



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

H.264

(03/2005)

СЕРИЯ Н: АУДИОВИЗУАЛЬНЫЕ И
МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

Инфраструктура аудиовизуальных служб –
Кодирование движущихся видеоизображений

**Усовершенствованное кодирование
изображений для общих аудиовизуальных
услуг**

Рекомендация МСЭ-Т H.264

РЕКОМЕНДАЦИИ МСЭ-Т СЕРИИ Н
АУДИОВИЗУАЛЬНЫЕ И МУЛЬТИМЕДИЙНЫЕ СИСТЕМЫ

ХАРАКТЕРИСТИКИ ВИДЕОТЕЛЕФОННЫХ СИСТЕМ	Н.100–Н.199
ИНФРАСТРУКТУРА АУДИОВИЗУАЛЬНЫХ СЛУЖБ	
Общие положения	Н.200–Н.219
Мультиплексирование и синхронизация при передаче	Н.220–Н.229
Системные аспекты	Н.230–Н.239
Процедуры связи	Н.240–Н.259
Кодирование движущихся видеонизображений	Н.260–Н.279
Сопутствующие системные аспекты	Н.280–Н.299
Системы и оконечное оборудование для аудиовизуальных служб	Н.300–Н.349
Архитектура служб каталогов для аудиовизуальных и мультимедийных служб	Н.350–Н.359
Качество архитектуры обслуживания для аудиовизуальных и мультимедийных служб	Н.360–Н.369
Дополнительные услуги для мультимедийных служб	Н.450–Н.499
ПРОЦЕДУРЫ МОБИЛЬНОСТИ И СОВМЕСТНОЙ РАБОТЫ	
Обзор мобильности и совместной работы, определений, протоколов и процедур	Н.500–Н.509
Мобильность для мультимедийных систем и служб серии Н	Н.510–Н.519
Приложения и службы мобильной мультимедийной совместной работы	Н.520–Н.529
Безопасность для мобильных мультимедийных систем и служб	Н.530–Н.539
Безопасность для приложений и служб мобильной мультимедийной совместной работы	Н.540–Н.549
Процедуры мобильного взаимодействия	Н.550–Н.559
Процедуры взаимодействия мобильной мультимедийной совместной работы	Н.560–Н.569
ШИРОКОПОЛОСНЫЕ И МУЛЬТИМЕДИЙНЫЕ УСЛУГИ TRIPLE-PLAY	
Предоставление широкополосных мультимедийных услуг по VDSL	Н.610–Н.619

Для получения более подробной информации просьба обращаться к перечню Рекомендаций МСЭ-Т.

Усовершенствованное кодирование изображений для общих аудиовизуальных услуг

Резюме

Настоящая Рекомендация | Международный стандарт отражает эволюцию существующих стандартов кодирования видеосигналов (Н.261, Н.262 и Н.263) и была разработана в ответ на возросшую потребность в более высокой сжатии движущихся изображений для разных приложений, таких как видеоконференция, цифровое хранение аудиовизуальных данных, телевизионное радиовещание, потоки трафика интернета и связь. Целью разработки данной Рекомендации является также обеспечение гибкого представления кодированного видеосигнала для разнообразных условий сетевого окружения. Использование настоящей Рекомендации | Международного стандарта позволит управлять движущимся изображением как видом компьютерных данных, хранить изображение на различных запоминающих устройствах, передавать и принимать по существующим и будущим сетям, а также распространять по существующим и будущим радиовещательным каналам.

Утвержденный в марте 2005 года пересмотр содержит изменения стандарта кодирования видеосигналов, заключающиеся в добавлении четырех новых профилей, называемых Высокий, Высокий 10, Высокий 4:2:2 и Высокий 4:4:4, обеспечении более высокого качества видеосигнала и расширении диапазона приложений, охватываемых настоящим стандартом (например, путем включения поддержки значительно более широкого диапазона значений точности образцов изображения и форматов цветности с более высоким разрешением). Кроме того, для дальнейшего расширения применимости данного стандарта кодирования видеосигналов введены определения новых типов дополнительных данных. Наконец, был исправлен ряд ошибок, допущенных в опубликованном тексте. Кроме расширения возможности кодирования видеосигналов настоящий пересмотр обеспечивает техническое соответствие с соответствующим совместно разработанным стандартом ИСО/МЭК 14496-10.

В Исправление 1 к Рекомендации МСЭ-Т Н.264 внесен ряд мелких исправлений и обновлений, с тем чтобы привести версию МСЭ-Т в соответствие с утвержденным как новое издание по состоянию на апрель 2005 года совместно разработанным и согласованным в техническом аспекте текстом ИСО/МЭК 14496-10. В нем также исправлены некоторые незначительные ошибки и неясности, а также определены ранее зарезервированные три коэффициента пропорциональности образца.

Настоящее издание включает текст, утвержденный в апреле 2005 года, и Исправление 1 к нему, утвержденное в сентябре 2005 года.

Источник

Рекомендация МСЭ-Т Н.264 утверждена 1 марта 2005 года 16-й Исследовательской комиссией МСЭ-Т (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации А.8 МСЭ-Т. В это издание включены изменения, введенные в Исправлении 1 к Рекомендации МСЭ-Т Н.264.1 (2005 г.), утвержденном 13 сентября 2005 г. 16-й Исследовательской комиссией (2005–2008 гг.) в соответствии с процедурой, изложенной в Рекомендации МСЭ-Т А.8.

ПРЕДИСЛОВИЕ

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи МСЭ (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе.

На Всемирной ассамблее по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяются темы для изучения Исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам.

Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ.

В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

ПРИМЕЧАНИЕ

В настоящей Рекомендации термин "администрация" используется для краткости и обозначает как администрацию электросвязи, так и признанную эксплуатационную организацию.

Соблюдение положений данной Рекомендации носит добровольный характер. Однако в Рекомендации могут содержаться определенные обязательные положения (например, для обеспечения возможности взаимодействия или применимости), и соблюдение положений данной Рекомендации достигается в случае выполнения всех этих обязательных положений. Для выражения необходимости выполнения требований используется синтаксис долженствования и соответствующие слова (такие, как "должен" и т. п.), а также их отрицательные эквиваленты. Использование этих слов не предполагает, что соблюдение положений данной Рекомендации является обязательным для какой-либо из сторон.

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на вероятность того, что практическое применение или реализация этой Рекомендации может включать использование заявленного права интеллектуальной собственности. МСЭ не занимает какую бы то ни было позицию относительно подтверждения, обоснованности или применимости заявленных прав интеллектуальной собственности, независимо от того, отстаиваются ли они членами МСЭ или другими сторонами вне процесса подготовки Рекомендации.

На момент утверждения настоящей Рекомендации МСЭ получил извещение об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для выполнения этой Рекомендации. Однако те, кто будет применять Рекомендацию, должны иметь в виду, что это может не отражать самую последнюю информацию, и поэтому им настоятельно рекомендуется обращаться к патентной базе данных БСЭ по адресу: <http://www.itu.int/ITU-T/ipr/>.

© ITU 2008

Все права сохранены. Никакая часть данной публикации не может быть воспроизведена с помощью каких-либо средств без предварительного письменного разрешения МСЭ.

СОДЕРЖАНИЕ

	Стр.
Предисловие.....	xiv
0 Введение	1
0.1 Пролог.....	1
0.2 Цель.....	1
0.3 Применения.....	1
0.4 Публикация и версии настоящей спецификации.....	1
0.5 Профили и уровни.....	2
0.6 Обзор проектируемых характеристик	2
0.6.1 Кодирование с предсказанием	3
0.6.2 Кодирование последовательных или чередующихся видеосигналов.....	3
0.6.3 Разделение изображения на макроблоки и меньшие части.....	3
0.6.4 Уменьшение пространственной избыточности.....	3
0.7 Как читать настоящую спецификацию.....	3
1 Сфера применения	4
2 Нормативные ссылки.....	4
3 Определения.....	4
4 Сокращения.....	12
5 Условные обозначения.....	12
5.1 Арифметические операторы.....	13
5.2 Логические операторы.....	13
5.3 Операторы отношения.....	13
5.4 Двоичные операторы.....	13
5.5 Операторы присвоения.....	14
5.6 Обозначение ранга.....	14
5.7 Математические функции.....	14
5.8 Переменные, элементы синтаксиса и таблицы.....	15
5.9 Описание логических операций.....	16
5.10 Процессы.....	17
6 Форматы исходных, кодированных, декодированных и выходных данных, процессы сканирования и взаимоотношения смежности.....	17
6.1 Форматы потоков битов.....	17
6.2 Форматы исходного, декодированного и выходного изображений.....	17
6.3 Пространственное разделение на части изображений и секций.....	22
6.4 Процессы инверсного сканирования и процессы образования смежных частей.....	23
6.4.1 Процесс инверсного сканирования макроблока.....	23
6.4.2 Процесс инверсного сканирования при разделении макроблока и субмакроблока.....	24
6.4.2.1 Процесс инверсного сканирования при разделении макроблока.....	25
6.4.2.2 Процесс инверсного сканирования при разделении субмакроблока.....	25
6.4.3 Процесс инверсного сканирования блока яркости 4x4.....	26
6.4.4 Процесс инверсного сканирования блока яркости 8x8.....	26
6.4.5 Процесс создания доступного адреса макроблока.....	26
6.4.6 Процесс создания адреса смежного макроблока и его доступность.....	27
6.4.7 Процесс создания адреса смежного макроблока и его доступность в кадрах MBAFF.....	27
6.4.8 Процессы создания смежных макроблоков, блоков и разделения на части.....	28
6.4.8.1 Процесс создания смежных макроблоков.....	29
6.4.8.2 Процесс создания смежных блоков яркости 8x8.....	29
6.4.8.3 Процесс создания смежных блоков яркости 4x4.....	30
6.4.8.4 Процесс создания смежных блоков цветности 4x4.....	30
6.4.8.5 Процесс создания смежных блоков при разделении.....	31
6.4.9 Процесс создания смежных местоположений.....	33
6.4.9.1 Спецификация для смежных местоположений в полях и кадрах (но не кадрах MBAFF).....	33
6.4.9.2 Спецификация для смежных местоположений в полях и кадрах MBAFF.....	34
7 Синтаксис и семантика.....	36
7.1 Метод описания синтаксиса в табличной форме.....	36
7.2 Описание синтаксиса функций, категорий и дескрипторов.....	37

	Стр.
7.3 Синтаксис в табличной форме	38
7.3.1 Синтаксис блока NAL	38
7.3.2 Синтаксис полезной нагрузки последовательностей исходных байтов и концевых битов RBSP	39
7.3.2.1 Синтаксис RBSP набора параметров последовательности	39
7.3.2.1.1 Синтаксис списка масштабирования	40
7.3.2.1.2 Синтаксис RBSP расширения набора параметров последовательности	40
7.3.2.2 Синтаксис RBSP набора параметров изображения	41
7.3.2.3 Синтаксис RBSP дополнительной расширенной информации	42
7.3.2.3.1 Синтаксис сообщения дополнительной расширенной информации	42
7.3.2.4 Синтаксис разграничителя блока доступа RBSP	43
7.3.2.5 Синтаксис RBSP конца последовательности	43
7.3.2.6 Синтаксис RBSP конца потока	43
7.3.2.7 Синтаксис RBSP данных заполнителя	43
7.3.2.8 Синтаксис RBSP слоя секции без разделения	43
7.3.2.9 Синтаксис RBSP разделения данных секции	43
7.3.2.9.1 Синтаксис RBSP части А данных секции	43
7.3.2.9.2 Синтаксис RBSP части В данных секции	44
7.3.2.9.3 Синтаксис RBSP части С данных секции	44
7.3.2.10 Синтаксис концевых битов секции RBSP	44
7.3.2.11 Синтаксис концевых битов RBSP	44
7.3.3 Синтаксис заголовка секции	45
7.3.3.1 Синтаксис изменения порядка списка контрольных изображений	46
7.3.3.2 Синтаксис таблицы веса предсказания	47
7.3.3.3 Синтаксис разметки декодированного контрольного изображения	48
7.3.4 Синтаксис данных секции	49
7.3.5 Синтаксис слоя макроблока	50
7.3.5.1 Синтаксис предсказания макроблока	51
7.3.5.2 Синтаксис предсказания субмакроблока	52
7.3.5.3 Синтаксис данных остатка	53
7.3.5.3.1 Синтаксис CAVLC блока остатка	54
7.3.5.3.2 Синтаксис CABAC блока остатка	55
7.4 Семантика	56
7.4.1 Семантика блока NAL	56
7.4.1.1 Инкапсуляция SODB в RBSP (для информации)	58
7.4.1.2 Порядок блоков NAL и связь с кодированными изображениями, блоками доступа и видеопоследовательностями	59
7.4.1.2.1 Порядок RBSP последовательности и набора параметров изображения и их активация	59
7.4.1.2.2 Порядок блоков доступа и связь с кодированными видеопоследовательностями	60
7.4.1.2.3 Порядок блоков NAL и кодированных изображений и их связь с блоками доступа	60
7.4.1.2.4 Обнаружение первого блока NAL VCL первично кодированного изображения	62
7.4.1.2.5 Порядок блоков NAL VCL и связь с кодированными изображениями	63
7.4.2 Полезная нагрузка последовательности исходных байтов и семантика концевых битов RBSP	63
7.4.2.1 Семантика RBSP набора параметров последовательности	63
7.4.2.1.1 Семантика списка масштабирования	68
7.4.2.1.2 Семантика RBSP расширения набора параметров последовательности	69
7.4.2.2 Семантика RBSP набора параметров изображения	70
7.4.2.3 Семантика дополнительного расширения информации RBSP	73
7.4.2.3.1 Семантика сообщения дополнительного расширения информации	73
7.4.2.4 Семантика разграничителя RBSP блока доступа	73
7.4.2.5 Семантика конца последовательности RBSP	73
7.4.2.6 Семантика конца потока битов RBSP	73
7.4.2.7 Семантика данных заполнителя RBSP	73
7.4.2.8 Семантика RBSP слоя без разделения	73
7.4.2.9 Семантика RBSP деления на части данных секции	74
7.4.2.9.1 Семантика RBSP деления на части данных А секции	74
7.4.2.9.2 Семантика RBSP деления на части данных В секции	74
7.4.2.9.3 Семантика RBSP деления на части данных С секции	74
7.4.2.10 Семантика секции концевых битов RBSP	74
7.4.2.11 Семантика RBSP концевых битов	75
7.4.3 Семантика заголовка секции	75
7.4.3.1 Семантика изменения порядка в списке контрольного изображения	80
7.4.3.2 Семантика таблицы веса предсказания	81
7.4.3.3 Семантика разметки декодированного контрольного изображения	82
7.4.4 Семантика данных секции	85

7.4.5	Семантика слоя макроблока.....	85
7.4.5.1	Семантика предсказания макроблока.....	92
7.4.5.2	Семантика предсказания субмакроблока.....	93
7.4.5.3	Семантика данных остатка.....	95
7.4.5.3.1	Семантика остатка блока CAVLC.....	96
7.4.5.3.2	Семантика остатка блока CABAC.....	96
8	Процесс декодирования.....	96
8.1	Процесс декодирования блока NAL.....	97
8.2	Процесс декодирования секции.....	98
8.2.1	Процесс декодирования для вычисления порядка изображений.....	98
8.2.1.1	Процесс декодирования для вычисления порядка изображений типа 0.....	99
8.2.1.2	Процесс декодирования для вычисления порядка изображений типа 1.....	100
8.2.1.3	Процесс декодирования для вычисления порядка изображений типа 2.....	101
8.2.2	Процесс декодирования для отображения макроблока в группу секции.....	102
8.2.2.1	Спецификация типа чередующейся группы секции.....	103
8.2.2.2	Спецификация типа отображения распределенной группы секции.....	103
8.2.2.3	Спецификация переднего плана с типом отображения сдвига группы секции налево вверх.....	104
8.2.2.4	Спецификация типов отображения групп секций вне блока.....	104
8.2.2.5	Спецификация типов отображения групп секций растрового сканирования.....	105
8.2.2.6	Спецификация типов отображения вытесненных групп секций.....	105
8.2.2.7	Спецификация типа отображения группы секции, заданной в явной форме.....	105
8.2.2.8	Спецификация для преобразования отображенного блока в отображенную группу секции и макроблока в отображенную группу секции.....	105
8.2.3	Процесс декодирования для разделения данных секции.....	105
8.2.4	Процесс декодирования для создания списков контрольных изображений.....	106
8.2.4.1	Процесс декодирования номеров изображений.....	107
8.2.4.2	Процесс инициализации списков контрольных изображений.....	107
8.2.4.2.1	Процесс инициализации списка контрольных изображений в кадрах секций P и SP.....	108
8.2.4.2.2	Процесс инициализации списка контрольных изображений в полях секций P и SP.....	108
8.2.4.2.3	Процесс инициализации списка контрольных изображений в кадрах секций B.....	109
8.2.4.2.4	Процесс инициализации списка контрольных изображений в полях секций B.....	109
8.2.4.2.5	Процесс инициализации списков контрольных изображений в полях.....	110
8.2.4.3	Процесс изменения порядка в списках контрольных изображений.....	110
8.2.4.3.1	Процесс изменения порядка в списках контрольных изображений для краткосрочных контрольных изображений.....	111
8.2.4.3.2	Процесс изменения порядка в списках контрольных изображений для долгосрочных контрольных изображений.....	112
8.2.5	Процесс разметки декодированного контрольного изображения.....	112
8.2.5.1	Последовательность операций для процесса разметки декодированного контрольного изображения.....	113
8.2.5.2	Процесс декодирования для промежутков в frame_num.....	113
8.2.5.3	Скользящее окно процесса разметки декодированного контрольного изображения.....	114
8.2.5.4	Процесс разметки декодированного контрольного изображения с адаптивным управлением памятью.....	114
8.2.5.4.1	Процесс разметки краткосрочного контрольного изображения как "не используется для контроля".....	114
8.2.5.4.2	Процесс разметки долгосрочного контрольного изображения как "не используется для контроля".....	115
8.2.5.4.3	Процесс присвоения значения LongTermFrameIdx краткосрочному контрольному изображению.....	115
8.2.5.4.4	Процесс декодирования MaxLongTermFrameIdx.....	115
8.2.5.4.5	Процесс разметки всех контрольных изображений как "не используется для контроля" и установка MaxLongTermFrameIdx на значение "нет индексов долгосрочных кадров".....	115
8.2.5.4.6	Процесс присвоения текущему изображению индекса долгосрочного кадра.....	116
8.3	Процесс внутреннего предсказания (Intra предсказания).....	116
8.3.1	Процесс предсказания образцов яркости Intra_4x4.....	117
8.3.1.1	Процесс вывода Intra4x4PredMode.....	117
8.3.1.2	Предсказание образца Intra_4x4.....	119
8.3.1.2.1	Описание режима предсказания Intra_4x4_Vertical.....	119
8.3.1.2.2	Описание режима предсказания Intra_4x4_Horizontal.....	119
8.3.1.2.3	Описание режима предсказания Intra_4x4_DC.....	120
8.3.1.2.4	Описание режима предсказания Intra_4x4_Diagonal_Down_Left.....	120
8.3.1.2.5	Описание режима предсказания Intra_4x4_Diagonal_Down_Right.....	120

	Стр.	
8.3.1.2.6	Описание режима предсказания Intra_4x4_Vertical_Right.....	121
8.3.1.2.7	Описание режима предсказания Intra_4x4_Horizontal_Down.....	121
8.3.1.2.8	Описание режима предсказания Intra_4x4_Vertical_Left.....	122
8.3.1.2.9	Описание режима предсказания Intra_4x4_Horizontal_Up.....	122
8.3.2	Процесс предсказания Intra_8x8 для образцов яркости.....	122
8.3.2.1	Процесс вывода Intra8x8PredMode.....	123
8.3.2.2	Предсказание образца Intra_8x8.....	124
8.3.2.2.1	Процесс фильтрации контрольных образцов для предсказания образцов Intra_8x8.....	125
8.3.2.2.2	Описание режима предсказания Intra_8x8_Vertical.....	126
8.3.2.2.3	Описание режима предсказания Intra_8x8_Horizontal.....	126
8.3.2.2.4	Описание режима предсказания Intra_8x8_DC.....	127
8.3.2.2.5	Описание режима предсказания Intra_8x8_Diagonal_Down_Left.....	127
8.3.2.2.6	Описание режима предсказания Intra_8x8_Diagonal_Down_Right.....	127
8.3.2.2.7	Описание режима предсказания Intra_8x8_Vertical_Right.....	128
8.3.2.2.8	Описание режима предсказания Intra_8x8_Horizontal_Down.....	128
8.3.2.2.9	Описание режима предсказания Intra_8x8_Vertical_Left.....	129
8.3.2.2.10	Описание режима предсказания Intra_8x8_Horizontal_Up.....	129
8.3.3	Процесс предсказания для образцов яркости Intra_16x16.....	129
8.3.3.1	Описание режима предсказания Intra_16x16_Vertical.....	130
8.3.3.2	Описание режима предсказания Intra_16x16_Horizontal.....	130
8.3.3.3	Описание режима предсказания Intra_16x16_DC.....	130
8.3.3.4	Описание режима предсказания Intra_16x16_Plane.....	131
8.3.4	Процесс Intra предсказания для образцов цветности.....	131
8.3.4.1	Описание режима предсказания Intra_Chroma_DC.....	132
8.3.4.2	Описание режима предсказания Intra_Chroma_Horizontal.....	134
8.3.4.3	Описание режима предсказания Intra_Chroma_Vertical.....	134
8.3.4.4	Описание режима предсказания Intra_Chroma_Plane.....	134
8.3.5	Процесс создания образцов для макроблоков I_PCM.....	134
8.4	<i>Процесс Inter предсказания.....</i>	<i>135</i>
8.4.1	Процесс вывода компонентов вектора движения и индексов контроля.....	137
8.4.1.1	Процесс вывода векторов движения яркости пропущенных макроблоков в секциях P и SP.....	138
8.4.1.2	Процесс вывода векторов движения яркости B_Skip, B_Direct_16x16 и B_Direct_8x8.....	139
8.4.1.2.1	Процесс вывода совпадающих разделенных частей субмакроблоков 4x4.....	139
8.4.1.2.2	Процесс вывода режима предсказания пространственного прямого вектора движения яркости и индекса контроля.....	142
8.4.1.2.3	Процесс вывода режимов предсказания временного прямого вектора движения яркости и индекса контроля.....	144
8.4.1.3	Процесс вывода предсказания вектора движения яркости.....	146
8.4.1.3.1	Процесс вывода предсказания медианы вектора движения яркости.....	147
8.4.1.3.2	Процесс вывода данных движения смежных частей.....	148
8.4.1.4	Процесс вывода векторов движения цветности.....	149
8.4.2	Процесс декодирования образцов Inter предсказания.....	149
8.4.2.1	Процесс выбора контрольного изображения.....	150
8.4.2.2	Процесс интерполяции фрагментарного образца.....	151
8.4.2.2.1	Процесс интерполяции образца яркости.....	152
8.4.2.2.2	Процесс интерполяции образца цветности.....	155
8.4.2.3	Процесс взвешенного предсказания образца.....	156
8.4.2.3.1	Процесс взвешенного по умолчанию предсказания образца.....	156
8.4.2.3.2	Процесс взвешенного предсказания образца.....	157
8.5	<i>Процесс декодирования коэффициента преобразования и процесс построения изображения до фильтра устранения блочности.....</i>	<i>159</i>
8.5.1	Описание процесса декодирования преобразования остаточных блоков яркости 4x4.....	160
8.5.2	Описание процесса декодирования преобразования для образцов яркости режима предсказания макроблока Intra_16x16.....	160
8.5.3	Описание процесса декодирования преобразования остаточных блоков яркости 8x8.....	161
8.5.4	Описание процесса декодирования преобразования образцов цветности.....	162
8.5.5	Процесс инверсного сканирования для коэффициентов преобразования.....	164
8.5.6	Процесс инверсного сканирования для коэффициентов преобразования яркости 8x8.....	164
8.5.7	Процесс вывода параметров квантования цветности и функция масштабирования.....	166
8.5.8	Процесс масштабирования и преобразования DC коэффициентов преобразования яркости макроблоков типа Intra_16x16.....	168
8.5.9	Процесс масштабирования и преобразования для DC коэффициентов преобразования цветности.....	169
8.5.10	Процесс масштабирования и преобразования остаточных блоков 4x4.....	171
8.5.11	Процесс масштабирования и преобразования остаточных блоков 8x8.....	173

8.5.12	Процесс построения изображения до процесса фильтра устранения блочности	176
8.5.13	Процесс остаточного преобразования цвета	177
8.6	<i>Процесс декодирования макроблоков P в секциях SP или макроблоков SI</i>	178
8.6.1	Процесс декодирования SP некомутируемых изображений	178
8.6.1.1	Процесс декодирования коэффициентов преобразования яркости	178
8.6.1.2	Процесс декодирования коэффициентов преобразования цветности	179
8.6.2	Процесс декодирования секций SP и SI для коммутации изображений	181
8.6.2.1	Процесс декодирования коэффициентов преобразования яркости	181
8.6.2.2	Процесс декодирования коэффициентов преобразования цветности	181
8.7	<i>Процесс фильтра устранения блочности</i>	182
8.7.1	Процесс фильтрации краев блоков	186
8.7.2	Процесс фильтрации набора образцов вдоль горизонтального и вертикального краев блока	188
8.7.2.1	Процесс вывода интенсивности фильтрации границы в зависимости от степени яркости	188
8.7.2.2	Процесс вывода порогов для края каждого блока	190
8.7.2.3	Процесс фильтрации краев при значении bS, меньшем 4	191
8.7.2.4	Процесс фильтрации краев при значении bS, равном 4	193
9	Процесс анализа	193
9.1	<i>Процесс анализа кодов Exp-Golomb</i>	194
9.1.1	Процесс отображения кодов Exp-Golomb со знаком	195
9.1.2	Процесс отображения кодированного блока образца	196
9.2	<i>Процесс анализа SA VLC для уровней коэффициентов преобразования</i>	198
9.2.1	Процесс анализа общего числа уровней коэффициентов преобразования и концевых уровней коэффициентов преобразования	199
9.2.2	Процесс анализа информационного уровня	201
9.2.2.1	Процесс анализа для level_prefix	202
9.2.3	Процесс анализа информации проходов	203
9.2.4	Объединение информации об уровне и проходе	206
9.3	<i>Процесс анализа CABAC данных секции</i>	206
9.3.1	Процесс инициализации	207
9.3.3.1	Процесс инициализации контекстных переменных	208
9.3.1.2	Процесс инициализации устройства арифметического декодирования	218
9.3.2	Процесс бинаризации	219
9.3.2.1	Унарный (U) процесс бинаризации	221
9.3.2.2	Усеченный унарный (TU) процесс бинаризации	221
9.3.2.3	Соединенный процесс бинаризации: унарный и k-ого порядка с экспоненциальным кодом Exp-Golomb (UEGk)	222
9.3.2.4	Процесс бинаризации фиксированной длины (FL)	222
9.3.2.5	Процесс бинаризации типов макроблока и субмакроблока	222
9.3.2.6	Процесс бинаризации для кодированного образца блока	225
9.3.2.7	Процесс бинаризации для mb_qp_delta	225
9.3.3	Процесс декодирования потока	225
9.3.3.1	Процесс вывода ctxIdx	226
9.3.3.1.1	Процесс присвоения ctxIdxInc с использованием ближайших элементов синтаксиса	228
9.3.3.1.1.1	Процесс вывода ctxIdxInc для элемента синтаксиса mb_skip_flag	228
9.3.3.1.1.2	Процесс вывода ctxIdxInc элемента синтаксиса mb_field_decoding_flag	228
9.3.3.1.1.3	Процесс вывода ctxIdxInc элемента синтаксиса mb_type	229
9.3.3.1.1.4	Процесс вывода ctxIdxInc элемента синтаксиса coded_block_pattern	229
9.3.3.1.1.5	Процесс вывода ctxIdxInc элемента синтаксиса mb_qp_delta	230
9.3.3.1.1.6	Процесс вывода ctxIdxInc элемента синтаксиса ref_idx_10 и ref_idx_11	230
9.3.3.1.1.7	Процесс вывода ctxIdxInc элементов синтаксиса mvd_10 и mvd_11	231
9.3.3.1.1.8	Процесс вывода ctxIdxInc элемента синтаксиса intra_chroma_pred_mode	232
9.3.3.1.1.9	Процесс вывода ctxIdxInc элемента синтаксиса coded_block_flag	233
9.3.3.1.1.10	Процесс вывода ctxIdxInc элемента синтаксиса transform_size_8x8_flag	234
9.3.3.1.2	Процесс присвоения значения ctxIdxInc, которое используют перед декодированием значений бинов	234
9.3.3.1.3	Процесс присвоения ctxIdxInc элементам синтаксиса significant_coeff_flag, last_significant_coeff_flag и coeff_abs_level_minus1	235
9.3.3.2	Процесс арифметического декодирования	237
9.3.3.2.1	Процесс арифметического декодирования для бинарного решения	238
9.3.3.2.1.1	Процесс перехода состояния	239
9.3.3.2.2	Процесс изменения нормировки в устройстве арифметического декодирования	241
9.3.3.2.3	Процесс обхода декодирования бинарных решений	242
9.3.3.2.4	Процесс декодирования бинарного решения перед завершением	242
9.3.4	Процесс арифметического кодирования (для информации)	243

9.3.4.1	Процесс инициализации устройства арифметического кодирования (для информации)	243
9.3.4.2	Процесс кодирования бинарного решения (для информации).....	243
9.3.4.3	Процесс изменения нормировки в устройстве арифметического кодирования (для информации).....	244
9.3.4.4	Процесс обхода кодирования бинарных решений (для информации)	246
9.3.4.5	Процесс кодирования бинарного решения перед завершением (для информации).....	246
9.3.4.6	Процесс стаффинга байтов (для информации)	247
Приложение А – Профили и уровни		249
A.1	Требования к возможностям видеodeкодера	249
A.2	Профили.....	249
A.2.1	Базовый профиль.....	249
A.2.2	Главный профиль	250
A.2.3	Расширенный профиль	250
A.2.4	Высокий профиль.....	250
A.2.5	Высокий 10 профиль.....	251
A.2.6	Высокий 4:2:2 профиль	251
A.2.7	Высокий 4:4:4 профиль	252
A.3	Уровни.....	252
A.3.1	Границы уровней, общие для базового, главного и расширенного профилей	252
A.3.2	Границы уровней, общие для высокого, высокого 10, высокого 4:2:2 и высокого 4:4:4 профилей.....	254
A.3.3	Границы уровней, определяемые профилем	255
A.3.3.1	Границы базового профиля.....	256
A.3.3.2	Границы главного, высокого, высокого 10, высокого 4:2:2 или высокого 4:4:4 профиля.....	257
A.3.3.3	Границы расширенного профиля.....	258
A.3.4	Влияние границ уровня на частоту кадров (для информации)	259
Приложение В – Формат потока байтов.....		262
V.1	Синтаксис и семантика блоков NAL потока байтов	262
V.1.1	Синтаксис блоков NAL потока байтов.....	262
V.1.2	Семантика блоков NAL потока байтов	262
V.2	Процесс декодирования блоков NAL потока байтов.....	262
V.3	Восстановление декодером выравнивания байтов (для информации)	263
Приложение С – Гипотетический контрольный декодер		264
C.1	Работа буфера кодированного изображения (CPB).....	266
C.1.1	Синхронизация поступления потока битов	266
C.1.2	Синхронизация удаления кодированного изображения	267
C.2	Работа буфера декодированного изображения (DPB).....	268
C.2.1	Декодирование промежутков в frame_num и хранение "несуществующих" кадров	268
C.2.2	Декодирование и выход изображения.....	268
C.2.3	Удаление изображения из DPB перед возможным введением текущего изображения.....	268
C.2.4	Разметка и хранение текущего декодированного изображения.....	269
C.2.4.1	Разметка и хранение контрольного декодированного изображения в DPB.....	269
C.2.4.2	Хранение неконтрольного изображения в DPB	269
C.3	Соответствие потока битов	269
C.4	Соответствие декодера	271
C.4.1	Работа DPB по организации очередности выхода	272
C.4.2	Декодирование промежутков в frame_num и хранение "несуществующих" изображений.....	272
C.4.3	Декодирование изображения	272
C.4.4	Удаление изображения из DPB перед возможным введением текущего изображения.....	272
C.4.5	Разметка и хранение текущего декодированного изображения.....	272
C.4.5.1	Разметка и хранение контрольного декодированного изображения в DPB.....	272
C.4.5.2	Разметка и хранение неконтрольного декодированного изображения в DPB.....	273
C.4.5.3	Процесс "пульсации"	273
Приложение D – Дополнительная расширенная информация		275
D.1	Синтаксис полезной нагрузки SEI.....	276
D.1.1	Синтаксис сообщения SEI о периоде буферизации	277
D.1.2	Синтаксис сообщения SEI о синхронизации изображения	277
D.1.3	Синтаксис сообщения SEI о прямоугольнике сканирования	278
D.1.4	Синтаксис сообщения SEI о полезной нагрузке заполнителя.....	278
D.1.5	Синтаксис сообщения SEI о данных пользователя, зарегистрированных по Рекомендации МСЭ-Т Т.35	279
D.1.6	Синтаксис сообщения SEI о незарегистрированных данных пользователя	279

