオのピクセル値を生成することを含む、ソリューション1~37のいずれかに記載の方法。

【0143】

(ソリューション40)ビデオ復号機器であって、ソリューション1~39のうちの1つ以上に記載の方法を実施するよう構成されるプロセッサを含むビデオ復号機器。

40

【0144】

(ソリューション41)ビデオ符号化機器であって、ソリューション1~39のうちの1つ以上に記載の方法を実施するよう構成されるプロセッサを含むビデオ符号化機器。

【0145】

(ソリューション42)格納されたコンピュータコードを有するコンピュータプログラムプロダクトであって、前記コードは、プロセッサにより実行されると、前記プロセッサにソリューション1~39のいずれかに記載の方法を実施させる、コンピュータプログラムプロダクト。

【0146】

50

(ソリューション 43) 本願明細書に記載された方法、機器、又はシステム。

【0147】

図7は、ビデオ処理の例示的な方法700のフローチャートである。操作702は、ルールに従って、1つ以上のビデオピクチャを含むビデオと、前記ビデオのビットストリームとの間の変換を実行するステップを含み、前記ルールは、一般的な制約情報シンタックス構造の第1シンタックス要素が、パラメータセット又はヘッダ内の1つ以上のシンタックス要素の存在又は前記1つ以上のシンタックス要素の1つ以上の値を制御することを指定する。

【0148】

方法700の幾つかの実施形態では、パラメータセットは、シーケンスパラメータセット又はピクチャパラメータセット又は適応パラメータセットに対応する。方法700の幾つかの実施形態では、ヘッダはピクチャヘッダ又はスライスヘッダを含む。方法700の幾つかの実施形態では、第1シンタックス要素は、1ビットを使用する符号なし整数を用いてコーディングされる、¥。方法700の幾つかの実施形態では、第1シンタックス要素は、Nビットを使用する符号なし整数を用いてコーディングされ、Nは正の定数値である。方法700の幾つかの実施形態では、ルールは、第1シンタックス要素が第1値を有することに応答して、コーディングツリー又はコーディング特徴に関連付けられている第2シンタックス要素が第2値に制約されることを指定する。方法700の幾つかの実施形態では、第1値は、制約が第2シンタックス要素に課されることを指定する1に等しい。方法700の幾つかの実施形態では、ルールは、第1シンタックス要素が第3値を有することに応答して、コーディングツリー又はコーディング特徴に関連付けられている第2シンタックス要素が制約されないことを指定する。方法700の幾つかの実施形態では、第3値は、第2シンタックス要素に課される制約が存在しないことを指定する1に等しい。

【0149】

方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、プロファイル、ティア、及びレベルシンタックス構造、復号ピクチャバッファパラメータシンタックス構造、並びにタイミング及び仮説参照デコーダパラメータシンタックス構造が、シーケンスパラメータセット内に存在するかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、ルマサンプルの中の最大変換サイズが64に等しいかどうかを指定する。方法700の幾つかの実施形態では、第2値は、ルマサンプルの中の最大変換サイズが32に等しいことを指定する0に等しい。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、サブピクチャ情報がコーディングレイヤビデオシーケンスに存在するかどうか、及び1つより多くのサブピクチャがコーディングレイヤビデオシーケンスの各ピクチャに含まれるかどうかを指定する。方法700の幾つかの実施形態では、第2値は、サブピクチャ情報がコーディングレイヤビデオシーケンスについて存在しないこと、及び1つのみのサブピクチャがコーディングレイヤビデオシーケンスの各ピクチャに含まれることを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、コーディングレイヤビデオシーケンス内の全部のサブピクチャ境界が、ピクチャ境界として扱われるかどうか、及びサブピクチャ境界に跨がるループフィルタリング操作が存在しないかどうかを指定する。

【0150】

方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、サブピクチャ識別子マッピングが、コーディングレイヤビデオシーケンスにより参照されるシーケンスパラメータセット又はピクチャパラメータセットのいずれかで明示的にシグナリングされるかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、別のフラグが1に等しい値を有する場合に、サブピクチャ識別子マッピングが、コーディングレイヤビデオシーケンスにより参照されるシーケンスパラメータセット又はピクチャパラメータセットのいずれかでシグナリングされるかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、コンテキスト変数の特定の同期処理が、シーケンスパラメータセットを参照する各ビデオピクチャ内の各タイル内の

10

20

30

40

50

コーディングツリーブロック行の第1コーディングツリーブロックを含むコーディングツリーユニットを復号する前に、呼び出されるかどうかを指定し、第2シンタックス要素の第2値は、コンテキスト変数の特定の記憶処理が、シーケンスパラメータセットを参照する各ビデオピクチャ内の各タイル内のコーディングツリーブロック行の第1コーディングツリーブロックを含むコーディングツリーユニットを復号した後に呼び出されるかどうかを指定する。