D.1.7	Синтаксис сообщения SEI о пункте восстановления.....	279
D.1.8	Синтаксис сообщения SEI о повторной разметке декодированного контрольного изображения	279
D.1.9	Синтаксис сообщения SEI о резервном изображении	280
D.1.10	Синтаксис сообщения SEI о сценической информации	280
D.1.11	Синтаксис сообщения SEI об информации субпоследовательности	281
D.1.12	Синтаксис сообщения SEI о характеристиках слоя субпоследовательности	281
D.1.13	Синтаксис сообщения SEI о характеристиках субпоследовательности	281
D.1.14	Синтаксис сообщения SEI о полностью фиксированном кадре	282
D.1.15	Синтаксис сообщения SEI об отмене режима полностью фиксированного кадра.....	282
D.1.16	Синтаксис сообщения SEI о стоп-кадре.....	282
D.1.17	Синтаксис сообщения SEI о запуске сегмента последовательного улучшения	282
D.1.18	Синтаксис сообщения SEI об окончании сегмента последовательного улучшения	282
D.1.19	Синтаксис сообщения SEI о наборе группы секций ограниченного движения	282
D.1.20	Синтаксис сообщения SEI о характеристиках зернистости пленки	283
D.1.21	Семантика сообщения SEI о предпочтениях отображения с учетом фильтра устранения блочности.....	283
D.1.22	Синтаксис сообщения SEI об информации, касающейся стереоизображения	284
D.1.23	Синтаксис зарезервированных сообщений SEI.....	284
D.2	<i>Семантика полезной нагрузки SEI</i>	284
D.2.1	Семантика сообщения SEI о периоде буферизации.....	284
D.2.2	Семантика сообщения SEI о синхронизации изображения.....	285
D.2.3	Семантика сообщения SEI о прямоугольнике сканирования.....	288
D.2.4	Семантика сообщения SEI о полезной нагрузке заполнителя	290
D.2.5	Семантика сообщения SEI о данных пользователя, зарегистрированных по Рекомендации МСЭ-Т Т.35	290
D.2.6	Семантика сообщения SEI о незарегистрированных данных пользователя	290
D.2.7	Семантика сообщения SEI о пункте восстановления	290
D.2.8	Семантика сообщения SEI о повторной разметке декодированного контрольного изображения	291
D.2.9	Семантика сообщения SEI о резервном изображении.....	292
D.2.10	Семантика сообщения SEI о сценической информации	293
D.2.11	Семантика сообщения SEI об информации субпоследовательности	295
D.2.12	Семантика сообщения SEI о характеристиках слоя субпоследовательности	296
D.2.13	Семантика сообщения SEI о характеристиках субпоследовательности	297
D.2.14	Семантика сообщения SEI о полностью фиксированном кадре	299
D.2.15	Семантика сообщения SEI об отмене режима полностью фиксированного кадра	299
D.2.16	Семантика сообщения SEI о стоп-кадре	299
D.2.17	Семантика сообщения SEI о запуске сегмента последовательного улучшения.....	299
D.2.18	Семантика сообщения SEI об окончании сегмента последовательного улучшения.....	300
D.2.19	Семантика сообщения SEI о наборе группы секций ограниченного движения	300
D.2.20	Семантика сообщения SEI о характеристиках зернистости пленки.....	301
D.2.21	Семантика сообщения SEI о предпочтениях отображения с учетом фильтра устранения блочности.....	306
D.2.22	Семантика сообщения SEI об информации, касающейся стереоизображения.....	308
D.2.23	Семантика зарезервированного сообщения SEI.....	309
Приложение Е – Информация об используемости изображения (VUI).....		310
E.1	<i>Синтаксис VUI</i>	311
E.1.1	Синтаксис параметров VUI.....	311
E.1.2	Синтаксис параметров HRD.....	312
E.2	<i>Семантика VUI</i>	312
E.2.1	Семантика параметров VUI.....	312
E.2.2	Семантика параметров HRD	323

СПИСОК РИСУНКОВ

Рисунок 6-1 – Номинальное вертикальное и горизонтальное расположение образцов яркости и цветности 4:2:0 в кадре.....	19
Рисунок 6-2 – Номинальное вертикальное и горизонтальное расположение образцов 4:2:0 выборки в верхних и нижних полях	20
Рисунок 6-3 – Номинальное вертикальное и горизонтальное расположение образцов 4:2:2 яркости и цветности в кадре.....	20
Рисунок 6-4 – Номинальное вертикальное и горизонтальное расположение образцов 4:2:2 выборки в верхнем и нижнем полях	21

Рисунок 6-5 – Номинальное вертикальное и горизонтальное расположение образцов 4:4:4 яркости и цветности в кадре.....	21
Рисунок 6-6 – Номинальное вертикальное и горизонтальное расположение образцов 4:4:4 выборки в верхнем и нижнем полях.....	22
Рисунок 6-7 – Изображение с 11 на 9 макроблоков, которые разделены на две секции.....	23
Рисунок 6-8 – Разделение декодированного кадра на пару макроблоков.....	23
Рисунок 6-9 – Разделенные части макроблоков и субмакроблоков. Сканированные разделенные части макроблоков и субмакроблоков.....	25
Рисунок 6-10 – Сканирование блоков яркости 4x4.....	26
Рисунок 6-11 – Сканирование блоков яркости 8x8.....	26
Рисунок 6-12 – Смежные макроблоки по отношению к заданному макроблоку.....	27
Рисунок 6-13 – Смежные макроблоки по отношению к заданному макроблоку в кадрах MBAFF.....	28
Рисунок 6-14 – Определение смежных макроблоков, блоков и разделенных частей (для информации).....	29
Рисунок 7-1 – Структура блока доступа, не содержащего никаких блоков NAL со значениями <code>nal_unit_type</code> , равными 0, 7, 8 или лежащими в диапазоне от 12 до 18 включительно или в диапазоне от 20 до 31 включительно.....	62
Рисунок 8-1 – Направления режимов предсказания Intra_4x4 (для информации).....	118
Рисунок 8-2 – Пример временного прямого режима предполагаемого вектора движения (для информации).....	146
Рисунок 8-3 – Предсказания ориентированной сегментации (для информации).....	147
Рисунок 8-4 – Расположение целых образцов (затененные блоки с заглавными буквами) и фракций образцов (незатененные блоки со строчными буквами) для интерполяции четверти образца яркости.....	153
Рисунок 8-5 – Расположение фракций образца в зависимости от переменных в интерполяции цветности и окружающих целочисленных местоположений образцов A, B, C и D.....	155
Рисунок 8-6 – Присвоение индексов массива <code>dcY</code> значению <code>luma4x4BlkIdx</code>	161
Рисунок 8-7 – Присвоение индексов массива <code>dcC</code> параметру <code>chroma4x4BlkIdx</code> : а) <code>chroma_format_idc</code> равно 1, б) <code>chroma_format_idc</code> равно 2, в) <code>chroma_format_idc</code> равно 3.....	163
Рисунок 8-8 – Сканирование блоков 4x4: а) сканирование зигзагом; б) сканирование полем.....	164
Рисунок 8-9 – Сканирование блоков 8x8: а) сканирование зигзагом 8x8; б) сканирование полем 8x8 (для информации).....	165
Рисунок 8-10 – Границы в макроблоке, которые следует фильтровать.....	183
Рисунок 8-11 – Условные обозначения для описания образцов блоков 4x4 вдоль горизонтальной или вертикальной границы.....	187
Рисунок 9-1 – Иллюстрация процесса анализа CABAC элемента синтаксиса SE (для информации).....	207
Рисунок 9-2 – Диаграмма последовательности процесса арифметического декодирования для одиночного бина (для информации).....	238
Рисунок 9-3 – Диаграмма последовательности для декодирования решения.....	239
Рисунок 9-4 – Диаграмма последовательности изменения нормировки.....	241
Рисунок 9-5 – Диаграмма последовательности процесса обхода декодирования.....	242
Рисунок 9-6 – Диаграмма последовательности декодирования решения перед завершением.....	243
Рисунок 9-7 – Диаграмма последовательности кодирования решения.....	244
Рисунок 9-8 – Диаграмма последовательности изменения нормировки в кодере.....	245
Рисунок 9-9 – Диаграмма последовательности <code>PutBit(B)</code>	245
Рисунок 9-10 – Диаграмма последовательности обхода кодирования.....	246
Рисунок 9-11 – Диаграмма последовательности кодирования решения перед завершением.....	247
Рисунок 9-12 – Диаграмма последовательности сдвига при завершении.....	247
Рисунок C-1 – Структура потоков байтов и потоков блоков NAL для проверки соответствия с помощью HRD.....	264
Рисунок C-2 – Модель буфера HRD.....	265
Рисунок E-1 – Расположение образцов цветности для верхнего и нижнего полей как функции <code>chroma_sample_loc_type_top_field</code> и <code>chroma_sample_loc_type_bottom_field</code>	320

СПИСОК ТАБЛИЦ

	Стр.
Таблица 6-1 – Значения SubWidthC и SubHeightC, полученное из chroma_format_idc	18
Таблица 6-2 – Спецификация присвоений входов и выходов в пп. 6.4.8.1–6.4.8.5	29
Таблица 6-3 – Спецификация mbAddrN	33
Таблица 6-4 – Спецификация mbAddrN и yM	35
Таблица 7-1 – Коды типов блока NAL	57
Таблица 7-2 – Присвоение мнемонических имен индексам списка масштабирования и спецификация правила обращения	65
Таблица 7-3 – Спецификация списков масштабирования по умолчанию Default_4x4_Intra и Default_4x4_Inter....	66
Таблица 7-4 – Спецификация списков масштабирования по умолчанию Default_8x8_Intra и Default_8x8_Inter....	66
Таблица 7-5 – Значения primary_pic_type	73
Таблица 7-6 – Наименование связи с slice_type	75
Таблица 7-7 – Списки контрольных изображений изменения порядка операций reordering_of_pic_nums_idc	81
Таблица 7-8 – Интерпретация значения adaptive_ref_pic_marking_mode_flag	83
Таблица 7-9 – Значения операций управления памятью (memory_management_control_operation)	84
Таблица 7-10 – Разрешенные типы макроблоков для значения slice_type	86
Таблица 7-11 – Типы макроблоков для секций I	87
Таблица 7-12 – Нулевые значения типа макроблока для секций SI	88
Таблица 7-13 – Значения типа макроблока от 0 до 4 для секций P и SP	89
Таблица 7-14 – Значения от 0 до 22 типов макроблоков для секций B	90
Таблица 7-15 – Спецификация значений CodedBlockPatternChroma	92
Таблица 7-16 – Взаимосвязь между режимами пространственных предсказаний intra_chroma_pred_mode	92
Таблица 7-17 – Подтип макроблоков для макроблоков типа P	93
Таблица 7-18 – Подтипы макроблоков для макроблоков типа B	94
Таблица 8-1 – Детальный тип отображения группы секции	102
Таблица 8-2 – Спецификация Intra4x4PredMode[luma4x4BlkIdx] и связанных имен	117
Таблица 8-3 – Спецификация Intra8x8PredMode[luma8x8BlkIdx] и связанных имен	123
Таблица 8-4 – Спецификация Intra16x16PredMode и связанные с этим имена	130
Таблица 8-5 – Спецификация режима Intra предсказания цветности и связанные с этим имена	132
Таблица 8-6 – Спецификация переменной colPic	140
Таблица 8-7 – Спецификация PicCodingStruct(X)	140
Таблица 8-8 – Спецификация mbAddrCol, yM и vertMvScale	141
Таблица 8-9 – Назначения флагов использованных предсказаний	143
Таблица 8-10 – Вывод вертикального компонента вектора цветности в режиме кодирования поля	149
Таблица 8-11 – Различное расположение полного образца яркости	153
Таблица 8-12 – Присвоение предсказанного образца яркости predPartLXL[xL, yL]	155
Таблица 8-13 – Спецификация отображения idx в sij для сканирования зигзагом и полем	164
Таблица 8-14 – Спецификация отображения idx в sij для сканирования зигзагом 8x8 и полем 8x8	166
Таблица 8-15 – Спецификация QPC как функции qPI	167
Таблица 8-16 – Вывод пороговых переменных α' и β' по indexA и indexB в зависимости от смещения	191
Таблица 8-17 – Значение переменной tC0 как функции indexA и bS	192
Таблица 9-1 – Строки битов с битами "префикса" и "суффикса" и с присвоением диапазону кода codeNum (для информации)	194
Таблица 9-2 – Строки битов Exp-Golomb и codeNum в явной форме, использованные как ue(v) (для информации)	195
Таблица 9-3 – Присвоение элементов синтаксиса значениям codeNum для элементов синтаксиса se(v), кодированных кодом Exp-Golomb со знаком	196

	Стр.
Таблица 9-4 – Присвоение codeNum параметру coded_block_pattern для режимов предсказания макроблока.....	196
Таблица 9-5 – Отображение coeff_token в TotalCoeff(coeff_token) и в TrailingOnes(coeff_token).....	200
Таблица 9-6 – Таблица кодовых слов для level_prefix (для информации).....	203
Таблица 9-7 – Таблицы total_zeros для блоков 4x4 при TotalCoeff(coeff_token) от 1 до 7.....	204
Таблица 9-8 – Таблицы total_zeros для блоков 4x4 при TotalCoeff(coeff_token) от 8 до 15.....	205
Таблица 9-9 – Таблицы total_zeros для DC блоков цветности 2x2.....	205
Таблица 9-10 – Таблицы для run_before.....	206
Таблица 9-11 – Объединение ctxIdx и элементов синтаксиса для каждого типа секций в процессе инициализации.....	208
Таблица 9-12 – Значения переменных m и n для ctxIdx от 0 до 10.....	209
Таблица 9-13 – Значения переменных m и n для ctxIdx от 11 до 23.....	210
Таблица 9-14 – Значения переменных m и n для ctxIdx от 24 до 39.....	210
Таблица 9-15 – Значения переменных m и n для ctxIdx от 40 до 53.....	210
Таблица 9-16 – Значения переменных m и n для ctxIdx от 54 до 59 и от 399 до 401.....	211
Таблица 9-17 – Значения переменных m и n для ctxIdx от 60 до 69.....	211
Таблица 9-18 – Значения переменных m и n для ctxIdx от 70 до 104.....	212
Таблица 9-19 – Значения переменных m и n для ctxIdx от 105 до 165.....	213
Таблица 9-20 – Значения переменных m и n для ctxIdx от 166 до 226.....	214
Таблица 9-21 – Значения переменных m и n для ctxIdx от 227 до 275.....	215
Таблица 9-22 – Значения переменных m и n для ctxIdx от 277 до 337.....	216
Таблица 9-23 – Значения переменных m и n для ctxIdx от 338 до 398.....	217
Таблица 9-24 – Значения переменных m и n для ctxIdx от 402 до 459.....	218
Таблица 9-25 – Элементы синтаксиса и объединенные типы бинаризации, maxBinIdxCtx и ctxIdxOffset.....	220
Таблица 9-26 – Строка бинов унарной бинаризации (для информации).....	221
Таблица 9-27 – Бинаризация типов макроблоков в I секции.....	223
Таблица 9-28 – Бинаризация типов макроблоков в P, SP и B секциях.....	224
Таблица 9-29 – Бинаризация типов субмакроблоков в P, SP и B секциях.....	225
Таблица 9-30 – Присвоение ctxIdxInc параметру binIdx для всех значений ctxIdxOffset, за исключением тех, которые относятся к элементам синтаксиса coded_block_flag, significant_coeff_flag, last_significant_coeff_flag и coeff_abs_level_minus1.....	227
Таблица 9-31 – Присвоение ctxIdxBlockCatOffset параметру ctxBlockCat для элементов синтаксиса coded_block_flag, significant_coeff_flag, last_significant_coeff_flag и coeff_abs_level_minus1.....	228
Таблица 9-32 – Спецификация ctxIdxInc для особых значений ctxIdxOffset и binIdx.....	235
Таблица 9-33 – Спецификация ctxBlockCat для различных блоков.....	235
Таблица 9-34 – Отображение позиции сканирования в ctxIdxInc для ctxBlockCat == 5.....	236
Таблица 9-35 – Спецификация RangeTabLPS в зависимости от pStateIdx и qCodiRangeIdx.....	240
Таблица 9-36 – Таблица перехода состояний.....	241
Таблица A-1 – Границы уровней.....	254
Таблица A-2 – Спецификация spbBrVclFactor и spbBrNalFactor.....	256
Таблица A-3 – Границы уровней базового профиля.....	257
Таблица A-4 – Границы уровней главного, высокого, высокого 10, высокого 4:2:2 или высокого 4:4:4 профиля.....	257
Таблица A-5 – Границы уровней расширенного профиля.....	258
Таблица A-6 – Максимальная частота кадров (кадров в секунду) для некоторых примеров размеров кадров.....	259
Таблица D-1 – Интерпретация pic_struct.....	286
Таблица D-2 – Отображение ct_type на сканирование исходного изображения.....	287
Таблица D-3 – Определение значений counting_type.....	287
Таблица D-4 – Значения scene_transition_type.....	294
Таблица D-5 – Значения model_id.....	301

	Стр.
Таблица D-6 – Значения <code>blending_mode_id</code>	302
Таблица E-1 – Значение указателя коэффициента пропорциональности образца	313
Таблица E-2 – Значение <code>video_format</code>	314
Таблица E-3 – Основные цвета	315
Таблица E-4 – Характеристика передачи	316
Таблица E-5 – Матрица коэффициентов	319
Таблица E-6 – Делитель для вычисления $\Delta t_{fi,dpb}(n)$	321

Предисловие

Международный союз электросвязи (МСЭ) является специализированным учреждением Организации Объединенных Наций в области электросвязи. Сектор стандартизации электросвязи (МСЭ-Т) – постоянный орган МСЭ. МСЭ-Т отвечает за изучение технических, эксплуатационных и тарифных вопросов и за выпуск Рекомендаций по ним с целью стандартизации электросвязи на всемирной основе. Всемирная ассамблея по стандартизации электросвязи (ВАСЭ), которая проводится каждые четыре года, определяет темы для изучения исследовательскими комиссиями МСЭ-Т, которые, в свою очередь, вырабатывают Рекомендации по этим темам. Утверждение Рекомендаций МСЭ-Т осуществляется в соответствии с процедурой, изложенной в Резолюции 1 ВАСЭ. В некоторых областях информационных технологий, которые входят в компетенцию МСЭ-Т, необходимые стандарты разрабатываются на основе сотрудничества с ИСО и МЭК.

ИСО (Международная организация по стандартизации) и МЭК (Международная электротехническая комиссия) образуют специализированную систему стандартизации на всемирном уровне. Национальные органы, являющиеся членами ИСО и МЭК, принимают участие в разработке международных стандартов через технические комитеты, учрежденные соответствующей организацией для рассмотрения вопросов в определенных сферах технической деятельности. Технические комитеты ИСО и МЭК сотрудничают в областях, представляющих взаимный интерес. В работе также принимают участие другие международные организации – правительственные и неправительственные – во взаимодействии с ИСО и МЭК. В области информационных технологий ИСО и МЭК создали объединенный технический комитет – ИСО/МЭК/ОТК 1. Принятые Объединенным техническим комитетом проекты международных стандартов рассылаются национальным органам для голосования. Для публикации в качестве Международного стандарта требуется, чтобы проект утвердили по крайней мере 75% национальных органов, участвующих в голосовании.

Настоящая Рекомендация | Международный стандарт подготовлены совместно ИК16 Q.6 МСЭ-Т, известной также как VCEG (Группа экспертов по кодированию видеосигналов), и ИСО/МЭК ОТК 1/ИК29/РГ11, известной также как MPEG (Группа экспертов по кодированию движущихся изображений). VCEG была сформирована в 1997 году для сопровождения ранее разработанных МСЭ-Т стандартов по кодированию видеосигналов и разработки нового стандарта (стандартов) по кодированию видеосигналов, пригодного для широкого диапазона речевых и неречевых служб. MPEG была создана в 1988 году для разработки стандартов по кодированию движущихся изображений и звукового сопровождения для различных применений, таких как цифровые средства хранения данных, распространение и связь.

В Приложениях А–Е к настоящей Рекомендации | Международному стандарту содержатся нормативные требования, и эти приложения являются неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Усовершенствованное кодирование изображений для общих аудиовизуальных услуг

0 Введение

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

0.1 Пролог

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Поскольку снизилась стоимость как вычислительной мощности, так и запоминающих устройств, модифицировалась сетевая поддержка кодированных видеоданных и усовершенствовалась технология кодирования видеосигналов, возросла необходимость в промышленном стандарте для сжатого представления видеосигналов с непрерывно возрастающей эффективностью кодирования и повышенной устойчивостью к сетевому окружению. С этой целью Группа экспертов по кодированию видеосигналов (VCEG) МСЭ-Т и Группа экспертов по кодированию движущихся изображений (MPEG) ИСО/МЭК создали в 2001 г. Объединенную группу по видеосигналам (JVT) для разработки новой Рекомендации | Международного стандарта.

0.2 Цель

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Настоящая Рекомендация | Международный стандарт была разработана в ответ на возросшую потребность в более высоком сжатии движущихся изображений для разных приложений, таких как видеоконференция, телевизионное вещание, цифровое хранение аудиовизуальных данных, потоков интернета и связи. Она также разработана, чтобы обеспечить гибкое представление кодированного видеосигнала для разнообразных условий сетевого окружения. Использование настоящей Рекомендации | Международного стандарта позволит управлять движущимся изображением как видом компьютерных данных, хранить изображение на различных запоминающих устройствах, а также передавать и принимать по существующим и будущим широкополосным каналам.

0.3 Применения

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Настоящая Рекомендация | Международный стандарт разработана, чтобы охватить широкий диапазон применений видеоконтента, включая (но не ограничивая) следующие области:

CATV	Cable TV on optical networks, copper, etc.	Кабельное телевидение по оптическим и медным сетям и т. д.
DBS	Direct broadcast satellite video services	Услуги видео спутникового прямого телевизионного радиовещания
DSL	Digital subscriber line video services	Услуги видео по цифровым абонентским линиям
DTTB	Digital terrestrial television broadcasting	Цифровое наземное телевизионное радиовещание
ISM	Interactive storage media (optical disks, etc.)	Интерактивные устройства хранения (оптические диски и т. д.)
MMM	Multimedia mailing	Мультимедийная почта
MSPN	Multimedia services over packet networks	Услуги мультимедиа по сетям с коммутацией пакетов
RTC	Real-time conversational services (videoconferencing, videophone, etc.)	Диалоговые услуги в реальном времени (видеоконференция, видеотелефон и т. д.)
RVS	Remote video surveillance	Удаленное видеонаблюдение
SSM	Serial storage media (digital VTR, etc.)	Серийные устройства хранения видеoinформации (цифровые видеомагнитофоны и т. д.)

0.4 Публикация и версии настоящей спецификации

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Настоящая спецификация разрабатывалась совместно Группой экспертов по кодированию видеосигналов (VCEG) МСЭ-Т и Группой экспертов по кодированию движущихся изображений ИСО/МЭК. Спецификация опубликована как согласованный в техническом аспекте парный текст в обеих организациях – МСЭ-Т и ИСО/МЭК.

Рекомендация МСЭ-Т Н.264 | ИСО/МЭК 14496-10, версия 1, является первой (2003 г.) утвержденной версией настоящей Рекомендации | Международного стандарта.

Рекомендация МСЭ-Т Н.264 | ИСО/МЭК 14496-10, версия 2, является сводным текстом, содержащим исправления, о которых говорится в первом техническом исправлении.

Рекомендация МСЭ-Т Н.264 | ИСО/МЭК 14496-10, версия 3, является сводным текстом, содержащим первое техническое исправление (2004 г.) и первую поправку, называемую "Расширение диапазона точности".

Рекомендация МСЭ-Т Н.264 | ИСО/МЭК 14496-10, версия 4 (настоящая спецификация), является сводным текстом, содержащим первое техническое исправление (2004 г.), первую поправку ("Расширение диапазона точности") и дополнительное техническое исправление (2005 г.). В МСЭ-Т следующей версией, опубликованной после версии 2, является версия 4 (поскольку проект версии 4 был завершен до того, как представилась возможность утверждения окончательного текста версии 3).

0.5 Профили и уровни

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Настоящая Рекомендация | Международный стандарт разработана, чтобы стать общей в том смысле, что она служит для широкого диапазона применений, значений скорости передачи битов, разрешающей способности, уровней качества и услуг. Кроме прочего, эти применения должны охватывать цифровые средства хранения аудиовизуальной информации, телевизионное вещание и связь в реальном времени. В процессе создания настоящей спецификации были рассмотрены различные требования для типичных применений, разработаны и объединены в единый синтаксис необходимые алгоритмические элементы. Таким образом, настоящая спецификация облегчит обмен видеоданными между различными приложениями.

Что касается практической применимости всего синтаксиса настоящей спецификации, ограниченное количество подмножеств синтаксиса обуславливается также "профилями" и "уровнями". Эти и другие, связанные с ними термины, формально определены в главе 3.

"Профиль" – это подмножество собственно синтаксиса потока битов, которое определено настоящей Рекомендацией | Международным стандартом. В границах этого синтаксиса для заданного профиля возможно все же требовать значительного разнообразия характеристик кодеров и декодеров в зависимости от значений, которые получены элементами синтаксиса в потоке битов, например заданного размера декодированных изображений. Для многих применений сейчас нет ни практического, ни экономического смысла реализовывать декодер, способный ко всем гипотетическим применениям этого синтаксиса в рамках конкретного профиля.

Чтобы рассмотреть эту проблему, в каждом профиле определены "уровни". Уровень – это особое множество ограничений, наложенных на значения элементов синтаксиса в потоке битов. Эти ограничения могут быть простыми ограничениями значений. Кроме того, они могут принимать форму ограничений на арифметические комбинации значений (например, на ширину изображения, умноженную на его высоту и умноженную на число изображений, декодированных за секунду).

Для кодированного видеоконтента, соответствующего настоящей Рекомендации | Международному стандарту, используют обычный синтаксис. Чтобы получить подмножество законченного синтаксиса, в поток битов включены флаги, параметры и другие элементы синтаксиса, сигнализирующие о присутствии или отсутствии элементов синтаксиса, которые могут позже появиться в потоке битов.

0.6 Обзор проектируемых характеристик

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Определенное в синтаксисе кодированное представление было разработано, чтобы обеспечить более высокое сжатие изображения, сохраняя при этом желательное качество этого изображения. За исключением режима работы с обходом преобразования для кодирования без потерь с профилем высокой 4:4:4 и режима работы I_PCM со всеми профилями, этот алгоритм, как правило, не является алгоритмом без потерь, поскольку точные значения образцов источника обычно не сохраняются в процессе кодирования и декодирования. Чтобы достичь высокой эффективности сжатия, можно использовать ряд методов. Для каждого изображения существует выбор между алгоритмами кодирования методами *inter* и *intra* (не определенными в настоящей Рекомендации | Международном стандарте) областей, имеющих форму блоков. При *inter* кодировании (внешнем кодировании) с помощью векторов движения производят поблочное предсказание, используя временные статистические зависимости между различными изображениями. При *intra* кодировании (внутреннем кодировании) различные пространственные режимы предсказания основаны на использовании статистических пространственных зависимостей в сигнале источника единичного изображения. Векторы движения и режимы *intra* предсказания (внутренних предсказаний) можно определить для различных размеров блоков в изображении. В дальнейшем остаток предсказания дополнительно сжимают, используя преобразование для удаления пространственной корреляции внутри преобразованного блока перед его квантованием, тем самым выполняя необратимый процесс, при котором обычно отбрасывают менее важную визуальную информацию, и в то же время формируя близкую аппроксимацию образов источника. Наконец, векторы движения или режимы *intra* предсказания объединяют с информацией квантованных коэффициентов преобразования и кодируют, используя либо арифметическое кодирование, либо коды с переменной длиной.

0.6.1 Кодирование с предсказанием

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Вследствие противоречивых требований прямого доступа и высоко эффективного сжатия определены два основных типа кодирования. Intra кодирование выполняется без сравнения с другими изображениями. Intra кодирование может обеспечить точки доступа к кодированным последовательностям в тех местах, где декодирование можно корректно начать и продолжить. Однако, как правило, при этом демонстрируется лишь умеренная эффективность сжатия. Inter кодирование (с предсказанием или с двойным предсказанием) более эффективно и использует метод inter предсказания (внешнего предсказания) значений образцов каждого блока по некоторым ранее декодированным изображениям, выбранным кодером. В противоположность некоторым другим стандартам кодирования видеосигнала, изображения, которые кодируют методом двойного inter предсказания, можно также использовать как образцы сравнения для кодирования других изображений.

Применение этих трех методов кодирования к изображениям в виде последовательности является гибким, а порядок процесса декодирования обычно не такой, как порядок процесса фиксации источника изображения в кодере или порядок отображения на выходе декодера. Выбор оставлен кодеру и зависит от требований приложения. Порядок декодирования определен таким образом, что декодирование изображений с inter предсказанием изображения происходит позже, чем декодирование прочих изображений, которые сравнивают в процессе декодирования.

0.6.2 Кодирование последовательных или чередующихся видеосигналов

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

В настоящей Рекомендации | Международном стандарте определен синтаксис и процесс декодирования видеосигналов, которые появляются в форме либо сканированных последовательных, либо сканированных с чередованием. В некоторых последовательностях эти формы могут быть смешаны. Два поля чередующегося кадра разделены во время фиксации, а два поля последовательного кадра имеют общее время фиксации. Каждое поле можно кодировать отдельно или два поля можно кодировать вместе как кадр. Последовательные кадры обычно кодируют как один кадр. При чередующихся видеосигналах кодер может выбирать между кодированием кадра и кодированием поля. Кодирование кадров или кодирование полей можно адаптивно выбирать по принципу изображение за изображением, а также по более ограниченному принципу кодирования. Кодирование кадров обычно предпочтительней в тех случаях, когда видеосцены значительно детализированы, а движение ограничено. Кодирование полей обычно лучше при быстрой смене изображений.

0.6.3 Разделение изображения на макроблоки и меньшие части

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Как и в предыдущих Рекомендациях и Международных стандартах по кодированию видеосигнала в качестве основной единицы обработки в процессе декодирования видеосигнала будет использован макроблок, состоящий из блока образцов яркости 16x16 и двух соответствующих образцов цветности.

При inter предсказании макроблок может быть далее разделен на части. Выбор размера этих частей для inter предсказания является результатом компромисса между эффективностью кодирования при использовании компенсации движения с помощью меньших блоков и количеством данных, необходимых для компенсации движения. В настоящей Рекомендации | Международном стандарте процесс inter предсказания может формировать сегментацию для представления движения небольшими по размеру образцами (например, образцами яркости 4x4), используя точность вектора движения с шагом сетки в одну четвертую образца яркости. Процесс inter предсказания блока образца может также включать выбор изображения из числа хранимых ранее декодированных изображений, которое следует использовать как контрольное для сравнения. Векторы движения кодируют по-разному, с учетом предсказанных значений смежных кодированных векторов движения.

Обычно кодер вычисляет соответствующие векторы движения и другие элементы данных, представленные в потоке видеоданных. Этот процесс оценки движения в кодере и выбор возможности использовать inter предсказание для представления каждой области видеоконтента в настоящей Рекомендации | Международном стандарте не определен.

0.6.4 Уменьшение пространственной избыточности

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Как источник изображений, так и разности от предсказаний обладают высокой пространственной избыточностью. Настоящая Рекомендация | Международный стандарт основаны на использовании метода поблочного преобразования для удаления пространственной избыточности. После inter предсказания по ранее декодированным образцам других изображений или на основе пространственного предсказания по ранее декодированным образцам текущего изображения результирующую разность предсказания разбивают на блоки 4x4. Эти блоки превращают в область преобразования, где их квантуют. После квантования многие из коэффициентов преобразования становятся нулями или имеют небольшую амплитуду и, таким образом, могут быть представлены небольшим количеством кодированных данных. Процессы квантования и преобразования не определены в настоящей Рекомендации | Международном стандарте.

0.7 Как читать настоящую спецификацию

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Предполагается, что читатель начнет с Раздела 1 (Область применения) и продвинется к Разделу 3 (Определения). Раздел 6 следует читать для понимания геометрических взаимодействий источника, входа и выхода декодера. Раздел

7 (Синтаксис и семантика) определяет порядок анализа элементов синтаксиса в потоке битов. См. пп. 7.1–7.3 для понимания синтаксического порядка, а п. 7.4 – семантики, т. е. области применения, условий и ограничений, которые наложены на элементы синтаксиса. Действительный анализ для большинства элементов синтаксиса определен в Разделе 9 (Процесс анализа). Наконец, в Разделе 8 (Процесс декодирования) определено, как элементы синтаксиса отражены в декодированных образцах. Во время чтения настоящей спецификации читателю следует при необходимости обращаться к Разделам 2 (Нормативные ссылки), 4 (Сокращения) и 5 (Условные обозначения). Приложения от А до Е также составляют неотъемлемые части настоящей Рекомендации | Международного стандарта.

В Приложении А определены семь профилей (базовый, главный, высокий, высокий 10, высокий 4:2:2 и высокий 4:4:4), каждый приспособлен для определенных областей приложений и определяет так называемые уровни профилей. В Приложении В определены синтаксис и семантика формата потока байтов для доставки кодированного видеосигнала в виде упорядоченного потока байтов. В Приложении С определен гипотетический контрольный декодер и его использование для контроля потока битов и соответствия этого декодера. В Приложении D определены синтаксис и семантика полезной нагрузки дополнительной расширенной информации. Наконец, в Приложении Е определены синтаксис и семантика удобства использования информационных параметров видеосигнала из набора параметров последовательности.

Присутствующие в настоящей спецификации сообщения с преамбулой "ПРИМЕЧАНИЕ" считаются информативными и не являются неотъемлемой частью настоящей Рекомендации | Международного стандарта.

1 Сфера применения

В настоящем документе содержится Рекомендация МСЭ-Т Н.264 | Международный стандарт ИСО/МЭК 14496-10 по кодированию изображений.

2 Нормативные ссылки

Указанные ниже Рекомендации и Международные стандарты содержат положения, которые путем ссылки на них в данном тексте составляют положения настоящей Рекомендации | Международного стандарта. На момент публикации указанные издания были действующими. Все Рекомендации и стандарты могут подвергаться пересмотру, поэтому участникам соглашений, основанных на настоящей Рекомендации | Международном стандарте, предлагается изучить возможность применения последнего издания Рекомендаций и стандартов, перечисленных ниже. Члены ИСО/МЭК ведут регистры действующих в настоящее время международных стандартов. Бюро стандартизации электросвязи МСЭ ведет список действующих в настоящее время Рекомендаций МСЭ-Т.

- ITU-T Recommendation T.35 (2000), *Procedure for the allocation of ITU-T defined codes for non-standard facilities*.
- ISO/IEC 11578:1996, Annex A, *Universal Unique Identifier*.
- ISO/CIE 10527:1991, *Colorimetric Observers*.

3 Определения

В настоящей Рекомендации | Международном стандарте использованы следующие определения.

- 3.1 блок доступа (access unit):** Набор блоков уровней *NAL*, в которых всегда содержится только одно начальное кодированное изображение. Кроме начального кодированного изображения, блок доступа может также содержать одно или более кодированных дополнительных изображений или другие блоки *NAL*, в которых отсутствуют секции или данные разделения кодированного изображения на секции. Результатом декодирования блока доступа всегда будет декодированное изображение.
- 3.2 коэффициент преобразования AC (AC transform coefficient):** Любой коэффициент преобразования, для которого одно- или двумерный индекс частоты не является нулевым.
- 3.3 процесс адаптивного арифметического бинарного декодирования (adaptive binary arithmetic decoding process):** Энтропийный *декодировующий процесс*, который извлекает значения битов из потока битов, создаваемого процессом *адаптивного арифметического бинарного кодирования*.
- 3.4 процесс адаптивного арифметического бинарного кодирования (adaptive binary arithmetic encoding process):** Энтропийный *кодировующий процесс*, не обязательно определенный этой Рекомендацией | Международным стандартом, с помощью которого кодируют бинарную последовательность и создают *поток битов*, который декодируют с помощью *процесса адаптивного арифметического бинарного декодирования*.
- 3.5 альфа-смешение (alpha blending):** Процесс, не описываемый настоящей Рекомендацией | Международным стандартом, в котором *дополнительно кодированное изображение* используется в комбинации с *первично кодированным изображением* и с другими, не описываемыми настоящей Рекомендацией | Международным стандартом, данными в *процессе отображения*. В процессе альфа-смешения образцы *дополнительно кодированного изображения* интерпретируются как индикация степени непрозрачности (или, что то же самое, степени прозрачности), связанная с соответствующими образцами *яркости первично кодированного изображения*.

- 3.6 произвольный порядок секций (arbitrary slice order):** Порядок декодирования *секций*, в котором *адрес макроблока* первого макроблока некоторой секции изображения может быть меньше, чем адрес макроблока первого макроблока другой предшествующей секции того же самого кодированного изображения.
- 3.7 дополнительно кодированное изображение (auxiliary coded picture):** *Изображение*, которое дополняет *первично кодированное изображение* и которое может использоваться в комбинации с другими данными, не описываемыми настоящей Рекомендацией | Международным стандартом, в *процессе отображения*. Дополнительно кодированное изображение имеет те же синтаксические и семантические ограничения, что и *монохромное избыточно кодированное изображение*. Дополнительно кодированное изображение должно содержать то же количество *макроблоков*, что и *первично кодированное изображение*. Дополнительно кодированное изображение не имеет нормативного воздействия на *процесс декодирования*. См. также *первично кодированное изображение* и *избыточно кодированное изображение*.
- 3.8 В секция (B slice):** *Секция*, которую можно декодировать, используя внутреннее предсказание (intra предсказание) от декодированных образцов внутри той же *секции*, или предсказание от первоначально декодированных *контрольных изображений*, используя не более двух *векторов движения* и *контрольные индексы* для *предсказания* значений образцов каждого блока.
- 3.9 бин (bin):** Один бит из *строки бинов*.
- 3.10 бинаризация (binarization):** Преобразование в двоичную форму – набор *строк бинов* для всех возможных значений *элемента синтаксиса*.
- 3.11 процесс бинаризации (binarization process):** Процесс преобразования в двоичную форму. Единственный процесс размещения всех возможных значений *элемента синтаксиса* в набор *строк бинов*.
- 3.12 строка бинов (bin string):** Строка бинов является промежуточным бинарным представлением значений *элементов синтаксиса* от преобразования в двоичную форму *элемента синтаксиса*.
- 3.13 В предсказанная секция (bi-predictive slice):** См. *В секция*.
- 3.14 поток битов (bitstream):** Последовательность битов, которая формирует представление *кодированных изображений* и связана с данными, формирующими одну или более *кодированных видеопоследовательностей*. Поток битов – это наиболее общий термин, используемый для обозначения либо потока *блоков уровней NAL*, либо *потока байтов*.
- 3.15 блок (block):** Массив $M \times N$ (M -столбцов на N -строк) образцов или матрица $M \times N$ *коэффициентов преобразования*.
- 3.16 нижнее поле (bottom field):** Нижнее поле – одно из двух *полей*, которые составляют *кадр*. Каждый ряд нижнего поля расположен в пространстве непосредственно под рядом соответствующего верхнего поля.
- 3.17 нижний макроблок (из пары макроблоков) (bottom macroblock (of a macroblock pair)):** Макроблок внутри пары макроблоков, который содержит образцы нижнего ряда образцов этой *пары макроблоков*. Для *пары макроблоков поля* нижний макроблок представляет образцы из области нижнего поля кадра, который находится в пространственной области *пары макроблоков*. Для *пары макроблоков кадра* нижний макроблок представляет образцы *кадра*, который находится в нижней половине пространственной области *пары макроблоков*.
- 3.18 разомкнутое звено связи (broken link):** Расположение в *потоке битов*, которое означает, что некоторые последующие *изображения* в *порядке декодирования* могут содержать серьезные визуальные дефекты вследствие неточно указанных действий, выполненных при организации *потока битов*.
- 3.19 байт (byte):** Последовательность из 8 битов, записанная и прочитанная со старшим значащим битом слева, а младшим значащим битом справа. При представлении в последовательности битов данных старший значащий бит байта является первым.
- 3.20 байт-синхронизованный (byte-aligned):** Положение в *потоке битов* считают байт-синхронизованным, когда это положение является целым кратным 8 битам от позиции первого бита в *потоке битов*. Бит, *байт* или *элемент синтаксиса* считают байт-синхронизованными, если позиция, на которой они появляются в *потоке битов*, является байт-синхронизованной.
- 3.21 поток байтов (byte stream):** Инкапсуляция *потока блоков NAL*, в котором содержатся *префиксы кодов запуска* и *блоки NAL*, как это определено в Приложении В.
- 3.22 может (в смысле возможности) (can):** Термин (can), который используют, чтобы определить действие, которое разрешено, но не обязательно требуется.
- 3.23 категория (category):** Число, связанное с каждым *элементом синтаксиса*. Категорию используют для определения расположения *элементов синтаксиса* в *блоках NAL* для *разделения данных секции*. Этот термин можно также использовать в смысле, определенном приложением по отношению к классам *элементов синтаксиса* в понятиях, которые не определены в настоящей Рекомендации | Международном стандарте.
- 3.24 цветовой (chroma):** Определение, указывающее, что образец массива или простой образец представляют один из двух цветоразностных сигналов, связанных с основными цветами. Символы, которые используют для цветových массивов или образцов, обозначают как C_b и C_r .
- ПРИМЕЧАНИЕ. – Термин "цветовой" используют чаще, чем термин "хроматический", чтобы избежать затруднений при использовании линейных характеристик передачи света, которые часто связывают с термином "хроматический".
- 3.25 кодированное поле (coded field):** *Кодированное представление поля*.

- 3.26 кодированный кадр (coded frame):** *Кодированное представление кадра.*
- 3.27 кодированное изображение (coded picture):** *Кодированное представление изображения. Кодированное изображение может быть либо кодированным полем, либо кодированным кадром. Кодированное изображение является собирательным термином, относящимся к основному кодированному изображению или к дополнительно кодированному изображению, но не к обоим сразу.*
- 3.28 буфер кодированного изображения (coded picture buffer, CPB):** *Буфер, действующий по принципу "первым пришел – первым обслужен", который содержит блоки доступа в порядке декодирования, определенном в Приложении С гипотетическим декодером контроля.*
- 3.29 кодированное представление (coded representation):** *Элемент данных, представленный в кодированной форме.*
- 3.30 кодированная видеопоследовательность (coded video sequence):** *Последовательность блоков доступа, состоящая (в порядке декодирования) из блоков доступа IDR, за которыми следуют нуль или другие блоки (но не доступа IDR). Блоки доступа IDR должны отсутствовать во всех последующих блоках доступа.*
- 3.31 компонент (component):** *Массив или простой образец одного из трех массивов (одного яркости и двух цветности), которые образуют поле или кадр.*
- 3.32 пара дополнительных полей (complementary field pair):** *Обобщенный термин для пар дополнительных контрольных полей и дополнительных неконтрольных полей.*
- 3.33 пара дополнительных неконтрольных полей (complementary non-reference field pair):** *Два неконтрольных поля, которые расположены в последовательных блоках доступа в порядке декодирования в виде двух кодированных полей с инверсным равенством, где первое поле уже не является парным полем.*
- 3.34 пара дополнительных контрольных полей (complementary reference field pair):** *Два контрольных поля, которые расположены в последовательных блоках доступа в порядке декодирования в виде двух кодированных полей и разделяют то же значение элемента синтаксиса frame_num, и второе поле в порядке декодирования не является изображением IDR и не включает элемент синтаксиса memory_management_control_operation, равный 5.*
- 3.35 контекстная переменная (context variable):** *Переменная, определенная для процесса адаптивного арифметического бинарного декодирования бина с помощью уравнения, в котором содержатся последние декодированные бины.*
- 3.36 коэффициент преобразования по постоянному току (DC transform coefficient):** *Коэффициент преобразования, для которого индекс частоты является нулем во всех координатах.*
- 3.37 декодированное изображение (decoded picture):** *Декодированное изображение, полученное декодированием кодированного изображения. Декодированное изображение является либо декодированным кадром, либо декодированным полем. Декодированное поле является либо декодированным верхним полем, либо декодированным нижним полем.*
- 3.38 буфер декодированного изображения (decoded picture buffer, DPB):** *Буфер, содержащий декодированные изображения для контроля, изменения порядка или задержки на выходе, как это определено в Приложении С гипотетическим декодером контроля.*
- 3.39 декодер (decoder):** *Устройство для выполнения процесса декодирования.*
- 3.40 порядок декодирования (decoding order):** *Порядок, в котором элементы синтаксиса обрабатывают в процессе декодирования.*
- 3.41 процесс декодирования (decoding process):** *Процесс, определенный в настоящей Рекомендации | Международном стандарте, который считывает поток битов и создает из этого потока декодированные изображения.*
- 3.42 прямое предсказание (direct prediction):** *Внешнее предсказание (inter предсказание) для блока, для которого не декодируют никакого вектора движения. Определены два режима прямого предсказания, которые обозначают как пространственное прямое предсказание и временное предсказание.*
- 3.43 процесс отображения (display process):** *Процесс, не описываемый в настоящей Рекомендации | Международном стандарте, входами которого служат кадрированные декодированные изображения, которые являются выходом процесса декодирования.*
- 3.44 тестируемый декодер (decoder under test, DUT):** *Декодер, тестируемый на соответствие настоящей Рекомендации | Международному стандарту с помощью гипотетического планировщика потока, который подает соответствующий поток битов на этот декодер и на гипотетический контрольный декодер и сравнивает значения и временные положения сигналов на выходах этих двух декодеров.*
- 3.45 байт предотвращения эмуляции (emulation prevention byte):** *Байт, равный 0x03, который можно представить блоком NAL. Присутствие байтов предотвращения эмуляции гарантирует, что последовательность следующих друг за другом байт-синхронизированных байтов в блоках NAL не содержит префикса кода запуска.*
- 3.46 кодер (encoder):** *Устройство для выполнения процесса кодирования.*
- 3.47 процесс кодирования (encoding process):** *Процесс, не определенный в настоящей Рекомендации | Международном стандарте, который создает поток битов, соответствующих настоящей Рекомендации | Международному стандарту.*
- 3.48 поле (field):** *Комплект чередующихся рядов кадров. Кадр состоит из двух полей: верхнего и нижнего.*
- 3.49 макроблок поля (field macroblock):** *Макроблок, содержащий образцы из единственного поля. Все макроблоки кодированного поля являются макроблоками поля. Если используют адаптивное декодирование макроблоками кадра/поля, некоторые макроблоки кодированного кадра могут быть макроблоками поля.*

- 3.50 пара макроблоков поля (field macroblock pair):** Пара макроблоков, декодированная как два макроблока поля.
- 3.51 сканирование полей (field scan):** Особая последовательность порядка коэффициентов преобразования, которая отличается от сканирования зигзагом тем, что сканирование столбцов происходит быстрее сканирования рядов. Сканирование полей используют для коэффициентов преобразования в макроблоках поля.
- 3.52 флаг (flag):** Переменная, которая может принимать одно из двух значений: 0 и 1.
- 3.53 кадр (frame):** Кадр содержит один массив образцов яркости и два соответствующих массива образцов цветности. Кадр состоит из двух полей: верхнего и нижнего.
- 3.54 макроблок кадра (frame macroblock):** Макроблок, представляющий образцы из двух полей кодированного кадра. Если не используют адаптивное декодирование макроблоками кадра/поля, все макроблоки кодированного кадра являются макроблоками кадра. Если используют адаптивное декодирование макроблоками кадра/поля, некоторые макроблоки кодированного кадра могут быть макроблоками кадра.
- 3.55 пара макроблоков кадра (frame macroblock pair):** Пара макроблоков, декодированная как два макроблока кадра.
- 3.56 индекс частоты (frequency index):** Одно- или двумерный индекс, связанный с коэффициентом преобразования перед этапом инверсного преобразования процесса декодирования.
- 3.57 гипотетический контрольный декодер (hypothetical reference decoder, HRD):** Модель гипотетического декодера, которая определяет ограничения на изменчивость соответствующих потоков блоков NAL или соответствующих потоков байтов, которые могут появляться в процессе кодирования.
- 3.58 гипотетический планировщик потоков (hypothetical stream scheduler, HSS):** Гипотетический механизм подачи потоков синхронизации и данных из входного потока битов в гипотетический контрольный декодер. Планировщик HSS используют для контроля соответствия потока битов или декодера.
- 3.59 I секция (I slice):** Не является SI секцией. Эту секцию декодируют, используя предсказание, полученное только от образцов внутри этой же секции.
- 3.60 информативный (informative):** Термин, применяемый к содержанию настоящей Рекомендации | Международного стандарта, который не является неотъемлемой частью настоящей Рекомендации | Международного стандарта. Для информативного содержания не устанавливают никаких обязательных требований на соответствие настоящей Рекомендации | Международному стандарту.
- 3.61 блок доступа мгновенного декодирующего восстановления (IDR) (instantaneous decoding refresh (IDR) access unit):** Блок доступа, в котором первично кодированное изображение является изображением IDR.
- 3.62 изображение мгновенного декодирующего восстановления (IDR) (instantaneous decoding refresh (IDR) picture):** Кодированное изображение, в котором все секции являются секциями I или SI, вследствие чего процесс декодирования помечает все контрольные изображения как "не используется для контроля" непосредственно после декодирования изображения IDR. После декодирования изображения IDR все последующие кодированные изображения в порядке декодирования могут быть декодированы без внутреннего предсказания (intra предсказания) из любого изображения, декодированного перед изображением IDR. Первое изображение каждой кодированной видеопоследовательности является изображением IDR.
- 3.63 внешнее кодирование (inter кодирование) (inter coding):** Кодирование блока, макроблока, секции или изображения, при котором используют внешнее предсказание (inter предсказание).
- 3.64 внешнее предсказание (inter предсказание) (inter prediction):** Предсказание, полученное из декодированных образцов контрольных изображений, иных чем текущее декодированное изображение.
- 3.65 значение образца интерпретации (interpretation sample value):** возможно измененное значение, соответствующее значению декодированного образца дополнительно кодированного изображения, которое может создаваться для использования в процессе отображения. Значения образца интерпретации не используются в процессе декодирования и не имеют нормативного воздействия на процесс декодирования.
- 3.66 внутреннее кодирование (intra кодирование) (intra coding):** Кодирование блока, макроблока, секции или изображения, при котором используют внутреннее предсказание (intra предсказание).
- 3.67 внутреннее предсказание (intra предсказание) (intra prediction):** Предсказание, полученное из декодированных образцов той же самой декодированной секции.
- 3.68 внутренняя секция (intra slice):** См. I секция.
- 3.69 инверсное преобразование (inverse transform):** Часть процесса декодирования, с помощью которого набор коэффициентов преобразования преобразуют в пространственно распределенные значения или с помощью которого набор коэффициентов преобразования преобразуют в коэффициенты преобразования DC.
- 3.70 слой (layer):** Один из наборов синтаксических структур с неветвящейся иерархической зависимостью. Высшие слои содержат низшие слои. Кодированными являются слои кодированной видеопоследовательности, изображения, секции и макроблока.
- 3.71 уровень (level):** Определенный набор ограничений на значения, которые могут принимать элементы синтаксиса и переменные в настоящей Рекомендации | Международном стандарте. Этот же набор уровней определен для всех профилей. Большинство аспектов определения каждого уровня являются общими для разных профилей. Для частных применений могут существовать особые ограничения, а также может поддерживаться разный уровень для каждого разрешенного профиля. В различных контекстах уровень представляет значение степени коэффициента преобразования до его масштабирования.

- 3.72 **вектор движения списка 0 (списка 1) (list 0 (list 1) motion vector):** Вектор движения, связанный с индексом контроля, указывающим на контрольное изображение списка 0 (списка 1).
- 3.73 **предсказание списка 0 (списка 1) (list 0 (list 1) prediction):** Внешнее предсказание содержания секции, использующее индекс контроля, который указывает на контрольное изображение списка 0 (списка 1).
- 3.74 **яркость (luma):** Определение, которое указывает, что образец массива или простой образец представляют монохроматический сигнал, связанный с первоначальными цветами. Символ или надпись для яркости обозначают как Y или L.
 ПРИМЕЧАНИЕ. – Термин "яркость" используют чаще, чем термин "сигнал яркости", чтобы избежать затруднений при использовании линейных характеристик передачи света, которые часто связывают с термином "сигнал яркости". Иногда символ L используют вместо символа Y, чтобы избежать путаницы с символом "y", который используют для обозначения вертикального направления.
- 3.75 **макроблок (macroblock):** Блок образцов яркости 16x16 или двух соответствующих блоков образцов цветности. Деление секции или пары макроблоков на макроблоки называют разделением на части (блоки).
- 3.76 **адаптивное декодирование макроблоками кадра/поля (macroblock-adaptive frame/field decoding):** Процесс декодирования кодированных кадров, в котором некоторые макроблоки могут быть декодированы как макроблоки кадров, а некоторые – как макроблоки поля.
- 3.77 **адрес макроблока (macroblock address):** Если не используют адаптивное декодирование макроблоками кадра/поля, то адресом является индекс макроблока при растровом сканировании макроблока изображения начиная с верхнего левого макроблока изображения. Если используют адаптивное декодирование макроблоками кадра/поля, то адресом верхнего из пары макроблоков является удвоенный индекс пары макроблоков растрового сканирования изображения, а адресом нижнего из пары макроблоков является адрес соответствующего верхнего макроблока плюс 1. Адрес верхнего макроблока из каждой пары макроблоков – это четное число, а адрес нижнего макроблока из каждой пары макроблоков – это нечетное число.
- 3.78 **местоположение макроблока (macroblock location):** Двумерные координаты макроблока в изображении, обозначенные как (x, y). Для верхнего левого макроблока в изображении значения (x, y) равны (0, 0). "x" изменяется на 1 для каждого столбца макроблоков слева направо. Если не используют адаптивное декодирование макроблоками кадра/поля, то "y" возрастает на 1 для каждого ряда макроблоков сверху вниз. Если используют адаптивное декодирование макроблоками кадра/поля, то "y" возрастает на 2 для каждого ряда пар макроблоков сверху вниз и дополнительно возрастает на 1, если макроблок является нижним макроблоком.
- 3.79 **пара макроблоков (macroblock pair):** Пара вертикальных смежных макроблоков в кадре, которые сдвоены для использования при адаптивном декодировании макроблоками кадра/поля. Деление секции на пару макроблоков – это разделение на части (partitioning).
- 3.80 **разделение на части макроблока (macroblock partition):** Блок образцов яркости и два соответствующих блока образцов цветности, которые получают в результате разделения на части макроблока для внешнего предсказания (inter предсказания).
- 3.81 **отображение макроблока в группу секции (macroblock to slice group map):** Средство отображения макроблоков изображения в группы секций. Отображение макроблока в группу секции состоит из перечня чисел (одного для каждого кодированного макроблока), определяющих группу секции, к которой принадлежит каждый кодированный макроблок.
- 3.82 **отображение блока в группу секции (map unit to slice group map):** Средство отображения блоков изображения в группы секций. Отображение блока в группу секции состоит из перечня чисел (одного для каждого отображенного блока), определяющих группу секции, к которой принадлежит каждый кодированный блок, отображенный в группу секций.
- 3.83 **может (в смысле пожелания) (may):** Термин (may), который используют, чтобы определить действие, которое разрешено, но не обязательно требуется. В некоторых местах, в которых желают подчеркнуть дополнительные свойства описываемого поведения, для выразительности используют фразу "может быть, а может и не быть".
- 3.84 **операция управления памятью (memory management control operation):** Семь операций, которые управляют разметкой контрольного изображения.
- 3.85 **вектор движения (motion vector):** Двумерный вектор, используемый для внешнего предсказания (inter предсказания), который создает сдвиг от координат декодированного изображения к координатам контрольного изображения.
- 3.86 **должен (must):** Термин (must), используемый для выражения наблюдения относительно требования или использования требования, которое оговорено где-либо в настоящей Рекомендации | Международном стандарте. Этот термин используется исключительно в информативном контексте.
- 3.87 **блок NAL (NAL unit):** Синтаксическая структура, содержащая указание типа данных, которые должны последовать, а также байты, содержащие эти данные в форме Rbsp, которые передают при необходимости вместе с байтами предотвращения эмуляции.
- 3.88 **поток блоков NAL (NAL unit stream):** Последовательность блоков NAL.
- 3.89 **несдвоенное поле (non-paired field):** Собираемый термин для обозначения несдвоенного (непарного) контрольного поля или несдвоенного неконтрольного поля.
- 3.90 **несдвоенное неконтрольное поле (non-paired non-reference field):** Декодированное неконтрольное поле, которое не является частью пары дополнительных неконтрольных полей.

- 3.91 **несдвоенное контрольное поле (non-paired reference field):** Декодированное контрольное *поле*, которое не является частью пары дополнительных контрольных полей.
- 3.92 **неконтрольное поле (non-reference field):** *Поле*, кодированное значением `nal_ref_idc`, равным 0.
- 3.93 **неконтрольный кадр (non-reference frame):** *Кадр*, кодированный значением `nal_ref_idc`, равным 0.
- 3.94 **неконтрольное изображение (non-reference picture):** *Изображение*, кодированное значением `nal_ref_idc`, равным 0. *Неконтрольное изображение* не используют для *внешнего предсказания* других *изображений*.
- 3.95 **примечание (note):** Термин, используемый для присоединения информативных замечаний. Этот термин используется исключительно в информативном контексте.
- 3.96 **инверсное равенство (opposite parity):** Пример инверсного равенства: вверх – это низ и наоборот.
- 3.97 **порядок выхода (output order):** Порядок, в котором *декодированные изображения* появляются на выходе буфера *декодированного изображения*.
- 3.98 **Р секция (P slice):** *Секция*, которую можно декодировать с помощью *внутреннего предсказания (intra предсказания)* от декодированных образцов той же самой *секции* или *внешнего предсказания (inter предсказания)* от ранее декодированных *контрольных изображений*, используя не более одного *вектора движения* и *индекс контроля*, чтобы предсказать значения образцов каждого блока.
- 3.99 **параметр (parameter):** *Элемент синтаксиса установки параметров последовательности* или *установки параметров изображения*. Параметр используют также как часть определенного термина *параметр квантования*.
- 3.100 **равенство (parity):** Равенством *поля* может быть верх или низ.
- 3.101 **разделение на части (partitioning):** Деление множества на подмножества таким образом, что каждый элемент множества является в точности элементом этого подмножества.
- 3.102 **изображение (picture):** Собираемый термин для *поля* или *кадра*.
- 3.103 **установка параметра изображения (picture parameter set):** Синтаксическая структура, содержащая *элементы синтаксиса*, которые применимы к нулю или более законченным частям *кодированных изображений*, как это определено *элементом синтаксиса* `pic_parameter_set_id`, расположенным в каждом заголовке *секции*.
- 3.104 **счетчик последовательности изображения (picture order count):** Переменная, принимающая неубывающие значения при возрастании позиции *изображения* в последовательности на выходе относительно первоначального *изображения IDR* в *порядке декодирования* или относительно первоначального *изображения*, содержащегося в операции управления памятью, которая помечает все *контрольные изображения* как "не используется для контроля".
- 3.105 **предсказание (prediction):** Устройство для выполнения *процесса предсказания*.
- 3.106 **процесс предсказания (prediction process):** Использование *предсказателя* для оценки значения образца или элементов данных, декодируемых в этот момент.
- 3.107 **предсказанная секция (predictive slice):** См. Р секция.
- 3.108 **предсказатель (predictor):** Комбинация особых значений или ранее декодированных значений образцов или элементов данных, использованных в *процессе декодирования* очередных значений образцов или элементов данных.
- 3.109 **первично кодированное изображение (primary coded picture):** Кодированное представление *изображения*, которое должно использоваться *процессом декодирования* для подтверждения соответствия потока битов настоящей Рекомендации | Международному стандарту. Первично кодированное изображение содержит все *макроблоки изображения*. Единственные *изображения*, которые оказывают воздействие на *процесс декодирования*, – это первично кодированные изображения. См. также *избыточно кодированное изображение*.
- 3.110 **профиль (profile):** Особое подмножество синтаксиса настоящей Рекомендации | Международного стандарта.
- 3.111 **параметр квантования (quantisation parameter):** Переменная, используемая *процессом декодирования* для масштабирования *степеней коэффициентов преобразования*.
- 3.112 **произвольный доступ (random access):** Действие, запускающее процесс декодирования *потока битов* в произвольной (не начальной) точке этого потока.
- 3.113 **растровое сканирование (raster scan):** Отображение прямоугольного двумерного шаблона в одномерный шаблон с таким свойством, что первые входы одномерного шаблона получены из первого верхнего ряда двумерного шаблона, сканированного слева направо. Далее следуют аналогичные действия для второго, третьего и т. д. рядов (сверху вниз), каждый их которых сканируют слева направо.
- 3.114 **полезная нагрузка последовательности исходных байтов (raw byte sequence payload, RBSP):** Синтаксическая структура, в которой содержится целое число *байтов*, инкапсулированных в блок *NAL*. Нагрузка RBSP может быть либо пустой, либо иметь форму строки битов данных с *элементами синтаксиса*, за которыми следует *стоповый бит RBSP*, а далее нуль или последующие биты, равные 0.
- 3.115 **стоповый бит полезной нагрузки последовательности исходных байтов (raw byte sequence payload (RBSP) stop bit):** Равный 1 бит, представленный в *полезной нагрузке последовательности исходных байтов (RBSP)*, после *строки битов данных*. Местоположение конца *строки битов данных* в *RBSP* может быть определено поиском от конца *RBSP* *стопового бита RBSP*, который является последним ненулевым битом *RBSP*.