【0151】

方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、タイルのエントリポイント又はタイル固有コーディングツリーユニット行が、シーケンスパラメータセットを参照するビデオピクチャのスライスヘッダに存在するかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、ピクチャオーダカウン

10

の復号処理に使用される変数MaxPicOrderCntLsbの第3値に等しく、以下の通りである：

$$\text{MaxPicOrderCntLsb} = 2 \left(\text{the third value} + 4 \right)$$
 方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、ph_poc_msb_cycle_present_flagシンタックス要素が、シーケンスパラメータセットを参照するピクチャヘッダに存在するかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、sps_max_sublayers_minus1が0より大きいとき、両端を含む0からsps_max_sublayers_minus1-1の範囲のiについて、シーケンスパラメータセット内のadpb_parameters()シンタックス構造の中のdpb_max_dec_pic_buffering_minus1[i]、dpb_max_num_reorder_pics[i]、及びdpb_max_latency_increase_plus1[i]シンタックス要素の存在を制御する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、ph_partition_constraints_override_flagが、シーケンスパラメータセットを参照するピクチャヘッダシンタックス構造に存在するかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値に2を加えたものは、最小ルマコーディングブロックサイズを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、コーディングツリーユニットの4分木分割により生じたルマリーフブロックのルマサンプルのうちの最小サイズの底2の対数と、sh_slice_typeがシーケンスパラメータセットを参照する2(1)に等しいスライスのルマコーディングユニットのルマサンプルのうちの最小コーディングブロックサイズの底2の対数との間のデフォルトの差を指定する。

20

30

【0152】

方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、sh_slice_typeがシーケンスパラメータセットを参照する2(1)に等しいスライスの4分木リーフのマルチタイプツリー分割により生じたコーディングユニットのデフォルト最大階層構造深さを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、コーディングツリーユニットの4分木分割により生じたルマリーフブロックのルマサンプルのうちの最小サイズの底2の対数と、sh_slice_typeがシーケンスパラメータセットを参照する0(B)又は1(P)に等しいスライスのルマコーディングユニットのルマサンプルのうちの最小ルマコーディングブロックサイズの底2の対数との間のデフォルトの差を指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、sh_slice_typeがシーケンスパラメータセットを参照する0(B)又は1(P)に等しいスライスの4分木リーフのマルチタイプツリー分割により生じたコーディングユニットのデフォルト最大階層構造深さを指定する。

40

【0153】

方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、(1)シーケンスパラメータセットが、クロマ残差に及びsps_joint_cbr_enabled_flagが1に等しい場合には共同クロマ残差に適用される1つのみのクロマQPマッピングを含むかどうか、及び(2)sps_joint_cbr_enabled_flagが1に等しいとき、シーケンスパラメータセットが、複数のクロマQPマッピングテーブルを含むかどうか、を指定し、複数のクロマQPマッピングテーブルは、2つのクロマビデオ成分のための2つのQPマッピングテーブル

50

と、共同クロマ残差のための1つのQPマッピングテーブルとを含む。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、変換スキップ(TS)操作のために使用される最大ブロックサイズを指定し、最大ブロックサイズは両端を含む0~3の範囲である。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、参照ピクチャリスト(RPL)シンタックス要素が、nal_unit_typeがIDR_N_LP又はIDR_W_RADLに等しいスライスのスライスヘッダに存在するかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、sps_num_ref_pic_lists[1]及びref_pic_list_struct(1,rplsIdx)が存在しないこと、及び以下が適用されることを指定する：sps_num_ref_pic_lists[1]の第3値が、sps_num_ref_pic_lists[0]の第4値と等しいと推定され、0からsps_num_ref_pic_lists[0]-1の範囲のrplsIdxについて、ref_pic_list_struct(1,rplsIdx)内のシンタックス要素の各々の値が、ref_pic_list_struct(0,rplsIdx)内の対応するシンタックス要素の別の値と等しいと推定される。

10

【0154】

方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、listIdxがシーケンスパラメータセットに含まれるiと等しいref_pic_list_struct(listIdx,rplsIdx)シンタックス構造の数を指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、ph_dmvr_disabled_flagが、シーケンスパラメータセットを参照するピクチャヘッダに存在するかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、動きベクトル差によるマージモードが、ビデオの現在ビデオピクチャに対して整数サンプル精度又は分数サンプル精度を使用するかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、シーケンスパラメータセットでサポートされるマージ動きベクトル予測(MVP)候補の最大数を6から減じたものを指定する。

20

【0155】

方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、予測サンプルを生成するために、4パラメータアフィンモデルに基づく補償、又は6パラメータアフィンモデルに基づく動き補償が使用されることを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、適応動きベクトル差分解能が、コーディングビデオレイヤシーケンスに対して有効かどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、ph_prof_disabled_flagが、シーケンスパラメータセットを参照するピクチャヘッダに存在するかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値に2を加えたものは、空間マージ候補の導出処理で及び履歴に基づく動きベクトル予測子リストの更新処理の呼び出しを制御するために使用される変数Log2ParMrgLevelの値を指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、対応するルマサンプル位置に対して水平シフトされない又は対応するルマサンプル位置に対してルマサンプル単位で0.5だけ右にシフトされない、クロマサンプル位置のために設計される方法で、予測処理が動作するかどうかを指定する。

30

【0156】

方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、対応するルマサンプル位置に対して垂直シフトされない又は対応するルマサンプル位置に対してルマサンプル単位で0.5だけ下にシフトされない、クロマサンプル位置のために設計される方法で、予測処理が動作するかどうかを指定する。方法700の幾つかの実施形態ではシンタックス要素の第2値は、内部ビット深度の増大に対する制約を指定する。方法700の幾つかの実施形態ではシンタックス要素の第2値は、スケーリングマトリクスが、コーディングレイヤビデオシーケンスの低周波非分離変換(LFNST)によりコーディングされたブロックに対して無効であるかどうかを指定する。方法700の幾つかの実施形態ではシンタックス要素の第2値は、コーディングレイヤビデオシーケンスのスケーリングマトリクスが無効であり、現在コーディングユニットの復号残差が色空間変換を用いて適用されるとき、コーディングユニットのビデオブロックに適用されないことを指定する。方法7

40

50

00の幾つかの実施形態ではシンタックス要素の第2値は、スケーリングマトリクス第1色空間が、色空間変換を使用しない又は復号残差に対して色空間変換を使用する第2色空間変換であることを指定する。

【0157】

方法700の幾つかの実施形態ではシンタックス要素の第2値は、仮説参照デコーダ(HRD)に対する制約を指定する。方法700の幾つかの実施形態ではシンタックス要素の第2値は、シーケンスパラメータセット内のols_timing_hrd_parameters()シンタックス構造が、両端を含む0~sps_max_sublayers_minus1の範囲であるか又はsps_max_sublayers_minus1に等しいTemporalIdを有するサブレイヤ表現のHDRパラメータを含むことを指定する。方法700の幾つかの実施形態では、第2シンタックス要素の第2値は、シンタックス構造vui_payload()が、シーケンスパラメータセットの未処理バイトシーケンスペイロード(RBSP)内に存在するかどうかを指定する。方法700の幾つかの実施形態では、第2シンタックス要素は、シーケンスパラメータセット、ピクチャパラメータセット、ピクチャヘッダ、又はスライスヘッダに含まれ、第2シンタックス要素は、選択的に名称変更される。

10

【0158】

図8は、ビデオ処理の例示的な方法800のフローチャートである。操作802は、ルールに従って、ビデオブロックを含むビデオとビデオのビットストリームとの間の変換を実行するステップを含み、ビデオブロックは、1つ以上のコーディングユニットを含むコーディングツリーノードであり、ルールは、ビデオブロックのコーディング情報が、ビデオブロックの1つ以上のコーディングユニットに対してコーディングモードが有効になっているかどうかを示すことを指定する。

20

【0159】

方法800の幾つかの実施形態では、ルールは、コーディング情報が、コーディングツリーノードのコーディングモードタイプを指定する変数を含むことを指定する。方法800の幾つかの実施形態では、ルールは、コーディング情報が、コーディングツリーノードに含まれる1つ以上のコーディングユニットがインター予測コーディングモードの使用を許可されるかどうかを示すシンタックス要素を含むことを指定する。方法800の幾つかの実施形態では、ルールは、コーディングツリーノードが特定のコーディングモードタイプに属することを指定する変数にตอบสนองして、コーディングモードが、コーディングツリーノードの1つ以上のコーディングユニットに対して有効なパレットコーディングモードを含むことを指定する。方法800の幾つかの実施形態では、変数は、コーディングツリーノードの特定のコーディングモードタイプがMODE_TYPE_INTRAに等しいことを示す。方法800の幾つかの実施形態では、変数は、コーディングツリーノードの特定のコーディングモードタイプがMODE_TYPE_ALLに等しいことを示す。