- 3.116 точка восстановления (recovery point):** Точка в *потоке битов*, в которой восстановление точного или приближенного представления *декодированных изображений*, представленных *потоком битов*, достигают с помощью *произвольного доступа* или *разомкнутого звена связи*.
- 3.117 избыточно кодированное изображение (redundant coded picture):** Кодированное представление *изображения* или части *изображения*. Содержание избыточно кодированного изображения не должно использоваться *процессом декодирования* для *потока битов*, соответствующих настоящей Рекомендации | Международному стандарту. Избыточно кодированное изображение не требует удержания всех *макроблоков* в первично кодированном изображении. Избыточно кодированное изображение не оказывает воздействия на *процесс декодирования*. См. также *первично кодированное изображение*.
- 3.118 контрольное поле (reference field):** *Контрольное поле* может быть использовано для *внешнего предсказания*, если декодированы *секции P, SP и B* кодированного поля или поля *макроблоков* кодированного кадра. См. также *контрольное изображение*.
- 3.119 контрольный кадр (reference frame):** *Контрольный кадр* может быть использован для *внешнего предсказания*, если декодированы *секции P, SP и B* кодированного кадра. См. также *контрольное изображение*.
- 3.120 индекс контроля (reference index):** Индекс в списке *контрольных изображений*.
- 3.121 контрольное изображение (reference picture):** *Изображение* с синтаксисом `nal_ref_idc`, не равным 0. *Контрольное изображение* содержит образцы, которые могут быть использованы для *внешнего предсказания (inter предсказания)* в *процессе декодирования* последующих *изображений* в *порядке декодирования*.
- 3.122 список контрольных изображений (reference picture list):** Список *контрольных изображений*, который используется для *внешнего предсказания* *секции P, B или SP*. Для *процесса декодирования* *секции P или SP* существует один список *контрольных изображений*. Для *процесса декодирования* *секции B* существуют два списка *контрольных изображений*.
- 3.123 список контрольных изображений 0 (reference picture list 0):** *список контрольных изображений*, использованных для *внешнего предсказания* *секций P, B или SP*. *Список контрольных изображений 0* использует все *внешние предсказания (inter предсказания)*, которые были использованы для *секций P и SP*. *Список контрольных изображений 0* – это один из двух списков *контрольных изображений*, использованных для *внешнего предсказания B-секции* совместно с другим *списком контрольных изображений 1*.
- 3.124 список контрольных изображений 1 (reference picture list 1):** *Список контрольных изображений*, использованных для *внешнего предсказания B-секции*. *Список контрольных изображений 1* – это один из двух списков *контрольных изображений*, использованных для *внешнего предсказания B секции* совместно с другим *списком контрольных изображений 0*.
- 3.125 метка контрольного изображения (reference picture marking):** Определяет в *потоке битов*, каким образом *декодированные изображения* помечены для *внешнего предсказания*.
- 3.126 зарезервировано (reserved):** В разделах, определяющих некоторые особенности *элемента синтаксиса*, термин "зарезервировано" применяют для будущего использования МСЭ-Т | ИСО/МЭК. Эти значения не должны использоваться при описании *потоков битов*, соответствующих настоящей Рекомендации | Международному стандарту, но могут быть использованы в будущих расширениях настоящей Рекомендации | Международного стандарта организациями МСЭ-Т | ИСО/МЭК.
- 3.127 разность (residual):** Разность при декодировании между *предсказанием* элемента образца или данных и его декодированным значением.
- 3.128 проход (run):** Число последовательных элементов данных, представленных в процессе декодирования. В одном контексте это число степеней *коэффициентов преобразования* с нулевым значением, предшествующих числу степеней *коэффициентов преобразования* с ненулевым значением в перечне степеней *коэффициентов преобразования*, который получают *сканированием зигзагом* или *сканированием полем*. В других контекстах *проходом* называют число *макроблоков*.
- 3.129 коэффициент (внешнего) вида образца (sample aspect ratio):** В процессе отображения, который не определяется в настоящей Рекомендации | Международном стандарте, – это отношение предполагаемого расстояния между столбцами по горизонтали к предполагаемому расстоянию между рядами по вертикали в массиве образцов яркости в *кадре*. Коэффициент вида образца выражают как $h:v$, где h – ширина по горизонтали, а v – высота по вертикали (в произвольных единицах пространственного расстояния).
- 3.130 масштабирование (scaling):** процесс умножения *степеней коэффициентов преобразования* на множитель, в результате чего получают *коэффициенты преобразования*.
- 3.131 установка параметра последовательности (sequence parameter set):** структура синтаксиса, содержащая *элементы синтаксиса*, которые применяют к нулевым и более полным *кодированным видеопоследовательностям*, как это определено содержанием *элемента синтаксиса seq_parameter_set_id*, который находится в *установке параметра изображения*, относящейся к *элементу синтаксиса pic_parameter_set_id*, расположенному в каждом *заголовке секции*.
- 3.132 должен (shall):** Термин (shall) для выражения обязательности требований соответствия настоящей Рекомендации | Международному стандарту. Термин используют, чтобы выразить обязательность ограничений на значения *элементов синтаксиса* или на результаты, полученные действием точно определенного *процесса декодирования*. Это ограничение выполняет кодер. При использовании в ссылке на действия, выполняемые *процессом декодирования*, подразумевают любой *процесс декодирования*, который приводит к идентичным описанным здесь результатам, соответствующим требованиям настоящей Рекомендации | Международного стандарта к *процессу декодирования*.

- 3.133 **следует (should)**: Термин (should) для описания поведения при реализации, которое предполагает ординарные обстоятельства, но не обязательного требования для соответствия настоящей Рекомендации | Международному стандарту.
- 3.134 **SI секция (SI slice)**: Секция, которая кодирована с использованием *предсказания* только от декодированных образцов этой же секции, а также с использованием квантования предсказанных образцов. SI секция может быть кодирована таким образом, что ее декодированные образцы будут идентичны образцам SP секции.
- 3.135 **пропущенный макроблок (skipped macroblock)**: Макроблок, для которого отсутствуют иные данные, кроме указания, что этот макроблок предполагают декодировать как "пропущенный". Такое указание может быть общим для нескольких макроблоков.
- 3.136 **секция (slice)**: Целое число макроблоков или пар макроблоков, расположенных последовательно при растровом сканировании в данной группе секций. Для первично кодированного изображения деление каждой группы секций на секции представляет *разделение на части*. Хотя секция содержит макроблоки или пары макроблоков, которые поступают последовательно при растровом сканировании в группе секций, эти макроблоки или пары макроблоков не обязательно поступают последовательно при растровом сканировании изображения. Адреса макроблоков находят по адресу первого макроблока в секции (как представленных в заголовке секции) и по *отображению макроблока в группу секции*.
- 3.137 **деление на части данных секции (slice data partitioning)**: Метод деления на части выбранных элементов синтаксиса в структурах синтаксиса, основанных на категориях, которые связаны с каждым элементом синтаксиса.
- 3.138 **группа секций (slice group)**: Подмножество макроблоков или пар макроблоков изображения. Деление изображения на группы секций представляет деление изображения на части. Деление на части определено отображением макроблока в группу секции.
- 3.139 **блоки отображения группы секций (slice group map units)**: Блоки отображения макроблока в группу секции.
- 3.140 **заголовок секции (slice header)**: Часть кодированной секции, содержащая элементы данных, которые принадлежат первому или всем макроблокам, представленным в этой секции.
- 3.141 **источник (source)**: Термин, используемый для описания видеоматериала или некоторых его атрибутов перед кодированием.
- 3.142 **SP секция (SP slice)**: Секция, которую кодируют, используя *внешнее предсказание* от ранее декодированных контрольных изображений. При этом используют не более одного вектора движения и индекса контроля для предсказания значений образца каждого блока. SP секция может быть кодирована таким образом, что ее декодированные образцы будут идентичны образцам другой SP секции или SI секции.
- 3.143 **префикс кода запуска (start code prefix)**: Особая последовательность из трех байтов, равных 0x000001 и включенных в поток байтов в виде префикса каждого блока NAL. Расположение префикса кода запуска может быть использовано декодером для идентификации начала нового блока NAL и конца предыдущего блока NAL. Эмуляцию префиксов кода запуска в блоках NAL предотвращают включением байтов предотвращения эмуляции.
- 3.144 **строка битов данных (string of data bits, SODB)**: Последовательность некоторого числа битов, представляющих элементы синтаксиса, расположенные в полезной нагрузке последовательности ряда байтов (RBSP) перед стоповым битом. В строках SODB крайний левый бит считают первым и старшим значащим битом, а крайний правый бит – последним и младшим значащим битом.
- 3.145 **субмакроблок (sub-macroblock)**: Одна четвертая образца макроблока, т. е. 8x8 блока яркости и двух соответствующих блоков цветности, каждый угол которых расположен в углу макроблока.
- 3.146 **деление субмакроблока (sub-macroblock partition)**: Образцы блока яркости и двух соответствующих блоков цветности, которые появляются в результате деления субмакроблока для создания внешнего предсказания.
- 3.147 **I секция переключения (switching I slice)**: См. SI секция.
- 3.148 **P секция переключения (switching P slice)**: См. SP секция.
- 3.149 **элемент синтаксиса (syntax element)**: Элемент данных, представленных в потоке битов.
- 3.150 **структура синтаксиса (syntax structure)**: Ноль или более элементов синтаксиса, представленных совместно в определенном порядке в потоке битов.
- 3.151 **верхнее поле (top field)**: Одно из двух полей, которые составляют кадр. Каждый ряд верхнего поля пространственно располагается непосредственно над рядом нижнего поля.
- 3.152 **верхний макроблок (из пары макроблоков) (top macroblock (of a macroblock pair))**: Макроблок в паре макроблоков, который содержит образцы верхнего ряда образцов пары макроблоков. Для пары макроблоков поля верхний макроблок представляет образцы из области верхнего поля кадра, который расположен в пространственной области пары макроблоков. Для пары макроблоков кадра верхний макроблок представляет образцы кадра, который расположен в пространственной области пары макроблоков.
- 3.153 **коэффициент преобразования (transform coefficient)**: Скалярная величина в частотной области, которая связана с конкретным одно- или двумерным индексом частоты в части инверсного преобразования процесса декодирования.

- 3.154 степень коэффициента преобразования (transform coefficient level):** Целая величина, представляющая значение, связанное с конкретным двумерным *индексом частоты* в *процессе декодирования* перед *масштабированием* для вычисления значения *коэффициента преобразования*.
- 3.155 универсальный уникальный идентификатор (universal unique identifier, UUID):** Идентификатор, который является уникальным в пространстве всех универсальных уникальных идентификаторов.
- 3.156 неопределенный (не установленный точно) (unspecified):** Термин "неопределенный" используют в разделах, описывающих некоторые значения конкретного *элемента синтаксиса*, и указывают, что эти значения не определены в понятиях настоящей Рекомендации | Международного стандарта, а также не будут иметь определенного значения в будущем как составляющей части настоящей Рекомендации | Международного стандарта.
- 3.157 кодирование с переменной длиной (variable length coding, VLC):** Обратимая процедура энтропии кодирования, присваивающая укороченные строки битов *символам*, появление которых ожидается чаще, а удлиненные строки битов – *символам*, появление которых ожидается реже.
- 3.158 сканирование зигзагом (zig-zag scan):** Особая последовательность упорядочения *степеней коэффициентов преобразования* от (примерно) самой нижней пространственной частоты к самой верхней. Сканирование зигзагом используют для *степеней коэффициентов преобразования* в *кадрах макроблоков*.

4 Сокращения

Для целей настоящей Рекомендации | Международного стандарта используются следующие сокращения.

CABAC	Context-based Adaptive Binary Arithmetic Coding	Основанное на контексте адаптивное арифметическое бинарное кодирование
CAVLC	Context-based Adaptive Variable Length Coding	Основанная на контексте адаптивное кодирование с переменной длиной
CBR	Constant Bit Rate	Передача потока битов с постоянной скоростью
CPB	Coded Picture Buffer	Буфер кодированного изображения
DPB	Decoded Picture Buffer	Буфер декодированного изображения
DUT	Decoder under test	Тестируемый декодер
FIFO	First-In, First-Out	"Первым прибыл – первым обслужен"
HRD	Hypothetical Reference Decoder	Гипотетический контрольный декодер
HSS	Hypothetical Stream Scheduler	Гипотетический планировщик потоков
IDR	Instantaneous Decoding Refresh	Мгновенное декодирующее восстановление
LSB	Least Significant Bit	Младший значащий бит
MB	Macroblock	Макроблок
MBAFF	Macroblock-Adaptive Frame-Field Coding	Адаптивное декодирование макроблоками кадра/поля
MSB	Most Significant Bit	Старший значащий бит
NAL	Network Abstraction Layer	Уровень абстракции сети
RBSP	Raw Byte Sequence Payload	Полезная нагрузка последовательности исходных байтов
SEI	Supplemental Enhancement Information	Дополнительная расширенная информация
SODB	String Of Data Bits	Строка битов данных
UUID	Universal Unique Identifier	Универсальный уникальный идентификатор
VBR	Variable Bit Rate	Передача потока битов с переменной скоростью
VCL	Video Coding Layer	Уровень видеокодирования
VLC	Variable Length Coding	Кодирование с переменной длиной
VUI	Video Usability Information	Информация об используемости изображения

5 Условные обозначения

ПРИМЕЧАНИЕ. – Математические операторы, использованные в настоящей Спецификации, аналогичны тем, которые использованы в языке программирования "Си". Однако операции целочисленного деления и арифметического сдвига определены особо. Обозначения нумерации и счета обычно начинаются с 0.

5.1 Арифметические операторы

Нижеперечисленные арифметические операторы определены следующим образом.

+	Сложение.
-	Вычитание (как оператор двух аргументов) или отрицание (как оператор унарного префикса).
*	Умножение.
x^y	Степень. Определяет x в степени y . В других контекстах это обозначение используют для надписи, не предназначенной для интерпретации в качестве степени.
/	Целочисленное деление с усечением результата в меньшую сторону. Например, $7/4$ и $-7/-4$ усечены до 1, а $-7/4$ и $7/-4$ усечены до -1.
\div	Используют для обозначения деления в математических уравнениях, где не предполагают усечения или округления.
$\frac{x}{y}$	Используют для обозначения деления в математических уравнениях, где не предполагают усечения или округления.
$\sum_{i=x}^y f(i)$	Суммирование $f(i)$ с i , принимающим все целые значения от x до y включительно.
$x \% y$	Модуль. Остаток от деления x на y , определенный только для целых x и y при $x \geq 0$ и $y > 0$.

Если порядок старшинства не выражен явно с помощью скобок, применимы следующие правила:

- операции умножения и деления предшествуют сложению и вычитанию;
- операции умножения и деления в последовательности вычисляются в порядке слева направо;
- операции сложения и вычитания в последовательности вычисляются в порядке слева направо.

5.2 Логические операторы

Нижеперечисленные логические операторы определены следующим образом.

$x \&\& y$	Булево логическое "и" x и y .
$x \ \ y$	Булево логическое "или" x и y .
!	Булево логическое "не".
$x ? y : z$	Если x – ИСТИНА или не равно 0 вычисляется значение y . Иначе вычисление значения y заменяют вычислением z .

5.3 Операторы отношения

Нижеперечисленные операторы отношения определены следующим образом.

>	Больше.
>=	Больше или равно.
<	Меньше.
<=	Меньше или равно.
==	Равно.
!=	Не равно.

Если оператор отношения применяется к элементу синтаксиса или к переменной, которым присвоено значение "na" (не применяется), значение "na" для этого элемента синтаксиса или переменной обрабатывается как отдельное значение. Значение "na" рассматривается не равным любому другому значению.

5.4 Двоичные операторы

Нижеперечисленные двоичные операторы определены следующим образом.

&	Побитовое "и". При операциях с целочисленными аргументами выполняет действие с представлением этого целого значения в виде дополнения до двух. При операциях с двоичным аргументом, который содержит меньше битов, чем другой аргумент, к более короткому аргументу добавляются значащие биты, равные 0.
---	--

- | Побитовое "или". При операциях с целочисленными аргументами выполняет действие с представлением этого целого значения в виде дополнения до двух. При операциях с двоичным аргументом, который содержит меньше битов, чем другой аргумент, к более короткому аргументу добавляются значащие биты, равные 0.
- $x \gg y$ Арифметический сдвиг вправо представления целого числа x в виде дополнения до двух на y двоичных разрядов. Эта функция определена только для положительных целочисленных значений y . Бит, сдвинутый к битам MSB в результате сдвига вправо, имеет значение, равное MSB для x до операции сдвига.
- $x \ll y$ Арифметический сдвиг влево представления целого числа x в виде дополнения до двух на y двоичных разрядов. Эта функция определена только для положительных целочисленных значений y . Бит, сдвинутый к битам LSB в результате сдвига влево, имеет значение, равное 0.

5.5 Операторы присвоения

Нижеперечисленные арифметические операторы определены следующим образом.

- = Оператор присвоения.
- ++ Приращение, т. е. $x++$ эквивалентно $x = x + 1$ при использовании в индексе массива (матрицы), вычисляются по значению переменной до операции приращения.
- Декремент, т. е. $x--$ эквивалентно $x = x - 1$ при использовании в индексе массива (матрицы), вычисляются по значению переменной до операции уменьшения.
- += Приращение на оговоренную величину, т. е. $x += 3$ эквивалентно $x = x + 3$, а $x += (-3)$ эквивалентно $x = x + (-3)$.
- = Декремент на оговоренную величину, т. е. $x -= 3$ эквивалентно $x = x - 3$, а $x -= (-3)$ эквивалентно $x = x - (-3)$.

5.6 Обозначение ранга

Для определения ранга значений использованы следующие обозначения:

$x = y .. z$ x принимает целые значения, начиная от y до z включительно, где x , y и z – целые числа.

5.7 Математические функции

Нижеперечисленные математические функции определены следующим образом.

$$\text{Abs}(x) = \begin{cases} x & ; x \geq 0, \\ -x & ; x < 0. \end{cases} \quad (5-1)$$

$$\text{Ceil}(x) \text{ — наименьшее целое, большее или равное } x. \quad (5-2)$$

$$\text{Clip}_{1Y}(x) = \text{Clip}_3(0, (1 \ll \text{BitDepth}_Y) - 1, x). \quad (5-3)$$

$$\text{Clip}_{1C}(x) = \text{Clip}_3(0, (1 \ll \text{BitDepth}_C) - 1, x). \quad (5-4)$$

$$\text{Clip}_3(x, y, z) = \begin{cases} x & ; z < x, \\ y & ; z > y, \\ z & ; \text{ иначе.} \end{cases} \quad (5-5)$$

$$\text{Floor}(x) \text{ — наибольшее целое, меньшее или равное } x. \quad (5-6)$$

$$\text{Инверсного RasterScan}(a, b, c, d, e) = \begin{cases} (a\%(d/b))*b; & e == 0, \\ (a/(d/b))*c; & e == 1. \end{cases} \quad (5-7)$$

$$\text{Log}_2(x) \text{ логарифм от } x \text{ по основанию } 2. \quad (5-8)$$

$$\text{Log}_{10}(x) \text{ логарифму от } x \text{ по основанию } 10. \quad (5-9)$$

$$\text{Median}(x, y, z) = x + y + z - \text{Min}(x, \text{Min}(y, z)) - \text{Max}(x, \text{Max}(y, z)). \quad (5-10)$$

$$\text{Min}(x, y) = \begin{cases} x & ; x \leq y, \\ y & ; x > y. \end{cases} \quad (5-11)$$

$$\text{Max}(x, y) = \begin{cases} x & ; x \geq y, \\ y & ; x < y. \end{cases} \quad (5-12)$$

$$\text{Round}(x) = \text{Sign}(x) * \text{Floor}(\text{Abs}(x) + 0,5). \quad (5-13)$$

$$\text{Sign}(x) = \begin{cases} 1 & ; x \geq 0, \\ -1 & ; x < 0. \end{cases} \quad (5-14)$$

$$\text{Sqrt}(x) = \sqrt{x}. \quad (5-15)$$

5.8 Переменные, элементы синтаксиса и таблицы

Элементы синтаксиса в потоке битов представлены **жирным** шрифтом. Каждый элемент синтаксиса описан по его названию (все строчные буквы с подчеркнутыми знаками), по одной или двум категориям синтаксиса и одному или двум дескрипторам метода кодирования представления. Процесс декодирования протекает соответственно значению элемента синтаксиса и значениям предыдущих декодированных элементов синтаксиса. Если значение элемента синтаксиса использовано в таблицах синтаксиса или в тексте, его тип считают стандартным (т. е. не выделяют жирным шрифтом).

В некоторых случаях в таблицах синтаксиса могут использоваться значения других переменных, полученных из значений элементов синтаксиса. Такие переменные, которые появляются в таблицах синтаксиса или в тексте, обозначены комбинированными строчными и заглавными буквами без подчеркнутых знаков. Переменные, которые начинаются со строчных букв, извлекают для декодирования текущей структуры синтаксиса и всех зависимых структур синтаксиса. Переменные, которые начинаются с заглавных букв, могут быть использованы в процессе декодирования для последующих структур синтаксиса, упоминающих первоначальную структуру синтаксиса этой переменной. Переменные, которые начинаются со строчных букв, используют только в том подразделе, в котором они введены.

В некоторых случаях "мнемонические" названия значений элементов синтаксиса или значений переменных используют поочередно с их численными значениями. Иногда "мнемонические" названия используют без всякой ассоциации с численными значениями. Ассоциация значений и названий определена в тексте. Названия создают из одной или более групп букв, разделенных знаком подчеркивания. Каждая группа начинается с заглавной буквы и может содержать много заглавных букв.

ПРИМЕЧАНИЕ. – Синтаксис описан способом, который близок синтаксическим структурам языка программирования Си.

Функции описывают по их названиям, которые создают в виде названий элементов синтаксиса, заключенных в круглые скобки, включая нуль и другие названия переменных (для определения) или значений (для использования), разделенных запятыми (если имеется более одной переменной).

Одномерный массив называется списком. Двумерный массив называется матрицей. Массивы могут быть либо элементами синтаксиса, либо переменными. Для индексации массивов используются индексы или квадратные скобки. При визуальном представлении матрицы первый индекс используется для обозначения ряда (вертикали), а второй индекс – для обозначения столбца (горизонтали). Порядок индексирования изменяется на противоположный, если для индексации вместо индексов применяются квадратные скобки. Таким образом, элемент матрицы s , занимающий по горизонтали позицию x , а по вертикали – позицию y , может быть обозначен как $s[x,y]$ или как s_{yx} .

Двоичное обозначение указывают, заключая строку значений битов в однократные кавычки. Например, '01000001' представляет строку из восьми битов, в которой только второй и последний биты равны 1.

Шестнадцатеричная запись, отмеченная префиксом с шестнадцатеричным числом "0x", может быть использована вместо двоичной записи, если число битов четное и кратно 4. Например, 0x41 представляет строку из восьми битов, в которой только второй и последний биты равны 1.

Численные значения, не заключенные в однократные кавычки и без префикса "0x", являются десятичными значениями.

Значение, равное 0, в сообщении тестирования представляет ЛОЖЬ. Значение ИСТИНА представлено любыми другими отличными от нуля значениями.

5.9 Описание логических операций

В этом тексте логические операции будут описаны как бы в псевдокоде следующим образом:

```
если ( условие 0 )
    утверждение 0
если еще (условие 1 )
    утверждение 1
...
еще /* информативная ремарка остающегося условия */
    утверждение n,
```

может быть описано следующим образом:

```
... как следует / ... применимо следующее:
– если условие 0, утверждение 0
– иначе, если условие 1, утверждение 1
– ...
– иначе (информативная ремарка остающегося условия) утверждение n.
```

Каждое утверждение "Если... Иначе, Если... Иначе..." в этом тексте вводят с помощью "... как следует" или "... применимо следующее", которые непосредственно следуют за "Если...". Последнее условие "Если... Иначе, Если... Иначе, ..." всегда является "Иначе...". Чередующиеся утверждения "Если... Иначе, Если... Иначе..." можно описать подходящим "... как следует" или "... применимо следующее" с окончанием "Иначе...".

В этом тексте логические операции будут описаны как бы в псевдокоде следующим образом:

```
если ( условие 0a && условие 0b )
    утверждение 0
если еще (условие 1a || условие 1b )
    утверждение 1
...
еще
    утверждение n,
```

может быть описано следующим образом:

```
... как следует / ... применимо следующее:
– если все следующие условия истинны, то утверждение 0
  – условие 0a
  – условие 0b
– иначе, если любое из следующих условий истинно, то утверждение 1
  – условие 1a
  – условие 1b
– ...
– иначе утверждение n.
```

В этом тексте логические операции будут описаны как бы в псевдокоде следующим образом:

если (условие 0)
 утверждение 0
если (условие 1)
 утверждение 1,

может быть описано следующим образом:

 когда условие 0, утверждение 0,
 когда условие 1, утверждение 1.

5.10 Процессы

Процессы используют для описания декодирования элементов синтаксиса. Процесс имеет отдельную спецификацию и осуществление. Все элементы синтаксиса и заглавных переменных, которые принадлежат текущей структуре синтаксиса и зависимых структур синтаксиса, доступны в процессе спецификации и осуществления. Процесс спецификации может также иметь строчные переменные, точно описанные на входе. Каждый процесс спецификации имеет однозначно описанный выход. Выход – это переменная, которая может быть либо заглавной, либо строчной переменной.

Назначение переменных описывают следующим образом.

- В процессе осуществления переменным однозначно присваивают строчные знаки переменных на входе и выходе спецификации процесса в случае, если эти переменные не имеют одних и тех же названий.
- Иначе (если переменные при осуществлении и спецификации имеют одни и те же названия) вводят присвоение.

При спецификации процесса конкретный макроблок может рассматриваться по названию переменной, имеющей значение, равное адресу этого конкретного макроблока.

6 Форматы исходных, кодированных, декодированных и выходных данных, процессы сканирования и взаимоотношения смежности

6.1 Форматы потоков битов

В этом подразделе описаны взаимоотношения между потоком блоков NAL и потоком байтов, причем каждый из них рассматривают как поток битов.

Поток битов может иметь один из двух форматов: формат потока блоков NAL или формат потока байтов. Формат потока блоков NAL концептуально является более "основным" типом. Он состоит из последовательности структур синтаксиса, которые называются блоками NAL. Эта последовательность расположена в порядке декодирования. Для потока блоков NAL на порядок декодирования (и содержание) блоков NAL введен ряд ограничений.

Формат потока байтов может быть создан из формата потока блоков NAL расстановкой блоков NAL в порядке декодирования и заданием префиксов каждого блока NAL с помощью префикса кода старта, нуля и дополнительных байтов с нулевым значением для формирования потока байтов. Формат потока блоков NAL можно извлечь из формата потока байтов отысканием расположения особого образца префикса кода старта среди этого потока байтов. Методы кадрирования блоков NAL способом, отличным от того, который используют для формата потока байтов, выходит за рамки настоящей Рекомендации | Международного стандарта. Формат потока байтов описан в Приложении В.

6.2 Форматы исходного, декодированного и выходного изображений

В этом подразделе описаны взаимосвязи между источником и декодированными кадрами и полями, которые рассмотрены с помощью потока битов.

Источник видеосигнала, который представлен потоком битов, является последовательностью как кадров, так и полей (собираательно называемых изображениями) в порядке их декодирования.

Каждый источник и декодированные изображения (кадры или поля) состоят из одного или более массивов образцов.

- Только яркость (Y) (монохромное изображение) с или без дополнительного массива.
- Яркость и два цвета (YCbCr или YCgCo) с или без дополнительного массива.
- Зеленый, синий и красный (GBR, также называемый RGB) с или без дополнительного массива.
- Массивы, представляющие другие неспецифицированные монохромные или трехцветную (например, YZX, также называемый XYZ) выборки цветов с или без дополнительного массива.

Для удобства обозначения и терминологии в настоящей спецификации переменные и термины, связанные с этими массивами, называются яркостью (или L либо Y) и цветностью, где два массива цветности обозначаются как Cb и Cr, независимо от фактически используемого метода представления цвета. Фактически используемый метод представления цвета может быть указан в синтаксисе, описание которого содержится в Приложении E. Дополнительные (монохромное изображение) массивы, которые могут быть или могут не быть представлены как дополнительные изображения в кодированной видеопоследовательности, являются необязательными для декодирования и могут использоваться для таких целей как альфа-смешение.

Переменные SubWidthC и SubHeightC определены в таблице 6-1 в зависимости от структуры выборки формата цветности, которая определяется через chroma_format_idc. Запись, помеченная как "-" в таблице 6-1, обозначает неопределенное значение SubWidthC или SubHeightC. В будущем МСЭ-Т | ИСО/МЭК могут быть определены другие значения chroma_format_idc, SubWidthC и SubHeightC.

Таблица 6-1 – Значения SubWidthC и SubHeightC, полученное из chroma_format_idc

chroma_format_idc	Формат цветности	SubWidthC	SubHeightC
0	Монохромное	–	–
1	4:2:0	2	2
2	4:2:2	2	1
3	4:4:4	1	1

При монохромной выборке имеется только один массив образцов, который номинально рассматривается как массив яркости.

При выборке 4:2:0 каждый из двух массивов цветности занимает половину высоты и половину ширины массива яркости.

При выборке 4:2:2 каждый из двух массивов цветности занимает ту же высоту и половину ширины массива яркости.

При выборке 4:4:4 каждый из двух массивов цветности занимает ту же высоту и ширину, что и массив яркости.

Значение ширины и высоты массивов образцов яркости, каждое, является целым числом, кратным 16. В потоках битов при использовании цветовой выборки 4:2:0 значение ширины и высоты массивов образцов цветности, каждое, является целым числом, кратным 8. В потоках битов при использовании цветовой выборки 4:2:2 ширина массивов образцов цветности является целым числом, кратным 8, а высота – целым числом, кратным 16. Высота массива яркости, закодированного как два отдельных поля или с использованием адаптивного кодирования макроблоками кадра/поля (см. ниже), является целым числом, кратным 32. В потоках битов при использовании цветовой выборки 4:2:0 высота каждого массива цветности, закодированного как два отдельных поля или с использованием адаптивного кодирования макроблоками кадра/поля (см. ниже), является целым числом, кратным 16. Ширина или высота изображений выхода процесса декодирования необязательно должна быть целым числом, кратным 16 и может быть описана с использованием кадрирующего прямоугольника.

Синтаксис для массивов яркости и (если имеются) цветности упорядочен таким образом, что если имеются данные для всех трех компонентов цвета, данные для массива яркости помещаются первыми, за ними следуют любые данные для массива Cb и за ними – любые данные для массива Cr, если не определен иной порядок.

Ширина полей, кодированных по специальной установке параметров этой последовательности, будет такой же, как и кадров, кодированных по той же установке параметров этой последовательности (см. ниже). Высота полей, кодированных по специальной установке параметров этой последовательности составляет половину от высоты кадров, кодированных по той же установке параметров этой последовательности (см. ниже).

Количество битов, необходимое для представления каждого из образцов в массивах яркости и цветности в видеопоследовательности, находится в диапазоне от 8 до 12, и количество битов, используемых в массиве яркости, может отличаться от количества битов, используемых в массивах цветности.

Если значение chroma_format_idc равно 1, номинальное вертикальное и горизонтальное относительное расположение образцов яркости и цветности в кадрах показано на рисунке 6-1. Альтернативные относительные расположения образцов цветности можно найти в информации об используемости изображения (см. Приложение E).

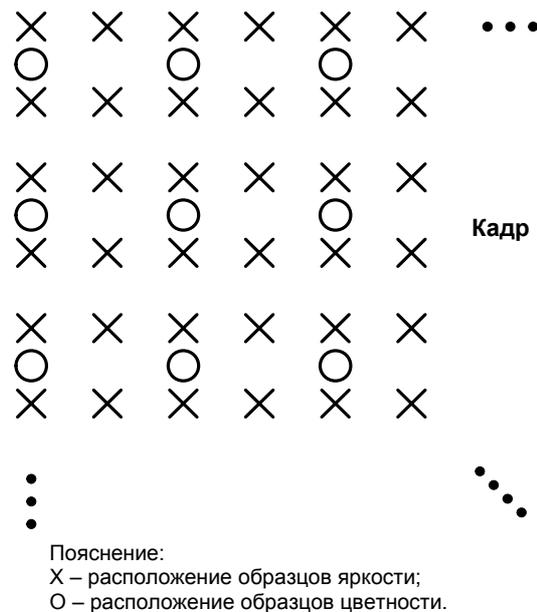


Рисунок 6-1 – Номинальное вертикальное и горизонтальное расположение образцов яркости и цветности 4:2:0 в кадре

Кадр состоит из двух полей, как описано ниже. Кодированное изображение может представлять кодированный кадр или отдельное кодированное поле. Кодированная видеопоследовательность, соответствующая настоящей Рекомендации | Международному стандарту, может содержать произвольные комбинации кодированных кадров и кодированных полей. Процесс декодирования также описан способом, который позволяет кодировать малые области кодированного кадра как область либо кадра, либо поля, используя адаптивное кодирование макроблоками кадра/поля.

Источник и декодированные поля бывают двух типов: верхнее поле и нижнее поле. Если два поля появляются на выходе в одно и то же время или их комбинируют, чтобы использовать как контрольный кадр (см. ниже), эти два поля (каждое должно быть типа инверсного равенства) чередуют. Первый (т. е. верхний), третий, пятый и т. д. ряды декодированных кадров являются верхними рядами полей. Второй, четвертый, шестой и т. д. ряды декодированного кадра являются нижними рядами полей. Верхнее поле состоит только из верхних рядов полей декодированного кадра. Если верхнее или нижнее поле декодированного кадра используют как контрольное поле (см. ниже), то используют только четные ряды (для верхнего поля) или нечетные ряды (для нижнего поля) декодированного кадра.

Если значение `chroma_format_idc` равно 1, номинальное вертикальное и горизонтальное относительное расположение образцов яркости и цветности в верхних и нижних полях показано на рисунке 6-2. Номинальное вертикальное относительное расположение выборки образцов цветности в верхнем поле определяется как сдвинутое на одну четверть высоты образца яркости относительно сетки выборки полей. Вертикальное расположение выборки образцов цветности в нижнем поле описано как сдвинутое вниз на одну четверть высоты образца яркости относительно сетки выборки полей. Альтернативные относительные расположения образцов цветности могут быть указаны в информации об используемости изображения (см. Приложение E).

ПРИМЕЧАНИЕ. – Сдвиг образцов цветности происходит в порядке выравнивания этих образцов вертикально к обычному расположению относительно сетки выборки полного кадра, как показано на рисунке 6-1.