30

【0160】

方法800の幾つかの実施形態では、変数は、ビットストリームが、パレットコーディングモードの使用が有効であることを示す第2シンタックス要素を含む。方法800の幾つかの実施形態では、特定のコーディングモードタイプが、コーディングツリーノードに含まれる1つ以上のコーディングユニットに対して、イントラ予測コーディングモード、イントラブロックコピー(IBC)コーディングモード、及びパレットコーディングモードのみが許可されることを示す第1モードタイプを含む。方法800の幾つかの実施形態では、特定のコーディングモードタイプが、コーディングツリーノードに含まれる1つ以上のコーディングユニットに対して、インター予測コーディングモードのみが許可されることを示す第2モードタイプを含む。方法800の幾つかの実施形態では、特定のコーディングモードタイプが、コーディングツリーノードに含まれる1つ以上のコーディングユニットに対して、イントラ予測コーディングモード、イントラブロックコピー(IBC)コーディングモード、パレットコーディングモード、及びインター予測コーディングモードが許可されることを示す第2モードタイプを含む。

40

【0161】

50

方法 800 の幾つかの実施形態では、ルールは、コーディングツリーノードが特定のコーディングモードタイプに属することを指定する変数に回答して、コーディングモードが、コーディングツリーノードのルマブロックのみに対して有効なパレットコーディングモードを含むことを指定する。方法 800 の幾つかの実施形態では、変数は、コーディングツリーノードの特定のコーディングモードタイプが MODE_TYPE_INTRA に等しいことを示す。方法 800 の幾つかの実施形態では、変数は、コーディングツリーノードの特定のコーディングモードタイプが MODE_TYPE_ALL に等しいことを示す。方法 800 の幾つかの実施形態では、変数は、ビットストリームが、パレットコーディングモードの使用が有効であることを示す第 2 シンタックス要素を含む。方法 800 の幾つかの実施形態では、特定のコーディングモードタイプが、コーディングツリーノードに含まれる 1 つ以上のコーディングユニットに対して、イントラ予測コーディングモード、イントラブロックコピー（IBC）コーディングモード、及びパレットコーディングモードのみが許可されることを示す第 1 モードタイプを含む。

10

【0162】

方法 800 の幾つかの実施形態では、特定のコーディングモードタイプが、コーディングツリーノードに含まれる 1 つ以上のコーディングユニットに対して、イントラ予測コーディングモード、イントラブロックコピー（IBC）コーディングモード、パレットコーディングモード、及びインター予測コーディングモードが許可されることを示す第 3 モードタイプを含む。方法 800 の幾つかの実施形態では、ルールが、コーディングツリーノードの 1 つ以上のコーディングユニットに対して有効なコーディングモードが、コーディングツリーノードにコーディングモード制限が適用されないことを指定する変数に回答して、以下：

20

- イントラ予測コーディングモード、
- イントラブロックコピー（IBC）コーディングモード、
- パレットコーディングモード、及び
- インター予測コーディングモード、

のいずれか 1 つ以上を含むことを指定する。方法 800 の幾つかの実施形態では、変数は、コーディングツリーノードのコーディングモードタイプが MODE_TYPE_ALL に等しいことを示す。方法 800 の幾つかの実施形態では、ルールが、コーディングツリーノードの 1 つ以上のコーディングユニットに対して有効なコーディングモードが、コーディングツリーノードにローカルデュアルツリーが適用されることに回答して、以下：

30

- イントラ予測コーディングモード、
- イントラブロックコピー（IBC）モード、及び
- パレットコーディングモード、

のいずれか 1 つ以上を含むことを指定する。方法 800 の幾つかの実施形態では、コーディングツリーノードのコーディングモードタイプが MODE_TYPE_INTRA に等しい。

【0163】

方法 800 の幾つかの実施形態では、変数は、コーディングツリーノードが、コーディングツリーノードに含まれる 1 つ以上のコーディングユニットに対してイントラ予測コーディングモード、IBCモード、及びパレットコーディングモードのみが許可されることを示す特定のコーディングモードタイプに属することを指定する。方法 800 の幾つかの実施形態では、ルールは、コーディングツリーノードのルマブロックに対して有効なコーディングモードが、ローカルデュアルツリーがコーディングツリーノードの 1 つ以上のコーディングユニットに適用されていることに回答して、イントラ予測コーディングモード、イントラブロックコピー（IBC）コーディングモード、及びパレットコーディングモードのいずれか 1 つ以上を含むことを指定する。方法 800 の幾つかの実施形態では、コーディングツリーノードのコーディングモードタイプが MODE_TYPE_INTRA に等しい。方法 800 の幾つかの実施形態では、ルールは、イントラ予測コーディングモードがコーディングツリーノードのクロマブロックに適用されることを指定する。方法 800 の幾つかの実施形態では、ルールは、イントラ予測コーディングモードが、コーディングツリーノード

40

50

ドの全部のルマブロックに適用されることを指定する。方法 800 の幾つかの実施形態では、ルールは、IBC コーディングモードがコーディングツリーノードの全部のルマブロックに適用されることを指定する。

【 0 1 6 4 】

方法 800 の幾つかの実施形態では、ルールは、パレットコーディングモードがコーディングツリーノードのルマブロックに適用されることを指定する。方法 800 の幾つかの実施形態では、ルールは、1 つより多くのコーディングモードがコーディングツリーノードのルマブロックに適用されることが許可されることを指定し、1 つより多くのコーディングモードが、イントラ予測コーディングモード、イントラブロックコピー (IBC) コーディングモード、及びパレットコーディングモードのうちのいずれか 2 つ以上を含む。方法 800 の幾つかの実施形態では、ルールは、ローカルデュアルツリーが適用されるコーディングツリーノード内の 1 つ以上のコーディングユニットに対して、インター予測コーディングモードが許可されないことを指定する。方法 800 の幾つかの実施形態では、ルールは、コーディングツリーノード内のコーディングユニット (CU) が特定のコーディングモードタイプを有することに応答して、パレットコーディングモードが、コーディングツリーノードの 1 つ以上のコーディングユニットに対して有効かどうか、ビットストリーム内で示されることを指定する。方法 800 の幾つかの実施形態では、特定のコーディングモードタイプが、コーディングツリーノードに含まれる 1 つ以上のコーディングユニットに対して、イントラ予測コーディングモード、イントラブロックコピー (IBC) コーディングモード、及びパレットコーディングモードのみが許可されることを示す第 1 モードタイプを含む。方法 800 の幾つかの実施形態では、特定のコーディングモードタイプが、コーディングツリーノードに含まれる 1 つ以上のコーディングユニットに対して、イントラ予測コーディングモード、イントラブロックコピー (IBC) コーディングモード、パレットコーディングモード、及びインター予測コーディングモードが許可されることを示す第 3 モードタイプを含む。

【 0 1 6 5 】

図 9 は、ビデオ処理の例示的な方法 900 のフローチャートである。操作 902 は、ルールに従って、ビデオブロックを含むビデオピクチャを含むビデオとビデオのビットストリームとの間の変換を実行するステップを含み、ルールは、ビデオブロックに対してパレットコーディングモードが有効であることを示すコーディングモードタイプを使用したパーティション処理を使用して、ビデオブロックを選択的にコーディングブロックにパーティションすることを指定実行するステップを含む。

【 0 1 6 6 】

方法 900 の幾つかの実施形態では、コーディングモードタイプは、コーディングツリーノードに含まれるコーディングユニットに対して、イントラ予測コーディングモード、イントラブロックコピー (IBC) コーディングモード、パレットコーディングモード、及びインター予測コーディングモードが許可されることを示す。方法 900 の幾つかの実施形態では、コーディングモードタイプは、コーディングツリーノードに含まれるコーディングユニットに対して、イントラ予測コーディングモード、イントラブロックコピー (IBC) コーディングモード、及びパレットコーディングモードのみが許可されることを示す。方法 900 の幾つかの実施形態では、コーディングモードタイプは、コーディングツリーノードに含まれるコーディングユニットに対して、インター予測コーディングモードのみが許可されることを示す。方法 900 の幾つかの実施形態では、パーティション処理は 4 分割処理を含む。方法 900 の幾つかの実施形態では、パーティション処理は 2 分割処理を含む。方法 900 の幾つかの実施形態では、パーティション処理は 3 分割処理を含む。

【 0 1 6 7 】

方法 800 ~ 900 の幾つかの実施形態では、MODE_TYPE_INTRA のコーディングモードタイプ又は特定のコーディングモードタイプは、イントラ予測コーディングモード、イントラブロックコピー (IBC) コーディングモード、及びパレットコーディングモードを含む。方法 800 ~ 900 の幾つかの実施形態では、MODE_TYPE_ALL のコーディン

グモードタイプ又は特定のコーディングモードタイプは、イントラ予測コーディングモード、イントラブロックコピー（IBC）コーディングモード、パレットコーディングモード、及びインター予測コーディングモードを含む。