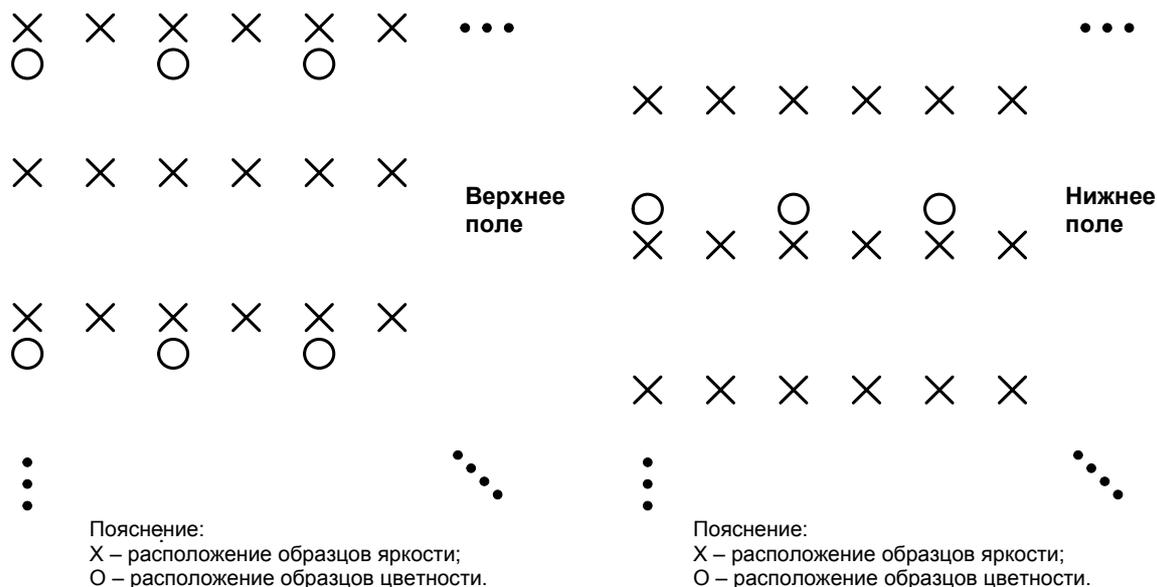


Рисунок 6-2 – Номинальное вертикальное и горизонтальное расположение образцов 4:2:0 выборки в верхних и нижних полях

Если значение `chroma_format_idc` равно 2, образцы цветности совмещаются с соответствующими образцами яркости, а номинальное расположение в кадре и в полях показано на рисунке 6-3 – Номинальное вертикальное и горизонтальное расположение образцов 4:2:2 яркости и цветности в кадре, и на рисунке 6-4 – Номинальное вертикальное и горизонтальное расположение выборки образцов 4:2:2 в верхнем и нижнем полях, соответственно.

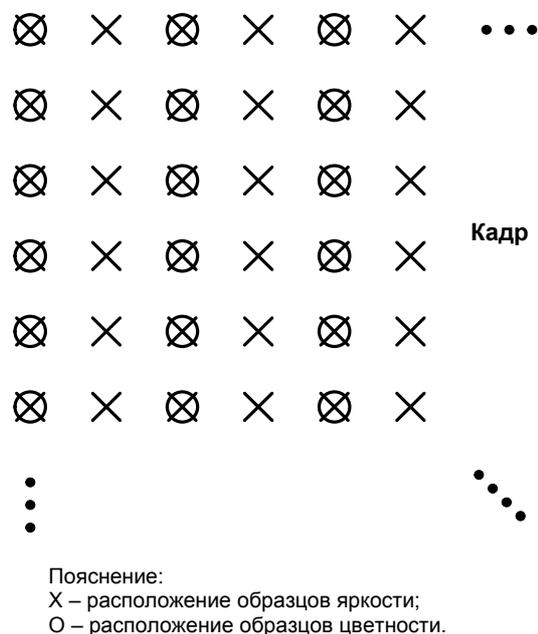


Рисунок 6-3 – Номинальное вертикальное и горизонтальное расположение образцов 4:2:2 яркости и цветности в кадре

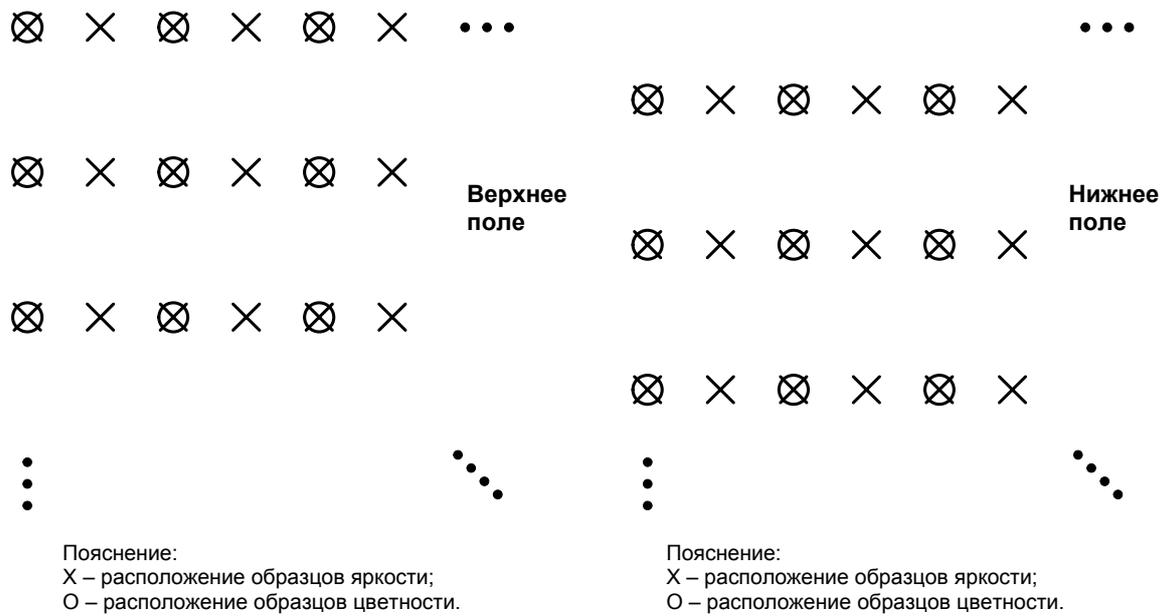


Рисунок 6-4 – Номинальное вертикальное и горизонтальное расположение образцов 4:2:2 выборки в верхнем и нижнем полях

Если значение `chroma_format_idc` равно 3, все образцы массивов совмещаются для всех случаев кадров и полей, а номинальное расположение в кадре и в полях показано на рисунках 6-5 и 6-6, соответственно.

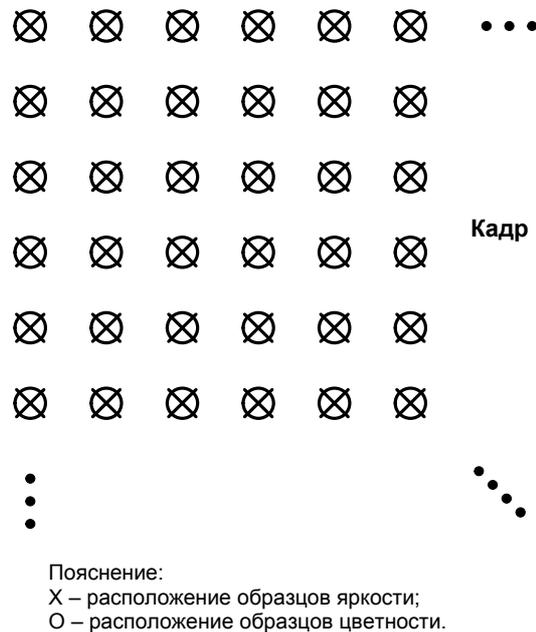


Рисунок 6-5 – Номинальное вертикальное и горизонтальное расположение образцов 4:4:4 яркости и цветности в кадре

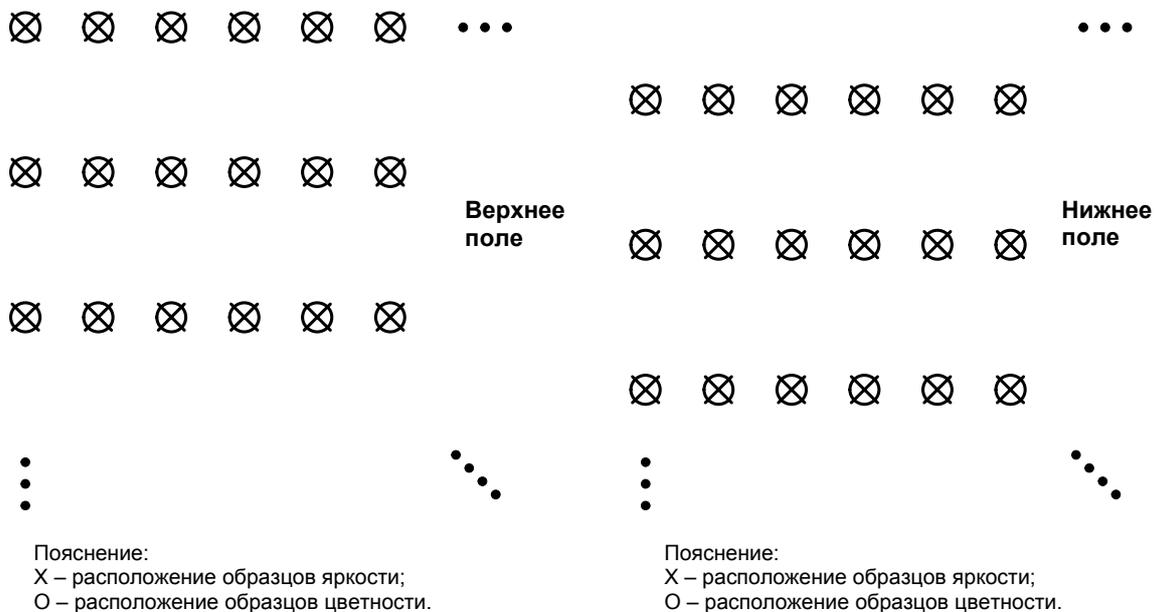


Рисунок 6-6 – Номинальное вертикальное и горизонтальное расположение образцов 4:4:4 выборки в верхнем и нижнем полях

Образцы обрабатываются в единицах макроблоков. Массив яркости для каждого макроблока составляют 16 образцов как по ширине, так и по высоте. Переменные MbWidthC и MbHeightC, которые определяют ширину и высоту, соответственно, массивов цветности для каждого макроблока, находят следующим образом.

- Если chroma_format_idc равно 0 (монохромное изображение), MbWidthC и MbHeightC, каждое, равно 0 (поскольку для монохромного изображения не определяется ни одного массива цветности).
- Иначе MbWidthC и MbHeightC определяют следующим образом:

$$\text{MbWidthC} = 16 / \text{SubWidthC}; \quad (6-1)$$

$$\text{MbHeightC} = 16 / \text{SubHeightC}. \quad (6-2)$$

6.3 Пространственное разделение на части изображений и секций

В этом подразделе описано, как изображение разделяют на секции и макроблоки. Изображение делят на секции. Секция – это последовательность макроблоков или, при использовании адаптивного декодирования макроблоками кадра/поля, это последовательность пары макроблоков.

Каждый макроблок включает один массив яркости 16x16 и, если формат изображения не является монохромным, два соответствующих массива образцов цветности. Если адаптивное декодирование макроблоками кадра/поля не используют, каждый макроблок представляет пространственную прямоугольную область изображения. Например, изображение может быть разделено на две секции, как показано на рисунке 6-7.

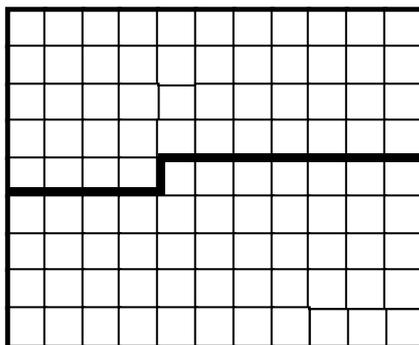


Рисунок 6-7 – Изображение с 11 на 9 макроблоков, которые разделены на две секции

Если используют адаптивное декодирование макроблоками кадра/поля, изображение разделяют на секции, содержащие целое число пар макроблоков, как показано на рисунке 6-8. Каждая пара макроблоков состоит из двух макроблоков.

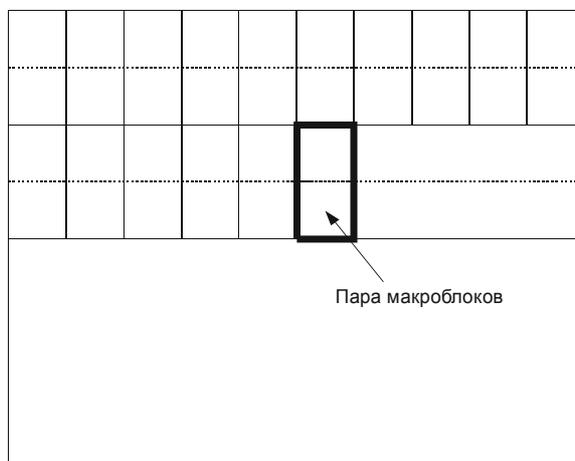


Рисунок 6-8 – Разделение декодированного кадра на пару макроблоков

6.4 Процессы инверсного сканирования и процессы образования смежных частей

В этом подразделе описаны процессы инверсного сканирования, т. е. размещение индексов на их места и процессы ответвления для смежных блоков.

6.4.1 Процесс инверсного сканирования макроблока

Вход этого процесса – адрес макроблока `mbAddr`.

Выход этого процесса – местоположение (x, y) левого верхнего образца яркости макроблока с адресом `mbAddr` относительно левого верхнего образца изображения.

Процесс инверсного сканирования макроблока описан следующим образом.

– Если MbaffFrameFlag равен 0:

$$x = \text{InverseRasterScan}(\text{mbAddr}, 16, 16, \text{PicWidthInSamples}_L, 0); \quad (6-3)$$

$$y = \text{InverseRasterScan}(\text{mbAddr}, 16, 16, \text{PicWidthInSamples}_L, 1). \quad (6-4)$$

– Иначе (MbaffFrameFlag равно 1) применяют следующее:

$$xO = \text{InverseRasterScan}(\text{mbAddr} / 2, 16, 32, \text{PicWidthInSamples}_L, 0); \quad (6-5)$$

$$yO = \text{InverseRasterScan}(\text{mbAddr} / 2, 16, 32, \text{PicWidthInSamples}_L, 1). \quad (6-6)$$

В зависимости от типа текущего макроблока применяют следующее.

– Если текущий макроблок – макроблок кадра:

$$x = xO; \quad (6-7)$$

$$y = yO + (\text{mbAddr} \% 2) * 16. \quad (6-8)$$

– Иначе (текущий макроблок – макроблок поля):

$$x = xO; \quad (6-9)$$

$$y = yO + (\text{mbAddr} \% 2). \quad (6-10)$$

6.4.2 Процесс инверсного сканирования при разделении макроблока и субмакроблока

Макроблоки или субмакроблоки могут быть разделены на части, а части сканированы для внешнего предсказания, как показано на рисунке 6-9. Внешние прямоугольники относят соответственно к образцам макроблока или субмакроблока. Прямоугольники относят к частям. Число в каждом прямоугольнике определяет индекс инверсно сканированной части макроблока или субмакроблока.

Функции MbPartWidth(), MbPartHeight(), SubMbPartWidth() и SubMbPartHeight(), описывающие ширину и высоту разделенной части макроблока и субмакроблока, указаны в таблицах 7-13, 7-14, 7-17 и 7-18. Функции MbPartWidth() и MbPartHeight() устанавливают соответствующие значения для каждого макроблока в зависимости от его типа. Функции SubMbPartWidth() и SubMbPartHeight() устанавливают соответствующие значения для каждого субмакроблока макроблока с типом, равным P_8x8, P_8x8ref0 или B_8x8 в зависимости от типа субмакроблока.



Рисунок 6-9 – Разделенные части макроблоков и субмакроблоков. Сканированные разделенные части макроблоков и субмакроблоков

6.4.2.1 Процесс инверсного сканирования при разделении макроблока

Вход этого процесса – индекс разделения макроблока mbPartIdx.

Выход этого процесса – местоположение (x , y) левого верхнего образца яркости разделенного макроблока mbPartIdx относительно левого верхнего образца этого макроблока.

Процесс инверсного сканирования макроблока описывают следующим образом:

$$x = \text{InverseRasterScan}(\text{mbPartIdx}, \text{MbPartWidth}(\text{mb_type}), \text{MbPartHeight}(\text{mb_type}), 16, 0); \quad (6-11)$$

$$y = \text{InverseRasterScan}(\text{mbPartIdx}, \text{MbPartWidth}(\text{mb_type}), \text{MbPartHeight}(\text{mb_type}), 16, 1). \quad (6-12)$$

6.4.2.2 Процесс инверсного сканирования при разделении субмакроблока

Входы этого процесса – индексы разделенной части макроблока mbPartIdx и индекс разделенной части субмакроблока.

Выход этого процесса – местоположение (x , y) левого верхнего образца яркости разделенного субмакроблока subMbPartIdx относительно левого верхнего образца этого субмакроблока.

Процесс инверсного сканирования субмакроблока описывают следующим образом.

– Если mb_type равен P_8x8, P_8x8ref0 или B_8x8:

$$x = \text{InverseRasterScan}(\text{subMbPartIdx}, \text{SubMbPartWidth}(\text{sub_mb_type}[\text{mbPartIdx}]), \text{SubMbPartHeight}(\text{sub_mb_type}[\text{mbPartIdx}]), 8, 0); \quad (6-13)$$

$$y = \text{InverseRasterScan}(\text{subMbPartIdx}, \text{SubMbPartWidth}(\text{sub_mb_type}[\text{mbPartIdx}]), \text{SubMbPartHeight}(\text{sub_mb_type}[\text{mbPartIdx}]), 8, 1). \quad (6-14)$$

– Иначе:

$$x = \text{InverseRasterScan}(\text{subMbPartIdx}, 4, 4, 8, 0); \quad (6-15)$$

$$y = \text{InverseRasterScan}(\text{subMbPartIdx}, 4, 4, 8, 1). \quad (6-16)$$

6.4.3 Процесс инверсного сканирования блока яркости 4x4

Входы этого процесса – индексы блока яркости 4x4 $luma4x4BlkIdx$.

Выход этого процесса – местоположение (x, y) левого верхнего образца блока яркости 4x4 с индексом яркости $4x4BlkIdx$ относительно левого верхнего образца этого макроблока.

На рисунке 6-10 показано сканирование для блоков яркости 4x4.

0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

Рисунок 6-10 – Сканирование блоков яркости 4x4

Процесс инверсного сканирования блока яркости 4x4 описывают следующим образом:

$$x = \text{InverseRasterScan}(luma4x4BlkIdx / 4, 8, 8, 16, 0) + \text{InverseRasterScan}(luma4x4BlkIdx \% 4, 4, 4, 8, 0); \quad (6-15)$$

$$y = \text{InverseRasterScan}(luma4x4BlkIdx / 4, 8, 8, 16, 1) + \text{InverseRasterScan}(luma4x4BlkIdx \% 4, 4, 4, 8, 1). \quad (6-16)$$

6.4.4 Процесс инверсного сканирования блока яркости 8x8

Входы этого процесса – индексы блока яркости 8x8 $luma8x8BlkIdx$.

Выход этого процесса – местоположение (x, y) левого верхнего образца яркости блока яркости 8x8 с индексом $luma8x8BlkIdx$ относительно левого верхнего образца яркости этого макроблока.

На рисунке 6-11 показано сканирование блоков яркости 8x8.

0	1
2	3

Рисунок 6-11 – Сканирование блоков яркости 8x8

Процесс инверсного сканирования блока яркости 8x8 описывают следующим образом:

$$x = \text{InverseRasterScan}(luma8x8BlkIdx, 8, 8, 16, 0); \quad (6-19)$$

$$y = \text{InverseRasterScan}(luma8x8BlkIdx, 8, 8, 16, 1). \quad (6-20)$$

6.4.5 Процесс создания доступного адреса макроблока

Вход этого процесса – индексы адреса макроблока $mbAddr$.

Выход этого процесса – доступный адрес макроблока $mbAddr$.

ПРИМЕЧАНИЕ. – Процесс доступности определяется при реализации.

Макроблок отмечают как доступный, если одно из следующих условий не является истиной, и тогда макроблок отмечают как недоступный:

- $mbAddr < 0$;
- $mbAddr > CurrMbAddr$;
- макроблок с адресом $mbAddr$ принадлежит к другой секции, чем макроблок с адресом $CurrMbAddr$.

6.4.6 Процесс создания адреса смежного макроблока и его доступность

Этот процесс может быть реализован при $MbaffFrameFlag$, равном 0.

Выходы этого процесса:

- $mbAddrA$: адрес и статус доступности макроблока слева от текущего макроблока;
- $mbAddrB$: адрес и статус доступности макроблока выше от текущего макроблока;
- $mbAddrC$: адрес и статус доступности макроблока справа от текущего макроблока;
- $mbAddrD$: адрес и статус доступности макроблока над левым текущим макроблоком.

На рисунке 6-12 показано относительные пространственные расположения макроблоков с адресами $mbAddrA$, $mbAddrB$, $mbAddrC$ и $mbAddrD$ относительно текущего макроблока с адресом $CurrMbAddr$.

$mbAddrD$	$mbAddrB$	$mbAddrC$
$mbAddrA$	$CurrMbAddr$	

Рисунок 6-12 – Смежные макроблоки по отношению к заданному макроблоку

Вход процесса согласно п. 6.4.5 – это $mbAddrA = CurrMbAddr - 1$, а выход (в случае доступности) – макроблок $mbAddrA$. Кроме того, $mbAddrA$ отмечают как недоступный, если $CurrMbAddr \% PicWidthInMbs$ равно 0.

Вход процесса согласно п. 6.4.5 – это $mbAddrB = CurrMbAddr - PicWidthInMbs$, а выход (в случае доступности) – макроблок $mbAddrB$.

Вход процесса согласно п. 6.4.5 – это $mbAddrC = CurrMbAddr - PicWidthInMbs + 1$, а выход (в случае доступности) – макроблок $mbAddrC$. Кроме того, $mbAddrC$ отмечают как недоступный, если $(CurrMbAddr + 1) \% PicWidthInMbs$ равно 0.

Вход процесса согласно п. 6.4.5 – это $mbAddrD = CurrMbAddr - PicWidthInMbs - 1$, а выход (в случае доступности) – макроблок $mbAddrD$. Кроме того, $mbAddrD$ отмечают как недоступный, если $CurrMbAddr \% PicWidthInMbs$ равно 0.

6.4.7 Процесс создания адреса смежного макроблока и его доступность в кадрах MBAFF

Этот процесс может быть реализован при $MbaffFrameFlag$, равном 1.

Выходы этого процесса:

- $mbAddrA$: адрес и статус доступности верхнего макроблока из пары макроблоков слева от текущей пары макроблоков;
- $mbAddrB$: адрес и статус доступности верхнего макроблока из пары макроблоков над текущей парой макроблоков;

- mbAddrC: адрес и статус доступности верхнего макроблока из пары макроблоков сверху справа от текущей пары макроблоков;
- mbAddrD: адрес и статус доступности верхнего макроблока из пары макроблоков сверху слева от текущей пары макроблоков.

На рисунке 6-13 показано относительное пространственное расположение макроблоков с адресами mbAddrA, mbAddrB, mbAddrC и mbAddrD относительно текущего макроблока с адресом CurrMbAddr.

Адреса mbAddrA, mbAddrB, mbAddrC и mbAddrD имеют идентичные значения независимо от того, находится ли текущий макроблок наверху, внизу или в паре макроблоков.

mbAddrD	mbAddrB	MbAddrC
mbAddrA	CurrMbAddr или	
	CurrMbAddr	

Рисунок 6-13 – Смежные макроблоки по отношению к заданному макроблоку в кадрах MBAFF

Вход процесса согласно п. 6.4.5 – это $mbAddrA = 2 * (CurrMbAddr / 2 - 1)$, а выход (в случае доступности) – макроблок mbAddrA. Кроме того, mbAddrA отмечают как недоступный, если $(CurrMbAddr / 2) \% PicWidthInMbs$ равно 0.

Вход процесса согласно п. 6.4.5 – это $mbAddrB = 2 * (CurrMbAddr / 2 - PicWidthInMbs)$, а выход (в случае доступности) – макроблок mbAddrB.

Вход процесса согласно п. 6.4.5 – это $mbAddrC = 2 * (CurrMbAddr / 2 - PicWidthInMbs + 1)$ а выход (в случае доступности) – макроблок mbAddrC. Кроме того, mbAddrC отмечают как недоступный, если $(CurrMbAddr / 2 + 1) \% PicWidthInMbs$ равно 0.

Вход процесса согласно п. 6.4.5 – это $mbAddrD = 2 * (CurrMbAddr / 2 - PicWidthInMbs - 1)$, а выход (в случае доступности) – макроблок mbAddrD. Кроме того, mbAddrD отмечают как недоступный, если $(CurrMbAddr / 2) \% PicWidthInMbs$ равно 0.

6.4.8 Процессы создания смежных макроблоков, блоков и разделения на части

Пункт 6.4.8.1 определяет процесс создания смежных макроблоков.

Пункт 6.4.8.2 определяет процесс создания смежных блоков яркости 8x8.

Пункт 6.4.8.3 определяет процесс создания смежных блоков яркости 4x4.

Пункт 6.4.8.4 определяет процесс создания смежных блоков цветности 4x4.

Пункт 6.4.8.5 определяет процесс разделения на части смежных блоков.

В таблице 6-2 приведены значения разности расположения яркости (xD, yD) для входа и замены N в параметрах mbAddrN, mbPartIdxN, subMbPartIdxN, luma8x8BlkIdxN, luma4x4BlkIdxN и chroma4x4BlkIdxN на выходе. Эти присвоения входа и выхода взяты из пп. 6.4.8.1–6.4.8.5. Переменную predPartWidth описывают при ссылках на таблицу 6-2.

Таблица 6-2 – Спецификация присвоений входов и выходов в пп. 6.4.8.1–6.4.8.5

N	xD	yD
A	-1	0
B	0	-1
C	predPartWidth	-1
D	-1	-1

На рисунке 6-14 показано относительное расположение смежных макроблоков, блоков или разделение на части A, B, C и D текущего макроблока, частей или блоков, если текущий макроблок, часть или блок появляются в режиме кодирования кадра.



Рисунок 6-14 – Определение смежных макроблоков, блоков и разделенных частей (для информации)

6.4.8.1 Процесс создания смежных макроблоков

Выходы этого процесса:

- mbAddrA: адрес макроблока слева от текущего макроблока и статус его доступности; и
- mbAddrB: адрес макроблока над текущим макроблоком и статус его доступности;
- mbAddrN (при N, равном A или B) создают следующим образом;
- разность расположения яркости (xD, yD) устанавливают в соответствии с таблицей 6-2;
- процесс создания смежных местоположений, описанный в п. 6.4.9, реализуют для расположения яркости с (xN, yN), равным (xD, yD), а выводу приписывают mbAddrN.

6.4.8.2 Процесс создания смежных блоков яркости 8x8

Вход этого процесса – блок яркости 8x8 с индексом luma8x8BlkIdx.

Индекс luma8x8BlkIdx определяет блок яркости 8x8 макроблока при растровом сканировании.

Выходы этого процесса:

- mbAddrA: либо равен CurrMbAddr, либо адресу макроблока слева от текущего макроблока и статусу его доступности;
- luma8x8BlkIdxA: индекс блок яркости 8x8 слева от блока 8x8 с индексом luma8x8BlkIdx и статус его доступности;
- mbAddrB: либо равен CurrMbAddr, либо адресу макроблока над текущим макроблоком и статусу его доступности;

- luma8x8BlkIdxB: индекс блок яркости 8x8 над блоком 8x8 с индексом luma8x8BlkIdx и статус его доступности.

mbAddrN и luma8x8BlkIdxN (при N, равном A или B) создают следующим образом.

- Разность расположения яркости (xD, yD) устанавливаются в соответствии с таблицей 6-2.

- Расположение яркости (xN, yN) описывают следующим образом:

$$xN = (luma8x8BlkIdx \% 2) * 8 + xD; \quad (6-21)$$

$$yN = (luma8x8BlkIdx / 2) * 8 + yD. \quad (6-22)$$

- Процесс создания смежных блоков, как это описано в п. 6.4.9, реализуют при местоположении яркости в (xN, yN), поскольку входу и выходу присвоены параметры mbAddrN и (xW, yW).

- Переменную luma8x8BlkIdxN определяют следующим образом:

- если mbAddrN недоступно, то luma8x8BlkIdxN помечают как недоступное;
- иначе (mbAddrN доступно) блоку яркости 8x8 в макроблоке mbAddrN, перекрывающем местоположение яркости (xW, yW), присваивается параметр luma8x8BlkIdxN.

6.4.8.3 Процесс создания смежных блоков яркости 4x4

Вход этого процесса – блок яркости 4x4 с индексом luma4x4BlkIdx.

Выходы этого процесса:

- mbAddrA: либо равен CurrMbAddr, либо адресу макроблока слева от текущего макроблока и статусу его доступности;
- luma4x4BlkIdxA: индекс блок яркости 4x4 слева от блока 4x4 с индексом luma4x4BlkIdx и статус его доступности;
- mbAddrB: либо равен CurrMbAddr, либо адресу макроблока над текущим макроблоком и статусу его доступности;
- luma4x4BlkIdxB: индекс блок яркости 4x4 над блоком 4x4 с индексом luma4x4BlkIdx и статус его доступности.

mbAddrN и luma4x4BlkIdxN (при N равном A или B) создают следующим образом.

- Разность расположения яркости (xD, yD) устанавливаются в соответствии с таблицей 6-2.

- Процесс инверсного сканирования блока яркости 4x4, как это описано в п. 6.4.3, реализуют с яркостью 4x4BlkIdx на входе и координатами (x, y) на выходе.

- Местоположение яркости (xN, yN) описано следующим образом:

$$xN = x + xD; \quad (6-23)$$

$$yN = y + yD. \quad (6-24)$$

- Процесс создания смежных блоков, как это описано в п. 6.4.9, реализуют при местоположении яркости в (xN, yN), поскольку входу и выходу присвоены параметры mbAddrN и (xW, yW).

- Переменную luma4x4BlkIdxN создают следующим образом:

- если mbAddrN недоступно, то luma8x8BlkIdxN помечают как недоступное;
- иначе (mbAddrN доступно) блоку яркости 4x4 в макроблоке mbAddrN, покрывающим местоположение яркости (xW, yW), присваивается параметр luma4x4BlkIdxN.

6.4.8.4 Процесс создания смежных блоков цветности 4x4

Вход этого процесса – индекс chroma4x4BlkIdx блока цветности 4x4.

Выходы этого процесса:

- mbAddrA (либо равен CurrMbAddr, либо адресу макроблока слева от текущего макроблока) и статус его доступности;

- chroma4x4BlkIdxA (индекс блока цветности 4x4 слева от блока цветности 4x4 с индексом chroma4x4BlkIdx) и статус его доступности;
- mbAddrB (либо равный CurrMbAddr, либо адресу макроблока над текущим макроблоком) и статус его доступности;
- chroma4x4BlkIdxB (индекс блока цветности 4x4 над блоком цветности 4x4 с индексом chroma4x4BlkIdx) и статус его доступности.

mbAddrN и chroma4x4BlkIdxN (с N, равным A или B) определяют следующим образом.

- Разница местоположения (xD, yD) устанавливается согласно таблице 6-2.
- В зависимости от chroma_format_idc позиция (x, y) верхнего левого образца блока цветности 4x4 с индексом chroma4x4BlkIdx определяют следующим образом:
 - Если chroma_format_idc равно 1 или 2, применяется следующее:

$$x = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 0); \quad (6-25)$$

$$y = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 1). \quad (6-26)$$

- Иначе (chroma_format_idc равно 3) применяется следующее:

$$x = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 0) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 0); \quad (6-27)$$

$$y = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 1) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 1). \quad (6-28)$$

- Местоположение цветности (xN, yN) определяется:

$$xN = x + xD; \quad (6-29)$$

$$yN = y + yD. \quad (6-30)$$

- Процесс создания смежных местоположений, описание которого содержится в п. 6.4.9, активизируют для местоположений цветности с (xN, yN), поскольку входу и выходу присваивают mbAddrN и (xW, yW).
- Переменную chroma4x4BlkIdxN определяют следующим образом:
 - если mbAddrN не доступно, chroma4x4BlkIdxN помечают как недоступное;
 - иначе (mbAddrN доступно) блоку цветности 4x4 в макроблоке mbAddrN, охватывающем местоположение цветности (xW, yW), присваивают chroma4x4BlkIdxN.