【0168】

方法700～900の幾つかの実施形態では、変換は、ビデオをビットストリームに符号化することを含む。方法700～900の幾つかの実施形態では、変換を実行するステップは、ビデオをビットストリームに符号化するステップを含み、方法は、ビットストリームを非一時的コンピュータ可読記録媒体に格納するステップを更に含む。方法800～900の幾つかの実施形態では、変換を実行するステップは、ビットストリームからビデオを復号することを含む。幾つかの実施形態では、ビデオ復号機器は、方法700～900に関連する実施形態で引用された方法を実施するよう構成されるプロセッサを含む。幾つかの実施形態では、ビデオ符号化機器は、方法700～900に関連する実施形態で引用された方法を実施するよう構成されるプロセッサを含む。幾つかの実施形態では、格納されたコンピュータ命令を有するコンピュータプログラムプロダクトであって、命令は、プロセッサにより実行されると、プロセッサに方法700～900に関連する実施形態について記載された方法を実施させる、コンピュータプログラムプロダクト。幾つかの実施形態では、非一時的コンピュータ可読記憶媒体は、方法700～900に関連する実施形態について記載された方法に従い生成されたビットストリームを格納する。幾つかの実施形態では、命令を格納している非一時的コンピュータ可読記憶媒体であって、命令は、プロセッサに方法700～900に関連する実施形態について記載された方法を実施させる、非一時的コンピュータ可読記憶媒体。幾つかの実施形態では、ビットストリームを生成する方法は、

方法700～900に関連する実施形態について記載された方法に従い、ビデオのビットストリームを生成するステップと、

ビットストリームをコンピュータ可読プログラム媒体に格納するステップと、

を含む。幾つかの実施形態では、本願明細書に記載された開示された方法又はシステムに従う方法、機器、生成されるビットストリーム。

【0169】

本願明細書では、用語「ビデオ処理」は、ビデオ符号化、ビデオ復号、ビデオ圧縮、又はビデオ伸長を表してよい。例えば、ビデオ圧縮アルゴリズムは、ビデオをピクセル表現から対応するビットストリーム表現へ又はその逆に変換する間に適用されてよい。現在ビデオブロックのビットストリーム表現は、シンタックスにより定義されるように、例えば、ビットストリーム内の同一位置にある又は異なる場所に拡散しているビットに対応してよい。例えば、マクロブロックは、変換されたコーディングされた誤差残差値の観点で、ヘッダ及びビットストリーム内の他のフィールドのビットも使用して、符号化されてよい。更に、変換中、デコーダは、上述のソリューションで説明されたように、決定に基づき、幾つかのフィールドが存在する又は不存在であり得るという知識により、ビットストリームをパースしてよい。同様に、エンコーダは、特定のシンタックスフィールドが含まれる又は含まれないことを決定し、コーディング表現にシンタックスフィールドを含める又は除外することにより、相応してコーディング表現を生成してよい。

【0170】

本願明細書に記載された本開示の及び他のソリューション、例、実施形態、モジュール、及び機能動作は、デジタル電子回路で、又は本願明細書に開示された構造を含む、コンピュータソフトウェア、ファームウェア、又はハードウェア、及びそれらの構造的均等物で、又はそれらの1つ以上の結合で、実装できる。本開示の及び他の実施形態は、1つ以上のコンピュータプログラムプロダクト、つまり、データ処理機器による実行のために又はその動作を制御するために、コンピュータ可読媒体上に符号化されたコンピュータプログラム命令の1つ以上のモジュールとして実装できる。コンピュータ可読媒体は、機械可読記憶装置、機械可読記憶基板、メモリ装置、機械可読伝搬信号に影響を与える物質の組成、又は1つ以上のそれらの組合せであり得る。用語「データ処理機器」は、データを処

理するあらゆる機器、装置、及び機械を包含し、例として、プログラマブルプロセッサ、コンピュータ、又は複数のプロセッサ若しくはコンピュータを含む。機器は、ハードウェアに加えて、対象となるコンピュータプログラムの実行環境を生成するコード、例えばプロセッサファームウェア、プロトコルスタック、データベース管理システム、オペレーティングシステム、又はそれらの1つ以上の組合せを構成するコードを含むことができる。伝搬信号は、人工的に生成された信号、例えば、適切な受信機機器への送信のために情報を符号化するために生成された、機械により生成された電気、光、又は電磁気信号である。

【0171】

コンピュータプログラム（プログラム、ソフトウェア、ソフトウェアアプリケーション、スクリプト、又はコードとしても知られる）は、コンパイルされた又はインタープリットされた言語を含む任意の形式のプログラミング言語で記述でき、それは、スタンドアロンプログラム又はモジュール、コンポーネント、サブルーチン、又はコンピューティング環境内での使用に適する他のユニットを含む任意の形式で展開できる。コンピュータプログラムは、必ずしもファイルシステム内のファイルに対応する必要はない。プログラムは、他のプログラム又はデータ（例えばマークアップ言語文書内に格納された1つ以上のスクリプト）を保持するファイルの一部に、問題のプログラムに専用の単一のファイルに、又は複数の連携ファイル（例えば、1つ以上のモジュール、サブプログラム、又はコードの部分を格納するファイル）に、格納できる。コンピュータプログラムは、1つのコンピュータ上で、又は1つの場所に置かれた若しくは複数の場所に分散されて通信ネットワークにより相互接続される複数のコンピュータ上で、実行されるよう展開できる。

【0172】

本願明細書に記載の処理及びロジックフローは、入力データに作用し及び出力を生成することにより機能を実行する1つ以上のコンピュータプログラムを実行する1つ以上のプログラマブルプロセッサにより実行できる。特定用途論理回路、例えば、FPGA（field programmable gate array）又はASIC（application specific integrated circuit）により、処理及びロジックフローが実行でき、それとして機器が実装できる。

【0173】

コンピュータプログラムの実行に適するプロセッサは、例えば、汎用及び特定用途向けマイクロプロセッサの両方、及び任意の種類デジタルコンピュータの任意の1つ以上のプロセッサを含む。通常、プロセッサは、命令及びデータを読み出し専用メモリ又はランダムアクセスメモリ又は両者から受信する。コンピュータの基本的要素は、命令を実行するプロセッサ、及び命令及びデータを格納する1つ以上のメモリ装置である。通常、コンピュータは、データを格納する1つ以上の大容量記憶装置、例えば、磁気、光磁気ディスク、又は光ディスク、も含み、又はそれらからデータを受信し又はそれらへデータを転送するために又は両者のために動作可能に結合される。しかしながら、コンピュータはこのような装置を有する必要はない。コンピュータプログラム命令及びデータを格納するのに適するコンピュータ可読媒体は、例えば半導体メモリ装置、例えばEPROM、EEPROM、及びフラッシュメモリ装置、磁気ディスク、例えば内部ハードディスク又は取り外し可能ディスク、光磁気ディスク、及びCD-ROM及びDVD-ROMディスクを含む、全ての形式の不揮発性メモリ、媒体、及びメモリ装置を含む。プロセッサ及びメモリは、特定用途向け論理回路により補足され、又はその中に組み込むことができる。

【0174】

本願明細書は多数の特定事項を含むが、これらは、任意の主題の又は請求され得るものの範囲に対する限定としてではなく、むしろ、特定の技術の特定の実施形態に固有の特徴の説明として考えられるべきである。別個の実装の文脈で本願明細書に記載された特定の特徴は、単一の実施形態において組み合わせることもできる。反対に、単一の実施形態の文脈で記載された種々の特徴は、複数の実施形態の中で別個に又は任意の適切な部分的組み合わせで実装されることもできる。更に、特徴は特定の組み合わせで動作するよう上述され、そのように初めに請求され得るが、請求される組み合わせからの1つ以上の特徴は、幾つかの場合には、組み合わせから切り離されてよく、請求される組み合わせは、部分

10

20

30

40

50

的組み合わせ又は部分的組み合わせの変形に向けられてよい。

【 0 1 7 5 】

同様に、動作は、図中に特定の順序で示されるが、これは、望ましい結果を達成するために、そのような動作が示された特定の順序で又はシーケンシャルに実行されること、及び全ての図示の動作が実行されること、を要求すると理解されるべきではない。更に、本願明細書に記載された実施形態における種々のシステムコンポーネントの分離は、全ての実施形態においてこのような分離を必要とすると理解されるべきではない。