6.4.8.5 Процесс создания смежных блоков при разделении

Входы этого процесса:

- индекс разделенного макроблока mbPartIdx;
- тип текущего субмакроблока currSubMbType;
- индекс разделенного субмакроблока subMbPartIdx.

Выходы этого процесса:

- mbAddrA\mbPartIdxA\subMbPartIdxA: определяющий разделенные макроблок или субмакроблок слева от текущего макроблока и статус доступности или разделенную часть субмакроблока CurrMbAddr\mbPartIdx\subMbPartIdx и статус доступности;

- $mbAddrB \setminus mbPartIdxB \setminus subMbPartIdxB$: определяющий разделенные макроблок или субмакроблок над текущим макроблоком и статус доступности или разделенную часть субмакроблока $CurrMbAddr \setminus mbPartIdx \setminus subMbPartIdx$ и статус доступности;
- $mbAddrC \setminus mbPartIdxC \setminus subMbPartIdxC$: определяющий разделенные макроблок или субмакроблок справа сверху над текущим макроблоком и статус доступности или разделенную часть субмакроблока $CurrMbAddr \setminus mbPartIdx \setminus subMbPartIdx$ и статус доступности;
- $mbAddrD \setminus mbPartIdxD \setminus subMbPartIdxD$: определяющий разделенные макроблок или субмакроблок слева сверху над текущим макроблоком и статус доступности или разделенную часть субмакроблока $CurrMbAddr \setminus mbPartIdx \setminus subMbPartIdx$ и статус доступности.

$mbAddrN$, $mbPartIdxN$ и $subMbPartIdx$ (при N , равном A , B , C или D) определяют следующим образом.

- Процесс инверсного сканирования разделенного макроблока, как описано в п. 6.4.2.1, реализован с $mbPartIdx$ на входе и (x, y) на выходе.
- Местоположение левого верхнего образца яркости внутри разделенного макроблока (xS, yS) определяют следующим образом:
 - если mb_type равно P_8x8 , $P_8x8ref0$ или B_8x8 , процесс инверсного сканирования разделенного субмакроблока, как описано в п. 6.4.2.2, реализуют с $subMbPartIdx$ на входе и (xS, yS) на выходе;
 - иначе (xS, yS) устанавливают на $(0, 0)$.
- Переменная $predPartWidth$ в таблице 6-2 описана следующим образом.
 - Если mb_type равно P_Skip , B_Skip или B_Direct_16x16 , то $predPartWidth = 16$.
 - Иначе, если mb_type равно B_8x8 , применяют следующее:
 - если $currSubMbType$ равно B_Direct_8x8 , то $predPartWidth = 16$;
 - ПРИМЕЧАНИЕ 1. – Если $currSubMbType$ равно B_Direct_8x8 , а $direct_spatial_mv_pred_flag$ равно 1, то предсказанный вектор движения является предсказанным вектором движения для всего макроблока.
 - иначе $predPartWidth = SubMbPartWidth(sub_mb_type[mbPartIdx])$.
 - Иначе, если mb_type равно P_8x8 или $P_8x8ref0$, $predPartWidth = SubMbPartWidth(sub_mb_type[mbPartIdx])$.
 - Иначе $predPartWidth = MbPartWidth(mb_type)$.
- Разность расположения яркости (xD, yD) устанавливают в соответствии с таблицей 6-2.
- Расположение смежных блоков яркости (xN, yN) описывают следующим образом:

$$xN = x + xS + xD; \quad (6-31)$$

$$yN = y + yS + yD. \quad (6-32)$$

- Процесс создания смежных местоположений, как это описано в п. 6.4.8, реализуют для местоположений яркости (xN, yN) на входе и присваивают выходу $mbAddrN$ и (xW, yW) .
- В зависимости от $mbAddrN$ применяют следующее.
 - Если $mbAddrN$ недоступно, то разделенные макроблок или субмакроблок $mbAddrN \setminus mbPartIdxN \setminus subMbPartIdxN$ помечают как недоступные.
 - Иначе ($mbAddrN$ доступно) применяют следующее:
 - разделенной части макроблока в макроблоке $mbAddrN$, перекрывающей местоположение яркости (xW, yW) , присваивается $mbPartIdxN$, а разделенной части субмакроблока внутри разделенной части макроблока $mbPartIdxN$, перекрывающей местоположение образца (xW, yW) в макроблоке $mbAddrN$, присваивается $subMbPartIdxN$;
 - если разделение, заданное $mbPartIdxN$ и $subMbPartIdxN$, еще не декодировано, разделенные части макроблока $mbPartIdxN$ и субмакроблока $subMbPartIdxN$ помечают как недоступные.

ПРИМЕЧАНИЕ 2. – Последнее условие существует, например, для случая, когда $mbPartIdx = 2$, $subMbPartIdx = 3$, $xD = 4$, $yD = -1$, т. е. когда запрашивают смежный блок C из последнего блока яркости $4x4$ третьего субмакроблока.

6.4.9 Процесс создания смежных местоположений

Вход этого процесса – местоположение яркости или цветности (xN , yN), выраженное относительно верхнего левого угла текущего макроблока.

Выходы этого процесса:

- $mbAddrN$: равно либо $CurrMbAddr$, либо адресу соседнего макроблока, который содержит (xN , yN), и статус его доступности;
- (xW , yW): местоположение (xN , yN), выраженное относительно левого верхнего угла макроблока $mbAddrN$ (в большей степени, чем относительно левого верхнего угла текущего макроблока).

Положим, что $maxW$ и $maxH$ – переменные, определяющие максимальные значения компонентов местоположения xN , xW и yN , yW , соответственно. Тогда $maxW$ и $maxH$ определяют следующим образом.

- Если этот процесс реализован для смежных местоположений яркости:

$$maxW = maxH = 16. \quad (6-33)$$

- Иначе (этот процесс реализован для смежных местоположений цветности):

$$maxW = MbWidthC; \quad (6-34)$$

$$maxH = MbHeightC. \quad (6-35)$$

В зависимости от переменной $MbaffFrameFlag$ смежные местоположения находят следующим образом:

- если $MbaffFrameFlag$ равен 0, применяют спецификацию для смежных местоположений в полях и кадрах (но не кадрах MBAFF), как описано в п. 6.4.9.1;
- иначе ($MbaffFrameFlag$ равен 1) применяют спецификацию для смежных местоположений в полях и кадрах MBAFF, как описано в п. 6.4.9.2.

6.4.9.1 Спецификация для смежных местоположений в полях и кадрах (но не кадрах MBAFF)

Спецификацию этого пункта применяют, когда $MbaffFrameFlag$ равен 0.

В п. 6.4.6 процесс создания адресов смежных макроблоков и их доступность реализуют с помощью $mbAddrA$, $mbAddrB$, $mbAddrC$ и $mbAddrD$ так же, как и статус доступности на выходе.

Таблица 6-3 определяет $mbAddrN$ в зависимости от (xN , yN).

Таблица 6-3 – Спецификация $mbAddrN$

xN	yN	$mbAddrN$
< 0	< 0	$mbAddrD$
< 0	$0 .. maxH - 1$	$mbAddrA$
$0 .. maxW - 1$	< 0	$mbAddrB$
$0 .. maxW - 1$	$0 .. maxH - 1$	$CurrMbAddr$
$> maxW - 1$	< 0	$mbAddrC$
$> maxW - 1$	$0 .. maxH - 1$	недоступно
	$> maxH - 1$	недоступно

Смежное местоположение (xW , yW) относительно левого верхнего угла макроблока $mbAddrN$ находят как:

$$xW = (xN + maxW) \% maxW; \quad (6-36)$$

$$yW = (yN + maxH) \% maxH. \quad (6-37)$$

6.4.9.2 Спецификация для смежных местоположений в полях и кадрах MBAFF

Спецификацию этого пункта применяют, когда MbaffFrameFlag равен 1.

В п. 6.4.7 процесс создания адресов смежных макроблоков и их доступность реализуют с помощью mbAddrA, mbAddrB, mbAddrC и mbAddrD так же, как и статус доступности на выходе.

Таблица 6-4 определяет адреса макроблоков mbAddrN и уМ двумя этапами.

1. Спецификация адрес макроблока mbAddrX в зависимости от (xN, yN) и следующих переменных.
 - Переменную currMbFrameFlag находят следующим образом:
 - если макроблок с адресом CurrMbAddr – макроблок кадра, то currMbFrameFlag устанавливают на 1;
 - иначе (макроблок с адресом CurrMbAddr – макроблок поля) currMbFrameFlag устанавливают на 0.
 - Переменную mbIsTopMbFlag находят следующим образом:
 - если макроблок с адресом CurrMbAddr – верхний макроблок (CurrMbAddr % 2 равно 0), то mbIsTopMbFlag устанавливают на 1;
 - иначе (макроблок с адресом CurrMbAddr – нижний макроблок, CurrMbAddr % 2 равно 1) mbIsTopMbFlag устанавливают на 0.
2. В зависимости от доступности mbAddrX применяют следующее.
 - Если mbAddrX не доступен, то mbAddrN помечают как недоступный.
 - Иначе (mbAddrX доступен) mbAddrN помечают как доступный, а в таблице 6-4 указывают mbAddrN и уМ, зависящие от (xN, yN), currMbFrameFlag, mbIsTopMbFlag и переменную mbAddrXFrameFlag, которую находят следующим образом:
 - если макроблок mbAddrX – макроблок кадра, то mbAddrXFrameFlag устанавливают на 1;
 - иначе (макроблок mbAddrX – макроблок поля) mbAddrXFrameFlag устанавливают на 0.

Значения вышеприведенных флагов в таблице 6-4 как на (неопределенные) обозначают, что значения соответствующих флагов не существенны для данного ряда таблицы.

Таблица 6-4 – Спецификация mbAddrN и yM

xN	yN	currMbFrameFlag	mbIsTopMbFlag	mbAddrX	mbAddrXFrameFlag	Дополнительное условие	mbAddrN	yM	
< 0	< 0	1	1	mbAddrD			mbAddrD + 1	yN	
			0	mbAddrA	1		mbAddrA	yN	
		0	1	mbAddrD	1		mbAddrD + 1	$(yN + \max H) \gg 1$	
			0	mbAddrD	0		mbAddrD	$2 * yN$	
< 0	$0 \dots \max H - 1$	1	1	mbAddrA	1		mbAddrA	yN	
					0	$yN \% 2 == 0$	mbAddrA	$yN \gg 1$	
					0	$yN \% 2 != 0$	mbAddrA + 1	$yN \gg 1$	
			0	mbAddrA	1		mbAddrA + 1	yN	
					0	$yN \% 2 == 0$	mbAddrA	$(yN + \max H) \gg 1$	
					0	$yN \% 2 != 0$	mbAddrA + 1	$(yN + \max H) \gg 1$	
		0	mbAddrA	1	1	$yN < (\max H / 2)$	mbAddrA	$yN \ll 1$	
					0	$yN \geq (\max H / 2)$	mbAddrA + 1	$(yN \ll 1) - \max H$	
				0	mbAddrA	1	$yN < (\max H / 2)$	mbAddrA	$(yN \ll 1) + 1$
						0	$yN \geq (\max H / 2)$	mbAddrA + 1	$(yN \ll 1) + 1 - \max H$
$0 \dots \max W - 1$	< 0	1	1	mbAddrB			mbAddrB + 1	yN	
			0	CurrMbAddr			CurrMbAddr - 1	yN	
		0	1	mbAddrB	1		mbAddrB + 1	$2 * yN$	
			0	mbAddrB	0		mbAddrB	yN	
$0 \dots \max W - 1$	$0 \dots \max H - 1$			CurrMbAddr			CurrMbAddr	yN	
> $\max W - 1$	< 0	1	1	mbAddrC			mbAddrC + 1	yN	
			0	недоступно			недоступно	па	
		0	1	mbAddrC	1		mbAddrC + 1	$2 * yN$	
			0	mbAddrC	0		mbAddrC	yN	
> $\max W - 1$	$0 \dots \max H - 1$			недоступно			недоступно	па	
	> $\max H - 1$			недоступно			недоступно	па	

Смежные местоположения яркости (xW, yW) относительно левого верхнего угла макроблока mbAddrN находят как:

$$xW = (xN + \max W) \% \max W; \quad (6-38)$$

$$yW = (yM + \max H) \% \max H. \quad (6-39)$$

7 Синтаксис и семантика

7.1 Метод описания синтаксиса в табличной форме

Таблицы синтаксиса описывают надмножество синтаксиса всех разрешенных потоков битов. Дополнительные ограничения на синтаксис могут быть описаны прямо или косвенно либо в других разделах.

ПРИМЕЧАНИЕ. – Реальный декодер должен реализовывать средства для идентификации входных точек в потоке битов и средства для идентификации и обработки несоответствующих потоков битов. Методы идентификации и обработки ошибок, а также других аналогичных ситуаций здесь не описываются.

В следующей ниже таблице перечислены примеры псевдокодов, использованных для описания синтаксиса. Если появляется элемент синтаксиса, он указывает, что элемент синтаксиса синтаксически выделен из потока битов, а указатель потока битов в процессе синтаксического выделения из потока битов продвигается на следующую позицию за пределами элемента синтаксиса.

	С	Дескриптор
/* Может быть сделано утверждение элемента синтаксиса с соответствующей категорией синтаксиса и дескрипторов, или может быть сделано представление, использованное для описания условий существования, типа и количества элементов синтаксиса (syntax_element) как в двух следующих примерах */		
syntax_element	3	ue(v)
Условное утверждение		
/* Группа утверждений, заключенных в фигурные скобки, представляет сложное утверждение и рассматривается функционально как простое утверждение. */		
{		
утверждение		
утверждение		
...		
}		
/* Структура "пока" описывает проверку, действительно ли это условие – истина, и если истина, описывает оценку утверждения (или сложного утверждения) повторно до тех пор, пока это условие не перестанет быть истиной */		
Пока (условие)		
утверждение		
/*"Действует ... пока" структура, описывающая оценку утверждения однократно, за которой следует проверка истинности, и если оно истинно, повторно описывает оценку до тех пор, пока это условие не перестанет быть истиной */		
действует		
утверждение		
while(условие)		
/* Структура "если ... еще" описывает проверку, действительно ли это условие – истина, и если истина, описывает оценку первичного утверждения, иначе описывает оценку альтернативного утверждения. "Еще" – часть структуры и последующее альтернативное утверждение опускают, если не требуется никакой оценки альтернативного утверждения */		
if(условие)		
предварительное утверждение		
еще		
альтернативное утверждение		
/* Структура "для" определяет оценку первоначального утверждения, за которым следует проверка условия, и если оно истинно, повторно описывает оценку, за которой следует последующее утверждение, до тех пор, пока это условие не перестанет быть истиной. */		
for(первоначальное утверждение; условие; последующее утверждение)		
первоначальное утверждение		

7.2 Описание синтаксиса функций, категорий и дескрипторов

Представленные здесь функции используют для синтаксического описания. Эти функции предполагают присутствие указателя потока битов с указанием положения следующего бита, который должен быть считан процессом декодирования из потока битов.

`byte_aligned()` описан следующим образом:

- если текущая позиция в потоке битов попадает на пограничный байт, т. е. следующий бит в потоке битов – это первый бит в байте, то возвратное значение `byte_aligned()` будет ИСТИНА;
- иначе возвратное значение `byte_aligned()` будет ЛОЖЬ.

Параметр `more_data_in_byte_stream()`, который используют только в структуре синтаксиса потока байтов блоков NAL, описанной в Приложении В, описывают следующим образом:

- если в потоке байтов имеются дополнительные данные, возвратное значение `more_data_in_byte_stream()` будет ИСТИНА;
- иначе возвратное значение `more_data_in_byte_stream()` будет ЛОЖЬ.

`more_rbsp_data()` описан следующим образом:

- если в RBSP имеются дополнительные данные перед `rbsp_trailing_битов()`, возвратное значение `more_rbsp_data()` будет ИСТИНА;
- иначе возвратное значение `more_rbsp_data()` будет ЛОЖЬ.

Метод, который дает возможность определить, действительно ли имеются дополнительные данные в RBSP, описан приложением (или в Приложении В для приложений, которые используют формат потока байтов).

`more_rbsp_trailing_data()` описан следующим образом:

- если в RBSP имеются дополнительные данные, возвратное значение `more_rbsp_trailing_data()` будет ИСТИНА;
- иначе возвратное значение `more_rbsp_trailing_data()` будет ЛОЖЬ.

`next_bits(n)` создает следующие биты в потоке битов для целей сравнения без помощи указателя потока битов. Формирует вид этих следующих n битов в потоке битов с n аргументами. При использовании в потоке байтов, как это определено в Приложении В, возвратное значение `next_bits(n)` будет 0, если меньше чем n битов остается в потоке байтов.

`read_bits(n)` считывает следующие n битов из потока битов и продвигает указатель потока битов в положение на n битов. Если n равно 0, `read_bits(n)` описан возвратным значением, равным 0, и не продвигает указатель потока битов.

Категории (отмеченные в таблицах как С) описывают разделение на части данных секции при разделении не более, чем на три части данных секций. Разделение данных секции А содержит все элементы синтаксисов категории 2. Разделение данных секции В содержит все элементы синтаксисов категории 3. Разделение данных секции С содержит все элементы синтаксисов категории 4. Значение других категорий не описано. Для некоторых элементов синтаксисов используют две категории значений, разделенные вертикальной чертой. В этих случаях значения категорий, разделенных вертикальной чертой, будет x . В таких случаях значение категории, которая должна быть использована в дальнейшем, описана в тексте. Для структур синтаксиса, которые использованы в других структурах синтаксиса, категории для всех элементов синтаксиса, расположенных во включенной структуре синтаксиса, перечисляют, разделяя вертикальной чертой. Элемент синтаксиса или структуры синтаксиса с категорией, отмеченной как "Все", представлен во всех структурах синтаксисов, которые включают этот элемент синтаксиса или структуру синтаксиса. Для структур синтаксиса, использованных внутри других структур синтаксиса, численное значение категории, приведенной в таблице синтаксиса в месте включения элемента в структуре синтаксиса, отмеченной как "Все", применительно к этому элементу считают как "Все".

Следующие дескрипторы определяют процесс анализа каждого элемента синтаксиса. Для некоторых элементов синтаксиса используют два дескриптора, разделенные вертикальной чертой. В таких случаях левые дескрипторы применяют тогда, когда флаг `entropy_coding_mode_flag` равен 0, а правый – когда `entropy_coding_mode_flag` равен 1.

- `ae(v)`: элемент синтаксиса энтропийно-кодированный и контекстно-адаптивный. Процесс анализа для этого дескриптора описан в п. 9.3.
- `b(8)`: байт, содержащий строку битов любого шаблона с возвратным значением функции (8 битов). Процесс анализа для этого дескриптора описан возвратным значением функции `read_bits(8)`.
- `ce(v)`: элемент синтаксиса энтропийно-кодированный переменной длины и контекстно-адаптивный, у которого первый бит левый. Процесс анализа для этого дескриптора описан в п. 9.2.
- `f(n)`: строка битов с закрепленным шаблоном, в которой используют n битов, записанных (слева направо) с первым левым битом. Процесс анализа для этого дескриптора описан возвратным значением функции `read_bits(n)`.
- `i(n)`: целое со знаком с использованием n битов. Если n – это "v" в таблице синтаксиса, то число битов изменяется способом, который зависит от значения других элементов синтаксиса. Процесс анализа для

этого дескриптора описан возвратным значением функции `read_bits(n)`, интерпретируемой как два дополнительных целых представления с наиболее значимым битом, записанным первым.

- `me(v)`: элемент синтаксиса, отображенный кодированием Exp-Golomb с левым первым битом. Процесс анализа для этого дескриптора описан в п. 9.1.
- `se(v)`: элемент синтаксиса, кодированный Exp-Golomb целым со знаком с левым первым битом. Процесс анализа для этого дескриптора описан в п. 9.1.
- `te(v)`: элемент синтаксиса, кодированный усеченным методом Exp-Golomb с левым первым битом. Процесс анализа для этого дескриптора описан в п. 9.1.
- `u(n)`: целое без знака с использованием `n` битов. Если `n` – это "v" в таблице синтаксиса, то число битов изменяется способом, который зависит от значения других элементов синтаксиса. Процесс анализа для этого дескриптора описан возвратным значением функции `read_bits(n)`, интерпретированным как бинарное представление целого без знака с наиболее значимым битом, записанным первым.
- `ue(v)`: элемент синтаксиса, кодированный Exp-Golomb целым без знака с левым первым битом. Процесс анализа для этого дескриптора описан в п. 9.1.

7.3 Синтаксис в табличной форме

7.3.1 Синтаксис блока NAL

	С	Дескриптор
<code>nal_unit(NumBytesInNALunit) {</code>		
forbidden_zero_bit	Все	f(1)
nal_ref_idc	Все	u(2)
nal_unit_type	Все	u(5)
NumBytesInRBSP = 0		
for(i = 1; i < NumBytesInNALunit; i++) {		
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {		
rbsp_byte[NumBytesInRBSP++]	Все	b(8)
rbsp_byte[NumBytesInRBSP++]	Все	b(8)
i += 2		
emulation_prevention_three_byte /* равное 0x03 */	Все	f(8)
} else		
rbsp_byte[NumBytesInRBSP++]	Все	b(8)
}		
}		

7.3.2 Синтаксис полезной нагрузки последовательностей исходных байтов и концевых битов RBSP

7.3.2.1 Синтаксис RBSP набора параметров последовательности

	С	Дескриптор
seq_parameter_set_rbsp() {		
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
reserved_zero_4bits /* равное 0 */	0	u(4)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc == 100 profile_idc == 110 profile_idc == 122 profile_idc == 144) {		
chroma_format_idc	0	ue(v)
if(chroma_format_idc == 3)		
residual_colour_transform_flag	0	u(1)
bit_depth_luma_minus8	0	ue(v)
bit_depth_chroma_minus8	0	ue(v)
qpprime_y_zero_transform_bypass_flag	0	u(1)
seq_scaling_matrix_present_flag	0	u(1)
if(seq_scaling_matrix_present_flag)		
for(i = 0; i < 8; i++) {		
seq_scaling_list_present_flag[i]	0	u(1)
if(seq_scaling_list_present_flag[i])		
if(i < 6)		
scaling_list(ScalingList4x4[i], 16, UseDefaultScalingMatrix4x4Flag[i])	0	
else		
scaling_list(ScalingList8x8[i - 6], 64, UseDefaultScalingMatrix8x8Flag[i - 6])	0	
}		
}		
}		
log2_max_frame_num_minus4	0	ue(v)
pic_order_cnt_type	0	ue(v)
if(pic_order_cnt_type == 0)		
log2_max_pic_order_cnt_lsb_minus4	0	ue(v)
else if(pic_order_cnt_type == 1) {		
delta_pic_order_always_zero_flag	0	u(1)
offset_for_non_ref_pic	0	se(v)
offset_for_top_to_bottom_field	0	se(v)
num_ref_frames_in_pic_order_cnt_cycle	0	ue(v)
for(i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++)		
offset_for_ref_frame[i]	0	se(v)
}		
num_ref_frames	0	ue(v)
gaps_in_frame_num_value_allowed_flag	0	u(1)
pic_width_in_mbs_minus1	0	ue(v)
pic_height_in_map_units_minus1	0	ue(v)

frame_mbs_only_flag	0	u(1)
if(!frame_mbs_only_flag)		
mb_adaptive_frame_field_flag	0	u(1)
direct_8x8_inference_flag	0	u(1)
frame_cropping_flag	0	u(1)
if(frame_cropping_flag) {		
frame_crop_left_offset	0	ue(v)
frame_crop_right_offset	0	ue(v)
frame_crop_top_offset	0	ue(v)
frame_crop_bottom_offset	0	ue(v)
}		
vui_parameters_present_flag	0	u(1)
if(vui_parameters_present_flag)		
vui_parameters()	0	
rbsp_trailing_bits()	0	
}		

7.3.2.1.1 Синтаксис списка масштабирования

scaling_list(scalingList, sizeOfScalingList, useDefaultScalingMatrixFlag) {	С	Дескриптор
lastScale = 8		
nextScale = 8		
for(j = 0; j < sizeOfScalingList; j++) {		
if(nextScale != 0) {		
delta_scale	0 1	se(v)
nextScale = (lastScale + delta_scale + 256) % 256		
useDefaultScalingMatrixFlag = (j == 0 && nextScale == 0)		
}		
scalingList[j] = (nextScale == 0) ? lastScale : nextScale		
lastScale = scalingList[j]		
}		
}		

7.3.2.1.2 Синтаксис RBSP расширения набора параметров последовательности

seq_parameter_set_extension_rbsp() {	С	Дескриптор
seq_parameter_set_id	10	ue(v)
aux_format_idc	10	ue(v)
if(aux_format_idc != 0) {		
bit_depth_aux_minus8	10	ue(v)
alpha_incr_flag	10	u(1)
alpha_opaque_value	10	u(v)
alpha_transparent_value	10	u(v)
}		
additional_extension_flag	10	u(1)
rbsp_trailing_bits()	10	
}		

7.3.2.2 Синтаксис Rbsp набора параметров изображения

	С	Дескриптор
pic_parameter_set_rbsp() {		
pic_parameter_set_id	1	ue(v)
seq_parameter_set_id	1	ue(v)
entropy_coding_mode_flag	1	u(1)
pic_order_present_flag	1	u(1)
num_slice_groups_minus1	1	ue(v)
if(num_slice_groups_minus1 > 0) {		
slice_group_map_type	1	ue(v)
if(slice_group_map_type == 0)		
for(iGroup = 0; iGroup <= num_slice_groups_minus1; iGroup++)		
run_length_minus1 [iGroup]	1	ue(v)
else if(slice_group_map_type == 2)		
for(iGroup = 0; iGroup < num_slice_groups_minus1; iGroup++) {		
top_left [iGroup]	1	ue(v)
bottom_right [iGroup]	1	ue(v)
}		
else if(slice_group_map_type == 3 slice_group_map_type == 4 slice_group_map_type == 5) {		
slice_group_change_direction_flag	1	u(1)
slice_group_change_rate_minus1	1	ue(v)
} else if(slice_group_map_type == 6) {		
pic_size_in_map_units_minus1	1	ue(v)
for(i = 0; i <= pic_size_in_map_units_minus1; i++)		
slice_group_id [i]	1	u(v)
}		
}		
}		
num_ref_idx_l0_active_minus1	1	ue(v)
num_ref_idx_l1_active_minus1	1	ue(v)
weighted_pred_flag	1	u(1)
weighted_bipred_idc	1	u(2)
pic_init_qp_minus26 /* относительно 26 */	1	se(v)
pic_init_qs_minus26 /* относительно 26 */	1	se(v)
chroma_qp_index_offset	1	se(v)
deblocking_filter_control_present_flag	1	u(1)
constrained_intra_pred_flag	1	u(1)
redundant_pic_cnt_present_flag	1	u(1)
if(more_rbsp_data()) {		
transform_8x8_mode_flag	1	u(1)
pic_scaling_matrix_present_flag	1	u(1)
if(pic_scaling_matrix_present_flag)		
for(i = 0; i < 6 + 2* transform_8x8_mode_flag; i++) {		
pic_scaling_list_present_flag [i]	1	u(1)
if(pic_scaling_list_present_flag[i])		
if(i < 6)		

scaling_list(ScalingList4x4[i], 16, UseDefaultScalingMatrix4x4Flag[i])	1	
Else		
scaling_list(ScalingList8x8[i – 6], 64, UseDefaultScalingMatrix8x8Flag[i – 6])	1	
}		
second_chroma_qp_index_offset	1	se(v)
}		
rbsp_trailing_bits()	1	
}		

7.3.2.3 Синтаксис Rbsp дополнительной расширенной информации

sei_rbsp() {	C	Дескриптор
Do		
sei_message()	5	
while(more_rbsp_data())		
rbsp_trailing_bits()	5	
}		

7.3.2.3.1 Синтаксис сообщения дополнительной расширенной информации

sei_message() {	C	Дескриптор
payloadType = 0		
while(next_bits(8) == 0xFF) {		
ff_byte /* равно 0xFF */	5	f(8)
payloadType += 255		
}		
last_payload_type_byte	5	u(8)
payloadType += last_payload_type_byte		
payloadSize = 0		
while(next_bits(8) == 0xFF) {		
ff_byte /* равно 0xFF */	5	f(8)
payloadSize += 255		
}		
last_payload_size_byte	5	u(8)
payloadSize += last_payload_size_byte		
sei_payload(payloadType, payloadSize)	5	
}		

7.3.2.4 Синтаксис разграничителя блока доступа RBSP

access_unit_delimiter_rbsp() {	C	Дескриптор
primary_pic_type	6	u(3)
rbsp_trailing_bits()	6	
}		

7.3.2.5 Синтаксис RBSP конца последовательности

end_of_seq_rbsp() {	C	Дескриптор
}		

7.3.2.6 Синтаксис RBSP конца потока

end_of_stream_rbsp() {	C	Дескриптор
}		

7.3.2.7 Синтаксис RBSP данных заполнителя

filler_data_rbsp() {	C	Дескриптор
while(next_bits(8) == 0xFF)		
ff_byte /* равное 0xFF */	9	f(8)
rbsp_trailing_bits ()	9	
}		

7.3.2.8 Синтаксис RBSP слоя секции без разделения

slice_layer_without_partitioning_rbsp() {	C	Дескриптор
slice_header()	2	
slice_data() /* все категории синтаксиса slice_data() */	2 3 4	
rbsp_slice_trailing_bits ()	2	
}		

7.3.2.9 Синтаксис RBSP разделения данных секции

7.3.2.9.1 Синтаксис RBSP части А данных секции

slice_data_partition_a_layer_rbsp() {	C	Дескриптор
slice_header()	2	
slice_id	Все	ue(v)
slice_data() /* только части категории 2 синтаксиса slice_data() */	2	
rbsp_slice_trailing_bits ()	2	
}		

7.3.2.9.2 Синтаксис RBSP части В данных секции

	С	Дескриптор
slice_data_partition_b_layer_rbsp() {		
slice_id	Все	ue(v)
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	Все	ue(v)
slice_data() /* только части категории 3 синтаксиса slice_data() */	3	
rbsp_slice_trailing_bits()	3	
}		

7.3.2.9.3 Синтаксис RBSP части С данных секции

	С	Дескриптор
slice_data_partition_c_layer_rbsp() {		
slice_id	Все	ue(v)
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	Все	ue(v)
slice_data() /* только части категории 4 синтаксиса slice_data() */	4	
rbsp_slice_trailing_bits()	4	
}		

7.3.2.10 Синтаксис концевых битов секции RBSP

	С	Дескриптор
rbsp_slice_trailing_bits() {		
rbsp_trailing_bits()	Все	
if(entropy_coding_mode_flag)		
while(more_rbsp_trailing_data())		
cabac_zero_word /* равно 0x0000 */	Все	f(16)
}		

7.3.2.11 Синтаксис концевых битов RBSP

	С	Дескриптор
rbsp_trailing_bits() {		
rbsp_stop_one_bit /* равно 1 */	Все	f(1)
while(!byte_aligned())		
rbsp_alignment_zero_bit /* равно 0 */	Все	f(1)
}		

7.3.3 Синтаксис заголовка секции

	С	Дескриптор
slice_header() {		
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(nal_unit_type == 5)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type == 0) {		
pic_order_cnt_lsb	2	u(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt_bottom	2	se(v)
}		
if(pic_order_cnt_type == 1 && !delta_pic_order_always_zero_flag) {		
delta_pic_order_cnt[0]	2	se(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt[1]	2	se(v)
}		
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	2	ue(v)
if(slice_type == B)		
direct_spatial_mv_pred_flag	2	u(1)
if(slice_type == P slice_type == SP slice_type == B) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_l0_active_minus1	2	ue(v)
if(slice_type == B)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		
ref_pic_list_reordering()	2	
if((weighted_pred_flag && (slice_type == P slice_type == SP)) (weighted_bipred_idc == 1 && slice_type == B))		
pred_weight_table()	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking()	2	
if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
if(slice_type == SP slice_type == SI) {		
if(slice_type == SP)		
sp_for_switch_flag	2	u(1)

slice_qs_delta	2	se(v)
}		
if(deblocking_filter_control_present_flag) {		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1) {		
slice_alpha_c0_offset_div2	2	se(v)
slice_beta_offset_div2	2	se(v)
}		
}		
if(num_slice_groups_minus1 > 0 && slice_group_map_type >= 3 && slice_group_map_type <= 5)		
slice_group_change_cycle	2	u(v)
}		