【 0 1 7 6 】

少数の実装及び例のみが記載され、本願明細書に記載され示されたものに基づき他の実装、拡張、及び変形が行われ得る。

10

20

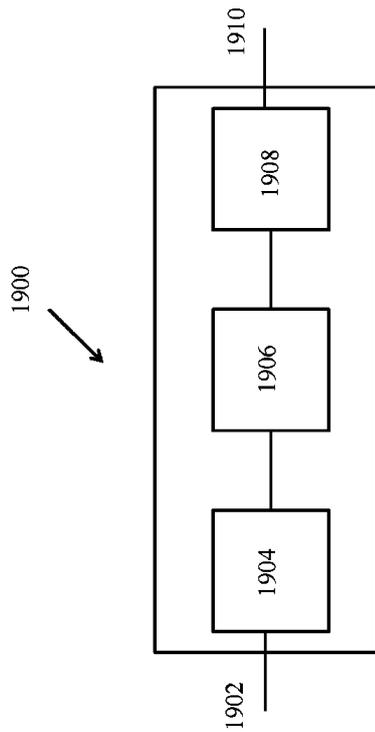
30

40

50

【図面】

【図 1】



【図 2】

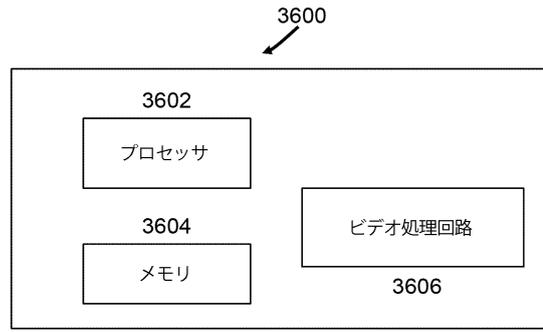
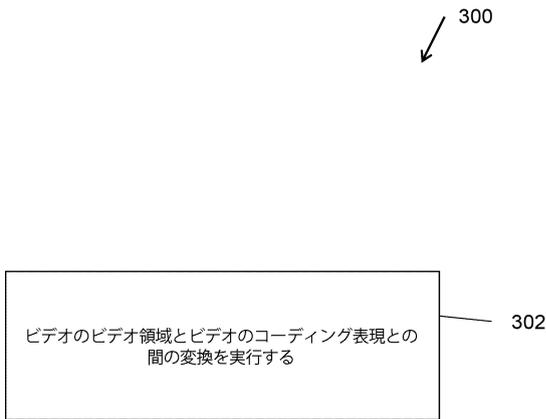
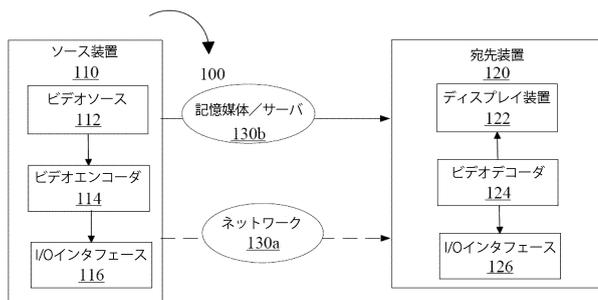


FIG. 1

【図 3】



【図 4】



10

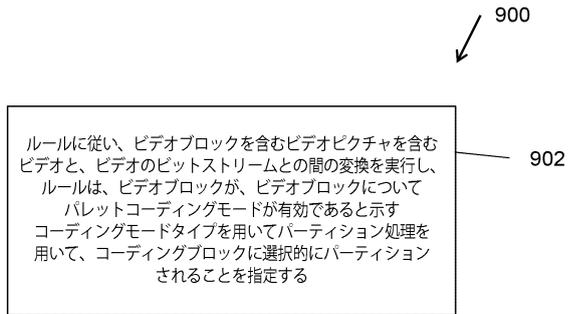
20

30

40

50

【図 9】



10

20

30

40

50

フロントページの続き

BYTEDANCE INC .

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロアー, スイート ナンバー・137

12655 West Jefferson Boulevard, Sixth Floor, Suite No. 137 Los Angeles, California 90066 United States of America

(74)代理人 100107766

弁理士 伊東 忠重

(74)代理人 100070150

弁理士 伊東 忠彦

(74)代理人 100135079

弁理士 宮崎 修

(72)発明者 デン, ジピン

中国 100080 ベイジン, ハイディエン ディストリクト, ズィチュン ロード, ナンバー・63, チャイナ サテライト コミュニケーションズ タワー, ジンリトウシャオ ポスト オフィス

(72)発明者 ワン, イェ - クイ

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロアー, スイート ナンバー・137

(72)発明者 ザン, リー

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロアー, スイート ナンバー・137

(72)発明者 ザン, カイ

アメリカ合衆国 カリフォルニア州 90066, ロサンジェルス, ウェスト ジェファーソン ブールヴァード 12655, シックス フロアー, スイート ナンバー・137

審査官 田中 純一

(56)参考文献 特表2022-525430 (JP, A)

米国特許出願公開第2021/0227241 (US, A1)

特表2022-549263 (JP, A)

米国特許出願公開第2022/0210449 (US, A1)

Zhipin Deng, et al., Non-CE3: Cleanups on local dual tree for non-4:2:0 chroma formats [online], JVET-P JVET-P0537-v2, ITU-T インターネット <URL:https://jvet-experts.org/doc_end_user/documents/16_Geneva/wg11/JVET-P0537-v3.zip>, 2019年10月09日, pp.1-4

Benjamin Bross, et al., Versatile Video Coding (Draft 6) [online], JVET-O JVET-O2001-vE, ITU-T インターネット <URL:https://jvet-experts.org/doc_end_user/documents/15_Gothenburg/wg11/JVET-O2001-v1.zip>, 2019年07月31日, pp.i-iv, pp.1-439

Karam NASER, et al., AHG9: Additional General Constraint Flags [online], JVET-S JVET-S0058, ITU-T インターネット <URL:https://jvet-experts.org/doc_end_user/documents/19_Teleconference/wg11/JVET-S0058-v1.zip>, 2020年05月22日, pp.1-4

(58)調査した分野 (Int.Cl., DB名)

H04N 7/12

H04N 19/00 - 19/98