7.3.3.1 Синтаксис изменения порядка списка контрольных изображений

	С	Дескриптор
ref_pic_list_reordering() {		
if(slice_type != I && slice_type != SI) {		
ref_pic_list_reordering_flag_l0	2	u(1)
if(ref_pic_list_reordering_flag_l0)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0 reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
if(slice_type == B) {		
ref_pic_list_reordering_flag_l1	2	u(1)
if(ref_pic_list_reordering_flag_l1)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0 reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
}		

7.3.3.2 Синтаксис таблицы веса предсказания

pred_weight_table() {	С	Дескриптор
luma_log2_weight_denom	2	ue(v)
if(chroma_format_idc != 0)		
chroma_log2_weight_denom	2	ue(v)
for(i = 0; i <= num_ref_idx_l0_active_minus1; i++) {		
luma_weight_l0_flag	2	u(1)
if(luma_weight_l0_flag) {		
luma_weight_l0[i]	2	se(v)
luma_offset_l0[i]	2	se(v)
}		
if(chroma_format_idc != 0) {		
chroma_weight_l0_flag	2	u(1)
if(chroma_weight_l0_flag)		
for(j = 0; j < 2; j++) {		
chroma_weight_l0[i][j]	2	se(v)
chroma_offset_l0[i][j]	2	se(v)
}		
}		
}		
if(slice_type == B)		
for(i = 0; i <= num_ref_idx_l1_active_minus1; i++) {		
luma_weight_l1_flag	2	u(1)
if(luma_weight_l1_flag) {		
luma_weight_l1[i]	2	se(v)
luma_offset_l1[i]	2	se(v)
}		
if(chroma_format_idc != 0) {		
chroma_weight_l1_flag	2	u(1)
if(chroma_weight_l1_flag)		
for(j = 0; j < 2; j++) {		
chroma_weight_l1[i][j]	2	se(v)
chroma_offset_l1[i][j]	2	se(v)
}		
}		
}		
}		

7.3.3.3 Синтаксис разметки декодированного контрольного изображения

	С	Дескриптор
dec_ref_pic_marking() {		
if(nal_unit_type == 5) {		
no_output_of_prior_pics_flag	2 5	u(1)
long_term_reference_flag	2 5	u(1)
} else {		
adaptive_ref_pic_marking_mode_flag	2 5	u(1)
if(adaptive_ref_pic_marking_mode_flag)		
do {		
memory_management_control_operation	2 5	ue(v)
if(memory_management_control_operation == 1 memory_management_control_operation == 3)		
difference_of_pic_nums_minus1	2 5	ue(v)
if(memory_management_control_operation == 2)		
long_term_pic_num	2 5	ue(v)
if(memory_management_control_operation == 3 memory_management_control_operation == 6)		
long_term_frame_idx	2 5	ue(v)
if(memory_management_control_operation == 4)		
max_long_term_frame_idx_plus1	2 5	ue(v)
} while(memory_management_control_operation != 0)		
}		
}		

7.3.4 Синтаксис данных секции

	С	Дескриптор
slice_data() {		
if(entropy_coding_mode_flag)		
while(!byte_aligned())		
cabac_alignment_one_bit	2	f(1)
CurrMbAddr = first_mb_in_slice * (1 + MbaffFrameFlag)		
moreDataFlag = 1		
prevMbSkipped = 0		
do {		
if(slice_type != I && slice_type != SI)		
if(!entropy_coding_mode_flag) {		
mb_skip_run	2	ue(v)
prevMbSkipped = (mb_skip_run > 0)		
for(i=0; i<mb_skip_run; i++)		
CurrMbAddr = NextMbAddress(CurrMbAddr)		
moreDataFlag = more_rbsp_data()		
} else {		
mb_skip_flag	2	ae(v)
moreDataFlag = !mb_skip_flag		
}		
if(moreDataFlag) {		
if(MbaffFrameFlag && (CurrMbAddr % 2 == 0 (CurrMbAddr % 2 == 1 && prevMbSkipped)))		
mb_field_decoding_flag	2	u(1) ae(v)
macroblock_layer()	2 3 4	
}		
if(!entropy_coding_mode_flag)		
moreDataFlag = more_rbsp_data()		
else {		
if(slice_type != I && slice_type != SI)		
prevMbSkipped = mb_skip_flag		
if(MbaffFrameFlag && CurrMbAddr % 2 == 0)		
moreDataFlag = 1		
else {		
end_of_slice_flag	2	ae(v)
moreDataFlag = !end_of_slice_flag		
}		
}		
CurrMbAddr = NextMbAddress(CurrMbAddr)		
} while(moreDataFlag)		
}		

7.3.5 Синтаксис слоя макроблока

маcroblock_layer() {	C	Дескриптор
mb_type	2	ue(v) ae(v)
if(mb_type == I_PCM) {		
while(!byte_aligned())		
pcm_alignment_zero_bit	2	f(1)
for(i = 0; i < 256; i++)		
pcm_sample_luma[i]	2	u(v)
for(i = 0; i < 2 * MbWidthC * MbHeightC; i++)		
pcm_sample_chroma[i]	2	u(v)
} else {		
noSubMbPartSizeLessThan8x8Flag = 1		
if(mb_type != I_NxN && MbPartPredMode(mb_type, 0) != Intra_16x16 && NumMbPart(mb_type) == 4) {		
sub_mb_pred(mb_type)	2	
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8) {		
if(NumSubMbPart(sub_mb_type[mbPartIdx]) > 1)		
noSubMbPartSizeLessThan8x8Flag = 0		
} else if(!direct_8x8_inference_flag)		
noSubMbPartSizeLessThan8x8Flag = 0		
} else {		
if(transform_8x8_mode_flag && mb_type == I_NxN)		
transform_size_8x8_flag	2	u(1) ae(v)
mb_pred(mb_type)	2	
}		
if(MbPartPredMode(mb_type, 0) != Intra_16x16) {		
coded_block_pattern	2	me(v) ae(v)
if(CodedBlockPatternLuma > 0 && transform_8x8_mode_flag && mb_type != I_NxN && noSubMbPartSizeLessThan8x8Flag && (mb_type != B_Direct_16x16 direct_8x8_inference_flag))		
transform_size_8x8_flag	2	u(1) ae(v)
}		
if(CodedBlockPatternLuma > 0 CodedBlockPatternChroma > 0 MbPartPredMode(mb_type, 0) == Intra_16x16) {		
mb_qp_delta	2	se(v) ae(v)
residual()	3 4	
}		
}		
}		
}		

7.3.5.1 Синтаксис предсказания макроблока

mb_pred(mb_type) {	C	Дескриптор
if(MbPartPredMode(mb_type, 0) == Intra_4x4 MbPartPredMode(mb_type, 0) == Intra_8x8 MbPartPredMode(mb_type, 0) == Intra_16x16) {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag [luma4x4BlkIdx]	2	u(1) ae(v)
if(!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode [luma4x4BlkIdx]	2	u(3) ae(v)
}		
if(MbPartPredMode(mb_type, 0) == Intra_8x8)		
for(luma8x8BlkIdx=0; luma8x8BlkIdx<4; luma8x8BlkIdx++) {		
prev_intra8x8_pred_mode_flag [luma8x8BlkIdx]	2	u(1) ae(v)
if(!prev_intra8x8_pred_mode_flag[luma8x8BlkIdx])		
rem_intra8x8_pred_mode [luma8x8BlkIdx]	2	u(3) ae(v)
}		
if(chroma_format_idc != 0)		
intra_chroma_pred_mode	2	ue(v) ae(v)
} else if(MbPartPredMode(mb_type, 0) != Direct) {		
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 mb_field_decoding_flag) && MbPartPredMode(mb_type, mbPartIdx) != Pred_L1)		
ref_idx_l0 [mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 mb_field_decoding_flag) && MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
ref_idx_l1 [mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode(mb_type, mbPartIdx) != Pred_L1)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0 [mbPartIdx][0][compIdx]	2	se(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1 [mbPartIdx][0][compIdx]	2	se(v) ae(v)
}		
}		

7.3.5.2 Синтаксис предсказания субмакроблока

sub_mb_pred(mb_type) {	C	Дескриптор
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
sub_mb_type [mbPartIdx]	2	ue(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 mb_field_decoding_flag) && mb_type != P_8x8ref0 && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
ref_idx_l0 [mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 mb_field_decoding_flag) && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
ref_idx_l1 [mbPartIdx]	2	te(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0 [mbPartIdx][subMbPartIdx][compIdx]	2	se(v) ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1 [mbPartIdx][subMbPartIdx][compIdx]	2	se(v) ae(v)
}		

7.3.5.3 Синтаксис данных остатка

	С	Дескриптор
residual() {		
if(!entropy_coding_mode_flag)		
residual_block = residual_block_cavlc		
else		
residual_block = residual_block_cabac		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16DCLevel, 16)	3	
for(i8x8 = 0; i8x8 < 4; i8x8++) /* each luma 8x8 block */		
if(!transform_size_8x8_flag !entropy_coding_mode_flag)		
for(i4x4 = 0; i4x4 < 4; i4x4++) { /* each 4x4 sub-block of block */		
if(CodedBlockPatternLuma & (1 << i8x8))		
if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
residual_block(Intra16x16ACLevel[i8x8 * 4 + i4x4], 15)	3	
else		
residual_block(LumaLevel[i8x8 * 4 + i4x4], 16)	3 4	
else if(MbPartPredMode(mb_type, 0) == Intra_16x16)		
for(i = 0; i < 15; i++)		
Intra16x16ACLevel[i8x8 * 4 + i4x4][i] = 0		
else		
for(i = 0; i < 16; i++)		
LumaLevel[i8x8 * 4 + i4x4][i] = 0		
if(!entropy_coding_mode_flag && transform_size_8x8_flag)		
for(i = 0; i < 16; i++)		
LumaLevel8x8[i8x8][4 * i + i4x4] =		
LumaLevel[i8x8 * 4 + i4x4][i]		
}		
else if(CodedBlockPatternLuma & (1 << i8x8))		
residual_block(LumaLevel8x8[i8x8], 64)	3 4	
else		
for(i = 0; i < 64; i++)		
LumaLevel8x8[i8x8][i] = 0		
if(chroma_format_idc != 0) {		
NumC8x8 = 4 / (SubWidthC * SubHeightC)		
for(iCbCr = 0; iCbCr < 2; iCbCr++)		
if(CodedBlockPatternChroma & 3) /* chroma DC residual present */		
residual_block(ChromaDCLevel[iCbCr], 4 * NumC8x8)	3 4	
else		
for(i = 0; i < 4 * NumC8x8; i++)		
ChromaDCLevel[iCbCr][i] = 0		
for(iCbCr = 0; iCbCr < 2; iCbCr++)		
for(i8x8 = 0; i8x8 < NumC8x8; i8x8++)		
for(i4x4 = 0; i4x4 < 4; i4x4++)		
if(CodedBlockPatternChroma & 2)		
/* chroma AC residual present */		
residual_block(ChromaACLevel[iCbCr][i8x8*4+i4x4], 15)	3 4	
else		
for(i = 0; i < 15; i++)		
ChromaACLevel[iCbCr][i8x8*4+i4x4][i] = 0		
}		

7.3.5.3.1 Синтаксис CAVLC блока остатка

	С	Дескриптор
residual_block_cavlc(coeffLevel, maxNumCoeff) {		
for(i = 0; i < maxNumCoeff; i++)		
coeffLevel[i] = 0		
coeff_token	3 4	ce(v)
if(TotalCoeff(coeff_token) > 0) {		
if(TotalCoeff(coeff_token) > 10 && TrailingOnes(coeff_token) < 3)		
suffixLength = 1		
else		
suffixLength = 0		
for(i = 0; i < TotalCoeff(coeff_token); i++)		
if(i < TrailingOnes(coeff_token)) {		
trailing_ones_sign_flag	3 4	u(1)
level[i] = 1 - 2 * trailing_ones_sign_flag		
} else {		
level_prefix	3 4	ce(v)
levelCode = (Min(15, level_prefix) << suffixLength)		
if(suffixLength > 0 level_prefix >= 14) {		
level_suffix	3 4	u(v)
levelCode += level_suffix		
}		
if(level_prefix >= 15 && suffixLength == 0)		
levelCode += 15		
if(level_prefix >= 16)		
levelCode += (1 << (level_prefix - 3)) - 4096		
if(i == TrailingOnes(coeff_token) && TrailingOnes(coeff_token) < 3)		
levelCode += 2		
if(levelCode % 2 == 0)		
level[i] = (levelCode + 2) >> 1		
else		
level[i] = (-levelCode - 1) >> 1		
if(suffixLength == 0)		
suffixLength = 1		
if(Abs(level[i]) > (3 << (suffixLength - 1)) && suffixLength < 6)		
suffixLength++		
}		
if(TotalCoeff(coeff_token) < maxNumCoeff) {		
total_zeros	3 4	ce(v)
zerosLeft = total_zeros		
} else		
zerosLeft = 0		
for(i = 0; i < TotalCoeff(coeff_token) - 1; i++) {		
if(zerosLeft > 0) {		
run_before	3 4	ce(v)
run[i] = run_before		
} else		
run[i] = 0		

zerosLeft = zerosLeft - run[i]		
}		
run[TotalCoeff(coeff_token) - 1] = zerosLeft		
coeffNum = -1		
for(i = TotalCoeff(coeff_token) - 1; i >= 0; i--) {		
coeffNum += run[i] + 1		
coeffLevel[coeffNum] = level[i]		
}		
}		
}		

7.3.5.3.2 Синтаксис САВАС блока остатка

residual_block_cabac(coeffLevel, maxNumCoeff) {	С	Дескриптор
if(maxNumCoeff == 64)		
coded_block_flag = 1		
else		
coded_block_flag	3 4	ae(v)
if(coded_block_flag) {		
numCoeff = maxNumCoeff		
i = 0		
do {		
significant_coeff_flag[i]	3 4	ae(v)
if(significant_coeff_flag[i]) {		
last_significant_coeff_flag[i]	3 4	ae(v)
if(last_significant_coeff_flag[i]) {		
numCoeff = i + 1		
for(j = numCoeff; j < maxNumCoeff; j++)		
coeffLevel[j] = 0		
}		
}		
i++		
} while(i < numCoeff - 1)		
coeff_abs_level_minus1[numCoeff - 1]	3 4	ae(v)
coeff_sign_flag[numCoeff - 1]	3 4	ae(v)
coeffLevel[numCoeff - 1] = (coeff_abs_level_minus1[numCoeff - 1] + 1) * (1 - 2 * coeff_sign_flag[numCoeff - 1])		
for(i = numCoeff - 2; i >= 0; i--)		
if(significant_coeff_flag[i]) {		
coeff_abs_level_minus1[i]	3 4	ae(v)
coeff_sign_flag[i]	3 4	ae(v)
coeffLevel[i] = (coeff_abs_level_minus1[i] + 1) * (1 - 2 * coeff_sign_flag[i])		
} else		
coeffLevel[i] = 0		
} else		
for(i = 0; i < maxNumCoeff; i++)		
coeffLevel[i] = 0		
}		

7.4 Семантика

В данном разделе содержится описание семантики, связанной с синтаксическими структурами и с элементами синтаксиса в пределах этих структур. Если семантика элемента синтаксиса определена с использованием таблицы или набора таблиц, значения, не указанные в таблице(ах), не должны присутствовать в потоке битов, если они не определены иным образом в настоящей Рекомендации | Международном стандарте.

7.4.1 Семантика блока NAL

ПРИМЕЧАНИЕ 1. – Уровень VCL описан, чтобы эффективно представить содержание видеоданных. Уровень NAL описан, чтобы сформатировать эти данные и создать информационный заголовок способом, соответствующим передаче по разным каналам связи и хранению медианформации. Все данные содержатся в блоках NAL, каждый из которых содержит целое число байтов. Блок NAL определяет групповой формат для использования как в пакетно-ориентированных системах, так и в системах с потоками битов. Формат блоков NAL для обоих видов систем идентичен, за исключением того, что каждому блоку NAL может предшествовать префикс кода запуска и дополнительное заполнение байтами в формате потока байтов.

NumBytesInNALunit определяет размер блока NAL в байтах. Эта величина требуется для декодирования блока NAL. Некоторые формы разграничения блоков NAL необходимы для возможности введения NumBytesInNALunit. Один из методов разграничения описан в Приложении В для формата потока байтов. Другие методы могут быть описаны вне настоящей Рекомендации | Международного стандарта.

forbidden_zero_bit должен быть равным 0.

nal_ref_idc, не равное 0, определяет, что блок NAL содержит набор параметров последовательности или набор параметров изображения, или секцию контрольного изображения, или секцию данных разделения контрольного изображения.

nal_ref_idc равно 0 для блока NAL, содержащего секцию или разделенную часть данных секции, и указывает, что секция или часть секции разделенных данных не являются частью неконтрольного изображения.

nal_ref_idc не должно быть равным 0 для набора параметров последовательности или расширения набора параметров последовательности либо набора параметров изображения блоков NAL. Если **nal_ref_idc** равно 0 для одной секции или секции разделенных данных блока NAL конкретного изображения, то оно должно быть равным 0 для всей секции и секции разделенных данных блока NAL изображения.

nal_ref_idc не должно быть равным 0 для блоков NAL IDR, т. е. блоков NAL с **nal_unit_type** равным 5.

nal_ref_idc должен быть равным 0 для всех блоков NAL, имеющих **nal_unit_type** равным 6, 9, 10, 11 или 12.

nal_unit_type определяет тип структуры данных RBSP, содержащуюся в блоке NAL, как описано в таблице 7-1. Блоки NAL VCL описаны как блоки NAL, имеющие **nal_unit_type** равным от 1 до 5 включительно. Все оставшиеся блоки NAL называют блоками NAL не уровня VCL.

В столбце, помеченном "С" в таблице 7-1, перечисляются категории элементов синтаксиса, которые могут быть представлены в блоке NAL. Кроме того, элементы синтаксиса с категорией синтаксиса "Все" могут быть представлены как определенные синтаксисом и семантикой структуры данных RBSP. Присутствие или отсутствие любого из элементов синтаксиса конкретно перечисленной категории определяют из синтаксиса и семантики связанных структур данных RBSP. **nal_unit_type** не должно быть равно 3 или 4, пока по крайней мере один элемент синтаксиса не будет представлен в структуре данных RBSP, имеющую значение категории элемента синтаксиса, равную значению **nal_unit_type** и не попавшую в категорию "Все".

Таблица 7-1 – Коды типов блока NAL

nal_unit_type	Содержания блока NAL и структура синтаксиса RBSP	C
0	Не определено	
1	Кодированная секция не IDR изображения slice_layer_without_partitioning_rbsp()	2, 3, 4
2	Часть А данных кодированной секции slice_data_partition_a_layer_rbsp()	2
3	Часть В данных кодированной секции slice_data_partition_b_layer_rbsp()	3
4	Часть С данных кодированной секции slice_data_partition_c_layer_rbsp()	4
5	Кодированная секция изображения IDR slice_layer_without_partitioning_rbsp()	2, 3
6	Дополнительная расширенная информация (SEI) sei_rbsp()	5
7	Набор параметров последовательности seq_parameter_set_rbsp()	0
8	Набор параметров изображения pic_parameter_set_rbsp()	1
9	Разграничитель блока доступа access_unit_delimiter_rbsp()	6
10	Конец последовательности end_of_seq_rbsp()	7
11	Конец потока end_of_stream_rbsp()	8
12	Данные заполнения filler_data_rbsp()	9
13	Расширение набора параметров последовательности	
14..18	Зарезервировано	
19	Кодированная секция дополнительно кодированного изображения без разделения slice_layer_without_partitioning_rbsp()	2, 3, 4
20..23	Зарезервировано	
24..31	Не определено	

Блоки NAL, значением nal_unit_type которых являются 13 и 19, могут отбрасываться декодерами, не затрагивая при этом процесс декодирования блоков NAL со значениями nal_unit_type, отличными от 13 и 19, и не нарушая соответствия настоящей Рекомендации | Международному стандарту.

Блоки NAL, которые используют nal_unit_type, равное 0 или в диапазоне 24..31 включительно, не должны влиять на процесс декодирования, описанный в настоящей Рекомендации | Международном стандарте.

ПРИМЕЧАНИЕ 2. – Типы блоков NAL 0 и 24..31 могут быть использованы, как определено приложением. Никакие процессы декодирования для этих значений nal_unit_type не определены в настоящей Рекомендации | Международном стандарте.

Декодеры должны игнорировать (удалять из потока битов и аннулировать) содержание всех блоков NAL, которые используют зарезервированные значения nal_unit_type.

ПРИМЕЧАНИЕ 3. – Это требование допускает будущее определение расширений совместимости с этой Рекомендацией | Международным стандартом.

В данном тексте кодированную секцию блока NAL собирательно считают кодированной секцией блока NAL IDR-изображения.

Если значение `nal_unit_type` равно 5 для блока NAL, содержащего секцию кодированного изображения, то значение `nal_unit_type` должно быть 5 для всех других VCL блоков NAL этого же кодированного изображения. Такое изображение считают изображением IDR.

ПРИМЕЧАНИЕ 4. – Секция разделения данных не может быть использована для IDR изображений.

`rbsp_byte[i]` – это *i*-й байт RBSP. Нагрузка RBSP описана как упорядоченная последовательность байтов следующим образом.

RBSP содержит SODB при следующих условиях.

- Если SODB пустое (т. е. длиной нуль битов), RBSP также пустое.
- Иначе RBSP содержит SODB следующим образом:
 - 1) Первый байт RBSP содержит (старших значащих, крайних левых) восемь битов SODB; следующий байт RBSP должен содержать следующие восемь SODB и т. д., пока не останется меньше восьми битов SODB.
 - 2) `rbsp_trailing_bits()` представлен после SODB следующим образом:
 - i) первые (старшие значащие, крайние левые) биты последнего байта RBSP содержат оставшиеся биты SODB, (если таковые имеются);
 - ii) следующий бит состоит из единственного `rbsp_stop_one_bit`, равного 1; и
 - iii) если `rbsp_stop_one_bit` не последний бит байт-синхронизированного байта, то одно или более значений `rbsp_alignment_zero_bit` представлено в результате синхронизации байтов.
 - 3) Одно или более значений `cabac_zero_word` 16-ти битовых элементов синтаксиса, равных 0x0000, могут быть представлены в некоторых нагрузках RBSP после `rbsp_trailing_bits()` в конце RBSP.

Структуры синтаксиса, имеющие эти свойства RBSP, описывают в таблицах синтаксиса, используя суффикс "`_rbsp`". Эти структуры должны находиться в блоках NAL как содержание `rbsp_byte[i]` данных байтов. Связь структур синтаксиса RBSP с блоками NAL должна быть такой, как это описано в таблице 7-1.

ПРИМЕЧАНИЕ 5. – Если границы RBSP известны, декодер может извлекать SODB из RBSP, объединяя биты байтов RBSP и отбрасывая значение `rbsp_stop_one_bit`, которое будет последним (младшим значащим, крайним правым) битом, равным 1, а также отбрасывая любые следующие за ним (менее значащие, ближе к правому краю) биты, которые равны 0. Данные, необходимые для этого процесса декодирования, содержатся в части SODB нагрузки RBSP.

`emulation_prevention_three_byte` – это байт, равный 0x03. Если в блоке NAL представлена эмуляция `emulation_prevention_three_byte`, то она должна быть отброшена процессом декодирования.

Последний байт блока NAL не должен быть равным 0x00.

В блоке NAL, следующие трехбайтовые последовательности не должны находиться в любых байт-синхронизированных положениях:

- 0x000000;
- 0x000001;
- 0x000002.

В блоке NAL любая четырехбайтовая последовательность, которая начинается с 0x000003 и иная, чем нижеперечисленные последовательности, не должна находиться в любом байт-синхронизированном положении:

- 0x00000300;
- 0x00000301;
- 0x00000302;
- 0x00000303.

ПРИМЕЧАНИЕ 6. – Если `nal_unit_type` равно 0, следует с особой тщательностью проектировать декодеры, с тем чтобы избежать наличия вышеперечисленных трехбайтовых и четырехбайтовых последовательностей в начале синтаксической структуры блока NAL, поскольку синтаксический элемент `emulation_prevention_three_byte` не может быть третьим байтом блока NAL.

7.4.1.1 Инкапсуляция SODB в RBSP (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Формирование инкапсуляции SODB в RBSP и пользование `emulation_prevention_three_byte` для инкапсуляции RBSP в блок NAL описано для следующих целей:

- предотвратить эмуляцию стартовых кодов (кодов запуска) в блоки NAL, в то же время позволяя представить любую строку SODB в блоке NAL;
- обеспечить идентификацию конца SODB в блоке NAL, отыскивая RBSP для значения `rbsp_stop_one_bit`, которое начинается в конце RBSP; и

- обеспечить, чтобы при тех же условиях блок NAL имел размер больше, чем SODB (используя одно или более значений `cabac_zero_word`).

Кодер может создать блок NAL из RBSP следующей процедурой.

Отыскивают строку данных RBSP для байт-синхронизированных битов бинарных шаблонов:

'00000000 00000000 000000xx' (где xx представляет любой 2-х битовый шаблон: 00, 01, 10 или 11),

и извлекают байт, равный 0x03, для замены этих битов шаблонов на шаблоны:

'00000000 00000000 00000011 000000xx',

наконец, если последний байт данных RBSP равен 0x00 (что может случиться, если RBSP оканчивается на `cabac_zero_word`), то конечный, равный 0x03 байт присоединяют к концу данных.

Затем результирующую последовательность байтов ставят впереди первого байта блока NAL, где имеется указание типа структуры данных RBSP, которую он содержит. Это является результатом создания собственно блока NAL.

Этот процесс допускает любое представление SODB в блоке NAL, в то же время обеспечивая, чтобы:

- байт-синхронизированный префикс кода запуска не эмулировался в блоке NAL; и
- последовательность 8 нулевых значений битов, которая следует за префиксом кода запуска (независимо от байт-синхронизации), не эмулировалась в блоке NAL.

7.4.1.2 Порядок блоков NAL и связь с кодированными изображениями, блоками доступа и видеопоследовательностями

В этом пункте описаны ограничения на порядок блоков NAL в потоке битов. Любой порядок блоков NAL в потоке битов, подчиняющийся этим ограничениям, в этом тексте считают порядком декодирования блоков NAL. В блоке NAL синтаксис в пп. 7.3, D.1 и E.1 определяет порядок декодирования элементов синтаксиса. Соответствующие настоящей Рекомендации | Международному стандарту декодеры должны быть в состоянии принимать блоки NAL и их элементы синтаксиса в порядке декодирования.

7.4.1.2.1 Порядок RBSP последовательности и набора параметров изображения и их активация

ПРИМЕЧАНИЕ 1. – Последовательность и механизм установки параметров изображения нарушает передачу редко изменяемой информации при передаче кодированных данных макроблока. Установки последовательности и параметры изображения для некоторых приложений можно передавать "вне полосы", используя надежный транспортный механизм.

RBSP набора параметров изображения включает параметры, которые можно считать кодированными секциями блоков NAL или кодированными секциями разделенных данных типа A блоков NAL одного или более кодированных изображений. Каждую RBSP набора параметров изображения первоначально считают неактивированной в начале операции процесса декодирования. Принято, что не более одной RBSP набора параметров изображения можно считать активной в любой момент во время операции процесса декодирования, а активация RBSP любого конкретного набора параметров изображения приводит к деактивации ранее действующей RBSP набора параметров изображения (если имеется).

Если RBSP набора параметров изображения (с конкретным значением `pic_parameter_set_id`) не действует и считается кодированной секцией блока NAL или кодированной секцией разделенных данных A блока NAL (при использовании этого значения `pic_parameter_set_id`), то она активирована. Эту RBSP набора параметров изображения называют активной RBSP набора параметров изображения до тех пор, пока она не будет деактивирована другой RBSP набора параметров изображения. RBSP набора параметров изображения с конкретным значением `pic_parameter_set_id` должна быть доступна процессу декодирования до ее активации.

Любая установка параметра изображения блока NAL со значением `pic_parameter_set_id` для действующей RBSP набора параметров изображения должна иметь то же самое содержание, что и действующая RBSP набора параметров изображения, пока за ней не последует последний блок NAL с VCL кодированным изображением и предшествующий первый блок NAL с VCL другого кодированного изображения.

RBSP набора параметров последовательности включает параметры, которые можно считать одной или более RBSP набора параметров изображения или одним или более блоками NAL SEI, содержащих сообщение SEI о периоде буферизации. Каждая RBSP набора параметров последовательности в исходном положении считается не действующей в начале операции процесса декодирования. Принято, что не более одной RBSP набора параметров изображения можно считать активной в любой момент во время операции процесса декодирования, а активация любой конкретной RBSP набора параметров изображения приводит к деактивации ранее действующей RBSP набора параметров изображения (если имеется).

Если RBSP набора параметров последовательности (с конкретным значением `seq_parameter_set_id`) уже не действует и считается активированной RBSP набора параметров изображения (при использовании значения `seq_parameter_set_id`) или считается блоком NAL SEI, содержащим сообщение SEI о периоде буферизации (при использовании этого же значения `seq_parameter_set_id`), то она активирована. Такую RBSP набора параметров последовательности называют активной RBSP набора параметров последовательности до тех пор, пока она не будет деактивирована другой RBSP набора параметров последовательности. RBSP набора параметров Последовательности с этим конкретным значением `seq_parameter_set_id` должна быть доступна процессу декодирования до ее активации. Активированная RBSP набора параметров последовательности должна оставаться активной для всей кодированной видеопоследовательности.

ПРИМЕЧАНИЕ 2. – Вследствие того, что блок доступа IDR начинает новую кодированную видеопоследовательность, а активированная RBSP набора параметров последовательности должна оставаться активной для всей кодированной видеопоследовательности, RBSP набора параметров последовательности может быть активирована только сообщением SEI о периоде буферизации, если сообщение SEI о периоде буферизации является частью блока доступа IDR.

Любой блок NAL набора параметров последовательности со значением `seq_parameter_set_id` для активной RBSP набора параметров последовательности должен иметь то же содержание, что и активная RBSP набора параметров последовательности, если он не следует за последним блоком доступа кодированной видеопоследовательности и не предшествует первому блоку NAL с VCL и первому блоку NAL SEI с сообщением SEI о периоде буферизации (если таковое имеется) другой кодированной видеопоследовательности.

ПРИМЕЧАНИЕ 3. – Если RBSP набора параметров изображения или RBSP набора параметров последовательности передают в потоке битов, эти ограничения накладывают ограничение на блоки NAL, которые содержат, соответственно, RBSP набора параметров изображения или RBSP набора параметров последовательности. Иначе (RBSP набора параметров изображения или RBSP набора параметров последовательности передают другими средствами, не описанными в настоящей Рекомендации | Международном стандарте) они должны быть доступны процессу декодирования вовремя, так чтобы эти ограничения выполнялись.

RBSP расширения набора параметров последовательности, если присутствует, включает параметры, выполняющие те же функции, что и RBSP набора параметров последовательности. С тем чтобы установить ограничения в отношении элементов синтаксиса RBSP расширения набора параметров последовательности и определить активацию RBSP расширения набора параметров последовательности, RBSP расширения набора параметров последовательности должна рассматриваться как часть предшествующей RBSP набора параметров последовательности с тем же значением `seq_parameter_set_id`. Если присутствует RBSP набора параметров последовательности, за которой до активации RBSP набора параметров последовательности не следует RBSP расширения набора параметров последовательности с тем же значением `seq_parameter_set_id`, RBSP расширения набора параметров последовательности и ее элементы синтаксиса должны рассматриваться как не присутствующие для активной RBSP набора параметров последовательности.

Все ограничения, которые выражают взаимодействие между значениями элементов синтаксиса (и значения переменных, полученных из этих элементов синтаксиса) в наборе параметров последовательности, наборе параметров изображения и другие элементы синтаксиса выражают ограничения, которые наложены только на активный набор параметров последовательности и на активный набор параметров изображения. Если представлена любая, не активированная в потоке битов RBSP набора параметров последовательности, ее элементы синтаксиса должны иметь значения, которые должны соответствовать определенным ограничениям в случае, если активация сделана с помощью контроля потока битов, подчиняющегося другим требованиям соответствия. Если представлена любая, ранее не активированная в потоке битов RBSP набора параметров изображения, ее элементы синтаксиса должны иметь значения, которые должны соответствовать определенным ограничениям в случае, если активация сделана с помощью контроля потока битов, подчиняющегося другим требованиям соответствия.

Во время операции процесса декодирования (см. раздел 8) должны учитываться значения активного набора параметров изображения и активного набора параметров последовательности. Для интерпретации сообщений SEI значения параметров набора параметров изображения и набора параметров последовательности, которые действуют в процессе декодирования для блоков NAL с VCL первично кодированного изображения в том же блоке доступа, должны учитываться, пока иное не будет определено в семантике сообщения SEI.

7.4.1.2.2 Порядок блоков доступа и связь с кодированными видеопоследовательностями

Поток битов, соответствующий настоящей Рекомендации | Международному стандарту, состоит из одной или более кодированных видеопоследовательностей.

Кодированная видеопоследовательность состоит из одного или более блоков доступа. Порядок блоков NAL и кодированных изображений и их связь с блоками доступа описаны в п. 7.4.1.2.3.

Первый блок доступа каждой кодированной видеопоследовательности – это блок доступа IDR. Все последующие блоки доступа в кодированной видеопоследовательности – это не блоки доступа IDR.

Значения порядка счета изображений для кодированных изображений в последующих блоках доступа в порядке декодирования, содержащего неконтрольные изображения, должны быть неубывающими.

Если такой присутствует, то блок доступа, следующий за блоком доступа, который содержит конечную последовательность блоков NAL, должен быть блоком доступа IDR.

Если блок NAL с SEI содержит данные, которые принадлежат более чем одному блоку доступа (например, если блок NAL с SEI предназначен для кодированной видеопоследовательности), этот блок должен находиться в первом блоке доступа, в котором он будет использован.

Если в блоке доступа представлен конец потока блоков NAL, то этот блок должен быть последним блоком доступа в потоке битов, а концом потока блоков NAL должен быть последний блок NAL в этом блоке доступа.

7.4.1.2.3 Порядок блоков NAL и кодированных изображений и их связь с блоками доступа

Блок доступа состоит из одного первично кодированного изображения, нуля или более соответствующих избыточно кодированных изображений и нуля или более блоков NAL без VCL. Связь блоков NAL VCL с первично или избыточно кодированными изображениями описана в п. 7.4.1.2.5.

Первый блок доступа в потоке битов начинается с первого блока NAL этого потока битов.

Первый из любых, следующих за блоками NAL (за последним блоком NAL VCL) первично кодированных изображений, определяет характеристики начала нового блока доступа:

- разграничитель блок доступа блока NAL (если имеется);
- установку параметра последовательности блока NAL (если имеется);
- установку параметра изображения блока NAL (если имеется);
- блок NAL с SEI (если имеется);
- блоки NAL с `nal_unit_type` в диапазоне от 14 до 18 включительно;
- первый блок NAL VCL первично кодированного изображения (имеется всегда).

Ограничения на обнаружение первого блока NAL VCL первично кодированного изображения описаны в п. 7.4.1.2.4.

Следующие ограничения должны подчиняться в блоке доступа порядку кодированных изображений и блоков NAL без VCL.

- Если имеется разграничитель блока доступа блока NAL, он должен быть первым блоком NAL. В каждом блоке доступа должно быть не более одного разграничителя блока доступа блока NAL.
- Если имеются любые блоки NAL с SEI, они должны предшествовать первично кодированному изображению.
- Если имеется блок NAL с SEI, содержащий сообщение SEI о периоде буферизации, это сообщение должно быть первой полезной нагрузкой сообщения SEI первого блока NAL SEI в блоке доступа.
- Первично кодированному изображению должны предшествовать соответствующие избыточно кодированные изображения.
- Если имеются дополнительные кодированные изображения, они должны располагаться в возрастающем порядке значения с `redundant_pic_cnt`.
- Если представлен блок NAL расширения набора параметров последовательности, он должен быть следующим блоком NAL после блока NAL набора параметров последовательности, который имеет то же значение `seq_parameter_set_id`, что и блок NAL расширения набора параметров последовательности.
- Если представлена одна или более кодированных секций дополнительно кодированного изображения без блоков NAL разделения, они должны следовать за первично кодированным изображением и всеми избыточно кодированными изображениями (если таковые имеются).
- Если представлен конец последовательности блоков NAL, за ним должно следовать первично кодированное изображение и все дополнительно кодированные изображения (если таковые имеются).
- Если представлен конец потока блоков NAL, то это должен быть последний блок NAL.
- Блоки NAL с `nal_unit_type`, равным 0, 12 или в диапазоне от 19 до 31 включительно, не должны предшествовать первому блоку NAL VCL первично кодированного изображения.

ПРИМЕЧАНИЕ 1. – Набор параметров последовательности блоков NAL или набор параметров изображения блоков NAL могут быть представлены в блоке доступа, но не могут следовать за последним блоком NAL VCL первично кодированного изображения в блоке доступа, поскольку это условие должно указывать на начало нового блока доступа.

ПРИМЕЧАНИЕ 2. – Если в блоке доступа имеется блок NAL с `nal_unit_type`, равным 7 или 8, он может быть или не быть отнесен к кодированным изображениям блока доступа, в котором он представлен, а может быть отнесен к кодированным изображениям последующих блоков доступа.

Структура блоков доступа, не содержащих никаких блоков NAL со значениями `nal_unit_type`, равными 0, 7, 8 или лежащими в диапазоне от 12 до 18 включительно или в диапазоне от 20 до 31 включительно, показана на рисунке 7-1.

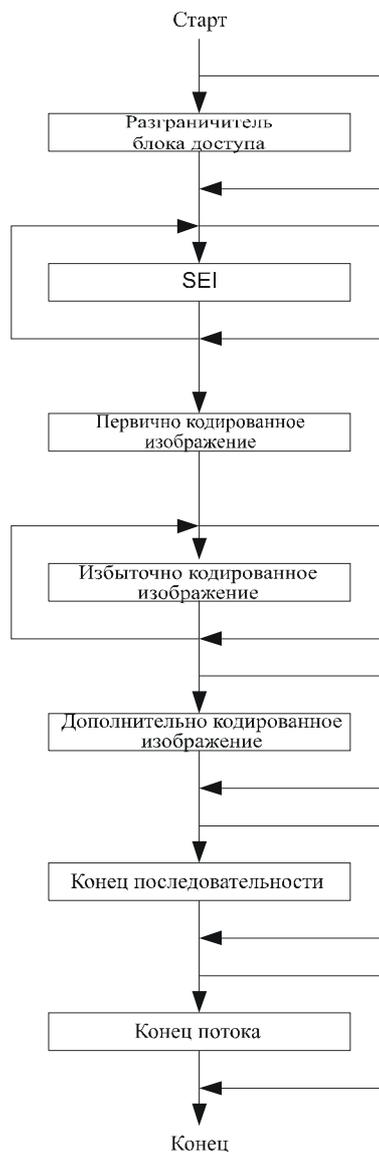


Рисунок 7-1 – Структура блока доступа, не содержащего никаких блоков NAL со значениями `nal_unit_type`, равными 0, 7, 8 или лежащими в диапазоне от 12 до 18 включительно или в диапазоне от 20 до 31 включительно

7.4.1.2.4 Обнаружение первого блока NAL VCL первично кодированного изображения

В этом подразделе описаны ограничения на синтаксис блока NAL VCL, которые достаточны, чтобы обеспечить обнаружение первого блока NAL VCL каждого раннее кодированного изображения.

Любой блок NAL кодированной секции или блок NAL кодированной секции с разделенными данными типа A первично кодированного изображения текущего блока доступа следует отличить от любого блока NAL кодированной секции или блока NAL кодированной секции с разделенными данными первично кодированного изображения предыдущего блока доступа одним или некоторыми из следующих способов:

- `frame_num` различается по значению. Значение `frame_num` используют для проверки, что это условие является тем значением `frame_num`, которое появляется в синтаксисе заголовка секции независимо от того, считают ли это значение равным 0 для последующего использования в процессе декодирования вследствие присутствия равного 5 значения `memory_management_control_operation`;
 ПРИМЕЧАНИЕ 1. – Последствие вышеприведенного утверждения заключается в том, что первично кодированное изображение с `frame_num`, равным 1, не может содержать `memory_management_control_operation`, равным 5, пока не будут выполнены некоторые другие перечисленные ниже условия для следующего первично кодированного изображения, которое следует за этим (если последует).
- `pic_parameter_set_id` отличается по величине;
- `field_pic_flag` отличается по величине;

- `bottom_field_flag` присутствует в обоих и отличается по величине;
- `nal_ref_idc` отличается по величине от `nal_ref_idc`, равного 0;
- `pic_order_cnt_type` равно 0 для обоих, и каждое значение `pic_order_cnt_lsb` отличается по величине либо отличается по величине значение `delta_pic_order_cnt_bottom`;
- `pic_order_cnt_type` равно 1 для обоих, и каждое значение `delta_pic_order_cnt[0]` отличается по величине либо отличается по величине `delta_pic_order_cnt[1]`;
- `nal_unit_type` отличается по величине от значения `nal_unit_type`, равного 5;
- `nal_unit_type` равно 5 для обоих и отличается по величине от `idr_pic_id`.

ПРИМЕЧАНИЕ 2. – Некоторые блоки NAL VCL в дополнительно кодированных изображениях или некоторые блоки NAL без VCL (например, разграничитель блока NAL блока доступа) можно также использовать для обнаружения границ между блоками доступа, и поэтому эти блоки могут помогать обнаруживать начало нового первично кодированного изображения.

7.4.1.2.5 Порядок блоков NAL VCL и связь с кодированными изображениями

Каждый блок NAL VCL – это часть кодированного изображения.

Порядок блоков NAL VCL в кодированном IDR изображении ограничен следующим образом.

- Если допустим произвольный порядок секций, как это определено в Приложении А, то кодированные секции блоков NAL IDR изображения могут иметь любой порядок относительно друг друга.
- Иначе (произвольный порядок секций не допустим) кодированные секции блоков NAL IDR изображения должны располагаться в возрастающем порядке адреса макроблока для первого макроблока каждой кодированной секции блока NAL IDR изображения.

Порядок блоков NAL VCL в кодированном не IDR изображении ограничен следующим образом.

- Если допустим произвольный порядок секций, как это определено в Приложении А, то кодированные секции блоков NAL не IDR изображения или А кодированные секции блоков NAL с разделенными данными типа А могут иметь любой порядок относительно друг друга. Кодированная секция блоков NAL разделенных данных А с конкретным значением `slice_id` должна предшествовать любой представленной кодированной секции блоков NAL разделенных данных В с тем же самым значением `slice_id`. Кодированная секция блоков NAL разделенных данных А с конкретным значением `slice_id` должна предшествовать любой представленной кодированной секции блоков NAL разделенных данных С с тем же самым значением `slice_id`. Если представлена кодированная секция блоков NAL разделенных данных В с конкретным значением `slice_id`, то она должна предшествовать любой представленной кодированной секции блока NAL разделенных данных С с тем же самым значением `slice_id`.
- Иначе (произвольный порядок секций не допустим) кодированные секции блоков NAL не IDR изображения или кодированные секции блоков NAL разделенных данных А должны располагаться в возрастающем порядке адреса макроблока для первого макроблока каждой кодированной секции блоков NAL не IDR изображения или кодированной секции блоков NAL разделенных данных А. Кодированная секция блоков NAL разделенных данных А с конкретным значением `slice_id` должна непосредственно предшествовать любой представленной кодированной секции блоков NAL разделенных данных В с тем же самым значением `slice_id`. Кодированная секция блоков NAL разделенных данных А с конкретным значением `slice_id` должна непосредственно предшествовать любой представленной кодированной секции блоков NAL разделенных данных С с тем же самым значением `slice_id`, если не представлена кодированная секции блоков NAL разделенных данных В с тем же самым значением `slice_id`. Если представлена кодированная секции блоков NAL разделенных данных В с тем же самым значением `slice_id`, она должна непосредственно предшествовать любой представленной кодированной секции блоков NAL разделенных данных С с тем же самым значением `slice_id`.

Блоки NAL с `nal_unit_type`, равным 12, могут присутствовать в блоке доступа, но не должны предшествовать первому блоку NAL VCL первично кодированного изображения в блоке доступа.

Блоки NAL с `nal_unit_type`, равным 0 или в диапазоне от 24 до 31 включительно, которые не определены, могут присутствовать в блоке доступа, но не должны предшествовать первому блоку NAL VCL первично кодированного изображения в блоке доступа.

Блоки NAL с `nal_unit_type` в диапазоне от 19 до 23 включительно, которые были зарезервированы, не должны предшествовать первому блоку NAL VCL первично кодированного изображения в блоке доступа (если они описаны МСЭ-Т | ИСО/МЭК для будущих применений).

7.4.2 Полезная нагрузка последовательности исходных байтов и семантика концевых битов RBSP

7.4.2.1 Семантика RBSP набора параметров последовательности

`profile_idc` и `level_idc` указывают профиль и уровень, которым соответствует поток битов, как это определено в Приложении А.

`constraint_set0_flag`, равный 1, указывает, что поток битов подчиняется всем ограничениям, описанным в п. А.2.1. `constraint_set0_flag`, равное 0, указывает, что поток битов может подчиняться или не подчиняться всем ограничениям, описанным в п. А.2.1.

constraint_set1_flag, равный 1, указывает, что поток битов подчиняется всем ограничениям, описанным в п. А.2.2. **constraint_set1_flag**, равное 0, указывает, что поток битов может подчиняться или не подчиняться всем ограничениям, описанным в п. А.2.2.

constraint_set2_flag, равный 1, указывает, что поток битов подчиняется всем ограничениям, описанным в п. А.2.3. **constraint_set2_flag**, равное 0, указывает, что поток битов может подчиняться или не подчиняться всем ограничениям, описанным в п. А.2.3.

ПРИМЕЧАНИЕ 1. – Если один или более элементов **constraint_set0_flag**, **constraint_set1_flag** или **constraint_set2_flag** равны 1, поток битов должен подчиняться ограничениям всех указанных подпунктов в п. А.2. Если **profile_idc** равно 100, 110, 122 или 144, значениями **constraint_set0_flag**, **constraint_set1_flag**, и **constraint_set2_flag** должен быть 0.

constraint_set3_flag указывает на следующее.

- Если **profile_idc** равно 66, 77 или 88, а **level_idc** равно 11, **constraint_set3_flag**, равный 1, указывает, что на поток битов действуют все ограничения, определенные в Приложении А для уровня 1b, а **constraint_set3_flag**, равный 0 указывает, что на поток битов могут действовать или могут не действовать все ограничения, определенные в Приложении А для уровня 1b.
- Иначе (**profile_idc** равно 100, 110, 122 или 144 или **level_idc** не равно 11) значение 1 для **constraint_set3_flag** резервируется для будущего использования МСЭ-Т | ИСО/МЭК. В потоках битов, соответствующих настоящей Рекомендации | Международному стандарту, значением **constraint_set3_flag** должен быть 0, если **profile_idc** равно 100, 110, 122 или 144 или **level_idc** не равно 11. Декодеры, соответствующие настоящей Рекомендации | Международному стандарту, должны игнорировать значение **constraint_set3_flag**, если **profile_idc** равно 100, 110, 122 или 144 или **level_idc** не равно 11.

reserved_zero_4bits должно быть равно 0. Прочие значения **reserved_zero_4bits** могут быть в дальнейшем определены МСЭ-Т | ИСО/МЭК. Декодеры должны игнорировать значение **reserved_zero_4bits**.

seq_parameter_set_id указывает набор параметров последовательности, на которую ссылается набор параметров изображения. Значение **seq_parameter_set_id** должно находиться в диапазоне от 0 до 31 включительно.

ПРИМЕЧАНИЕ 2. – В случае реализуемости кодеры должны использовать различающиеся значения **seq_parameter_set_id**, если значения других элементов синтаксиса набора параметров последовательности отличаются больше, чем изменяющиеся значения элементов синтаксиса, связанных с особым значением **seq_parameter_set_id**.

chroma_format_idc определяет выборку цветности относительно выборки яркости согласно п. 6.2. Значение **chroma_format_idc** должно находиться в диапазоне от 0 до 3 включительно. Если **chroma_format_idc** не присутствует, его значение должно приниматься равным 1 (формат цветности 4:2:0).

residual_colour_transform_flag, равное 1, указывает, что применяется преобразование цвета остатка, описанное в п. 8.5. **residual_colour_transform_flag**, равное 0, указывает, что преобразование цвета остатка не применяется. Если **residual_colour_transform_flag** не представлен, его значение должно приниматься равным 0.

bit_depth_luma_minus8 определяет глубину цвета образцов массива яркости и значение сдвига диапазона параметров квантования яркости $QpBdOffset_Y$, которые определяются следующим образом:

$$BitDepth_Y = 8 + bit_depth_luma_minus8; \quad (7-1)$$

$$QpBdOffset_Y = 6 * bit_depth_luma_minus8. \quad (7-2)$$

Если **bit_depth_luma_minus8** не представлено, его значение должно приниматься равным 0. Значение **bit_depth_luma_minus8** должно находиться в диапазоне от 0 до 4 включительно.

bit_depth_chroma_minus8 определяет глубину цвета образцов массива цветности и значение сдвига диапазона параметров квантования $QpBdOffset_C$, которые определяются следующим образом:

$$BitDepth_C = 8 + bit_depth_chroma_minus8; \quad (7-3)$$

$$QpBdOffset_C = 6 * (bit_depth_chroma_minus8 + residual_colour_transform_flag). \quad (7-4)$$

Если **bit_depth_chroma_minus8** не представлено, его значение должно приниматься равным 0. Значение **bit_depth_chroma_minus8** должно находиться в диапазоне от 0 до 4 включительно.

Переменную **RawMbBits** находят следующим образом:

$$RawMbBits = 256 * BitDepth_Y + 2 * MbWidth_C * MbHeight_C * BitDepth_C. \quad (7-5)$$

qpprime_y_zero_transform_bypass_flag, равное 1, указывает, что если QP'_{γ} равно 0, для процесса декодирования коэффициентов преобразования и процесса построения изображения до процесса фильтра устранения блочности должны применяться операции с обходом преобразования согласно п. 8.5. **qpprime_y_zero_transform_bypass_flag**, равное 0, указывает, что процесс декодирования коэффициентов преобразования и процесс построения изображения до процесса фильтра устранения блочности не должны использовать операции с обходом преобразования. Если **qpprime_y_zero_transform_bypass_flag** не представлено, его значение должно приниматься равным 0.

seq_scaling_matrix_present_flag, равное 1, указывает, что представлены флаги **seq_scaling_list_present_flag[i]** для $i = 0..7$. **seq_scaling_matrix_present_flag**, равное 0, указывает, что эти флаги не представлены и что значение списка масштабирования уровней последовательности, определяемое значением **Flat_4x4_16**, должно приниматься для $i = 0..5$, а значение списка масштабирования уровней последовательности, определяемое значением **Flat_8x8_16**, должно приниматься для $i = 6..7$. Если флаг **seq_scaling_matrix_present_flag** не представлен, его значение должно приниматься равным 0.

Списки масштабирования **Flat_4x4_16** и **Flat_8x8_16** находят следующим образом:

$$\text{Flat_4x4_16}[i] = 16 \quad \text{при } i = 0..15; \quad (7-6)$$

$$\text{Flat_8x8_16}[i] = 16 \quad \text{при } i = 0..63. \quad (7-7)$$

seq_scaling_list_present_flag[i], равное 1, указывает, что синтаксическая структура для списка масштабирования i представлена в наборе параметров последовательности. **seq_scaling_list_present_flag[i]**, равное 0, указывает, что синтаксическая структура для списка масштабирования i не представлена в наборе параметров последовательности и что для вывода списка масштабирования уровней последовательности для индекса i должен использоваться набор правил обращения к списку масштабирования **A**, определенный в таблице 7-2.

Таблица 7-2 – Присвоение мнемонических имен индексам списка масштабирования и спецификация правила обращения

Значение индекса списка масштабирования	Мнемоническое имя	Размер блока	Тип предсказания МВ	Компонент	Набор правил обращения к списку масштабирования А	Набор правил обращения к списку масштабирования В	Список масштабирования по умолчанию
0	SI_4x4_Intra_Y	4x4	Intra	Y	Список масштабирования по умолчанию	Список масштабирования уровня последовательности	Default_4x4_Intra
1	SI_4x4_Intra_Cb	4x4	Intra	Cb	Список масштабирования для $i = 0$	Список масштабирования для $i = 0$	Default_4x4_Intra
2	SI_4x4_Intra_Cr	4x4	Intra	Cr	Список масштабирования для $i = 1$	Список масштабирования для $i = 1$	Default_4x4_Intra
3	SI_4x4_Inter_Y	4x4	Inter	Y	Список масштабирования по умолчанию	Список масштабирования уровня последовательности	Default_4x4_Inter
4	SI_4x4_Inter_Cb	4x4	Inter	Cb	Список масштабирования для $i = 3$	Список масштабирования для $i = 3$	Default_4x4_Inter
5	SI_4x4_Inter_Cr	4x4	Inter	Cr	Список масштабирования для $i = 4$	Список масштабирования для $i = 4$	Default_4x4_Inter
6	SI_8x8_Intra_Y	8x8	Intra	Y	Список масштабирования по умолчанию	Список масштабирования уровня последовательности	Default_8x8_Intra
7	SI_8x8_Inter_Y	8x8	Inter	Y	Список масштабирования по умолчанию	Список масштабирования уровня последовательности	Default_8x8_Inter

В таблице 7-3 определяются списки масштабирования по умолчанию **Default_4x4_Intra** и **Default_4x4_Inter**. В таблице 7-4 определяются списки масштабирования по умолчанию **Default_8x8_Intra** и **Default_8x8_Inter**.

Таблица 7-3 – Спецификация списков масштабирования по умолчанию
Default_4x4_Intra и Default_4x4_Inter

Idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Default_4x4_Intra[idx]	6	13	13	20	20	20	28	28	28	28	32	32	32	37	37	42
Default_4x4_Inter[idx]	10	14	14	20	20	20	24	24	24	24	27	27	27	30	30	34

Таблица 7-4 – Спецификация списков масштабирования по умолчанию
Default_8x8_Intra и Default_8x8_Inter

Idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Default_8x8_Intra[idx]	6	10	10	13	11	13	16	16	16	16	18	18	18	18	18	23
Default_8x8_Inter[idx]	9	13	13	15	13	15	17	17	17	17	19	19	19	19	19	21

Таблица 7-4 (продолжение) – Спецификация списков масштабирования по умолчанию
Default_8x8_Intra и Default_8x8_Inter

Idx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Default_8x8_Intra[idx]	23	23	23	23	23	25	25	25	25	25	25	25	27	27	27	27
Default_8x8_Inter[idx]	21	21	21	21	21	22	22	22	22	22	22	22	24	24	24	24

Таблица 7-4 (продолжение) – Спецификация списков масштабирования по умолчанию
Default_8x8_Intra и Default_8x8_Inter

Idx	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Default_8x8_Intra[idx]	27	27	27	27	29	29	29	29	29	29	29	31	31	31	31	31
Default_8x8_Inter[idx]	24	24	24	24	25	25	25	25	25	25	25	27	27	27	27	27

Таблица 7-4 (продолжение) – Спецификация списков масштабирования по умолчанию
Default_8x8_Intra и Default_8x8_Inter

Idx	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Default_8x8_Intra[idx]	31	33	33	33	33	33	36	36	36	36	38	38	38	40	40	42
Default_8x8_Inter[idx]	27	28	28	28	28	28	30	30	30	30	32	32	32	33	33	35

$\log_2 \text{max_frame_num_minus4}$ определяет значение переменной MaxFrameNum, которую используют в кадре frame_numrelated, полученных следующим образом:

$$\text{MaxFrameNum} = 2^{(\log_2 \text{max_frame_num_minus4} + 4)} \quad (7-8)$$

Значение $\log_2 \text{max_frame_num_minus4}$ должно быть в диапазоне от 0 до 12 включительно.

pic_order_cnt_type определяет метод вычисления порядка декодирования изображения (как описано в п. 8.2.1). Значение pic_order_cnt_type должно быть в диапазоне от 0 до 2 включительно.

pic_order_cnt_type не должно быть равно 2 в закодированной видеопоследовательности, в которой содержится любое из следующих:

- блок доступа с неконтрольным кадром, за которым сразу следует блок доступа с неконтрольным изображением;
- два блока доступа, каждый содержащий поле с двумя полями, вместе формирующее пару дополнительных неконтрольных полей, за которыми сразу следует блок доступа с неконтрольным изображением;
- блок доступа с неконтрольным полем, за которым сразу следует блок доступа с другим неконтрольным изображением, которое не формирует пару дополнительных неконтрольных полей с первым из двух блоков доступа.

log2_max_pic_order_cnt_lsb_minus4 определяет значение переменной **MaxPicOrderCntLsb**, которую используют в процессе вычисления порядка декодирования изображения, как описано в п. 8.2.1, следующим образом:

$$\text{MaxPicOrderCntLsb} = 2^{(\text{log2_max_pic_order_cnt_lsb_minus4} + 4)} \quad (7-9)$$

Значение **log2_max_pic_order_cnt_lsb_minus4** должно быть в диапазоне от 0 до 12 включительно.

delta_pic_order_always_zero_flag, равный 1, определяет, что **delta_pic_order_cnt[0]** и **delta_pic_order_cnt[1]** отсутствуют в заголовках секции последовательности и должны считаться равными 0. **delta_pic_order_always_zero_flag**, равный 0, определяет, что **delta_pic_order_cnt[0]** присутствует в заголовках секции последовательности, а **delta_pic_order_cnt[1]** может находиться в заголовках секции последовательности.

offset_for_non_ref_pic используют для вычисления в порядке изображения неконтрольного изображения, как описано в п. 8.2.1. Значение **offset_for_non_ref_pic** должно быть в диапазоне от 2^{31} до $2^{31} - 1$ включительно.

offset_for_top_to_bottom_field используют для вычисления в порядке изображения нижнего поля, как описано в п. 8.2.1. Значение **offset_for_top_to_bottom_field** должно быть в диапазоне от -2^{31} до $2^{31} - 1$ включительно.

num_ref_frames_in_pic_order_cnt_cycle используют в процессе декодирования для учета порядка изображений, как описано в п. 8.2.1. Значение **num_ref_frames_in_pic_order_cnt_cycle** должно быть в диапазоне от 0 до 255 включительно.

offset_for_ref_frame[i] – это элемент списка значений **num_ref_frames_in_pic_order_cnt_cycle**, использованных в процессе декодирования для учета порядка изображений, как описано в п. 8.2.1. Значение **offset_for_ref_frame[i]** должно быть в диапазоне от -2^{31} до $2^{31} - 1$ включительно.

num_ref_frames определяет максимальное число сокращенных и удлиненных контрольных кадров, дополняющих пар контрольных полей и непарных(не сдвоенных) контрольных полей, которые могут быть использованы процессом декодирования для внешнего предсказания любого изображения в последовательности. **num_ref_frames** также определяет размер действия скользящего окна, как описано в п. 8.2.5.3. Значение **num_ref_frames** должно быть в диапазоне от 0 до **MaxDpbSize** включительно (как описано в п. А.3.1 или А.3.2).

gaps_in_frame_num_value_allowed_flag определяет разрешенные значения **frame_num**, как описано в п. 7.4.3, и процесс декодирования в случае предполагаемого расхождения между значениями **frame_num**, как описано в п. 8.2.5.2.

pic_width_in_mbs_minus1 плюс 1 определяет ширину каждого декодированного изображения в блоках макроблоков.

Переменную ширины изображения в блоках макроблоков находят следующим образом:

$$\text{PicWidthInMbs} = \text{pic_width_in_mbs_minus1} + 1. \quad (7-10)$$

Переменную ширины изображения компонента яркости находят следующим образом:

$$\text{PicWidthInSamplesL} = \text{PicWidthInMbs} * 16. \quad (7-11)$$

Переменную ширины изображения компонентов цветности находят следующим образом:

$$\text{PicWidthInSamplesC} = \text{PicWidthInMbs} * \text{MbWidthC}. \quad (7-12)$$

pic_height_in_map_units_minus1 плюс 1 определяет высоту блоков секции отображения группы декодированного кадра или поля.

Переменные **PicHeightInMapUnits** и **PicSizeInMapUnits** находят следующим образом:

$$\text{PicHeightInMapUnits} = \text{pic_height_in_map_units_minus1} + 1; \quad (7-13)$$

$$\text{PicSizeInMapUnits} = \text{PicWidthInMbs} * \text{PicHeightInMapUnits}. \quad (7-14)$$

frame_mbs_only_flag, равный 0, определяет, что кодированные изображения кодированной видеопоследовательности могут быть либо кодированными полями, либо кодированными кадрами. **frame_mbs_only_flag**, равный 1, определяет, что каждое кодированное изображение кодированной видеопоследовательности – это кодированный кадр, содержащий только макроблоки кадра.

Разрешенный диапазон значений для **pic_width_in_mbs_minus1**, **pic_hieght_in_map_units_minus1** и **frame_mbs_only_flag** описан в ограничениях Приложения А.

В зависимости от значения `frame_mbs_only_flag` значения семантики `pic_hieght_in_map_units_minus1` присваивают следующим образом.

- Если `frame_mbs_only_flag` равно 0, то `pic_hieght_in_map_units_minus1` плюс 1 – это высота поля в блоках макроблоков.
- Иначе (`frame_mbs_only_flag` равно 1) `pic_hieght_in_map_units_minus1` плюс 1 – это высота кадра в блоках макроблоков.

Переменную `FrameHeightInMbs` находят следующим образом:

$$\text{FrameHeightInMbs} = (2 - \text{frame_mbs_only_flag}) * \text{PicHeightInMapUnits}. \quad (7-15)$$

`mb_adaptive_frame_field_flag`, равное 0, определяет отсутствие переключения между кадром и полем макроблоков в изображении. **`mb_adaptive_frame_field_flag`**, равное 1, определяет возможное переключение между макроблоками кадра и поля в кадрах. Если **`mb_adaptive_frame_field_flag`** отсутствует, его считают равным 0.

`direct_8x8_inference_flag` определяет метод, использованный в процессе определения векторов движения яркости для значений `B_Skip`, `B_Direct_16x16` и `B_Direct_8x8`, как описано в п. 8.4.1.2. Если `frame_mbs_only_flag` равно 0, **`direct_8x8_inference_flag`** должен быть равен 1.

`frame_cropping_flag`, равное 1, определяет, что параметры сдвига обрезки (изменения масштаба) кадра поступают после следующего набора параметров последовательности. **`frame_cropping_flag`**, равный 0, определяет, что параметры сдвига обрезки отсутствуют.

`frame_crop_left_offset`, **`frame_crop_right_offset`**, **`frame_crop_top_offset`**, **`frame_crop_bottom_offset`** определяют образцы изображений в кодированной видеопоследовательности, которые являются выходом процесса декодирования, как область прямоугольника, описанную в координатах кадра для выхода.

Переменные `CropUnitX` и `CropUnitY` находят следующим образом.

- Если `chroma_format_idc` равно 0, `CropUnitX` и `CropUnitY` определяют как:

$$\text{CropUnitX} = 1; \quad (7-16)$$

$$\text{CropUnitY} = 2 - \text{frame_mbs_only_flag}. \quad (7-17)$$

- Иначе (`chroma_format_idc` равно 1, 2 или 3) `CropUnitX` и `CropUnitY` определяют как:

$$\text{CropUnitX} = \text{SubWidthC}; \quad (7-18)$$

$$\text{CropUnitY} = \text{SubHeightC} * (2 - \text{frame_mbs_only_flag}). \quad (7-19)$$

Кадрирующий прямоугольник кадра содержит образцы яркости с горизонтальными координатами кадра от `CropUnitX * frame_crop_left_offset` до `PicWidthInSamplesL - (CropUnitX * frame_crop_right_offset + 1)` и вертикальными координатами кадрами от `CropUnitY * frame_crop_top_offset` до `(16 * FrameHeightInMbs) - (CropUnitY * frame_crop_bottom_offset + 1)` включительно. Значение `frame_crop_left_offset` должно быть в диапазоне от 0 до `(PicWidthInSamplesL / CropUnitX) - (frame_crop_right_offset + 1)` включительно; значение `frame_crop_top_offset` должно быть в диапазоне от 0 до `(16 * FrameHeightInMbs / CropUnitY) - (frame_crop_bottom_offset + 1)` включительно.

Если `frame_cropping_flag` равно 0, значения: `frame_crop_left_offset`, `frame_crop_right_offset`, `frame_crop_top_offset` и `frame_crop_bottom_offset` должны приниматься равными 0.

Если `chroma_format_idc` не равно 0, соответствующие определенные образцы двух массивов цветности – это образцы с координатами кадра (`x / SubWidthC`, `y / SubHeightC`), где (`x`, `y`) – координаты кадра определенных образцов яркости.

Для декодированных полей описанные образцы декодированных полей являются такими, которые попадают в прямоугольник, определенный в координатах кадра.

`vui_parameters_present_flag`, равное 1, определяет, что структура синтаксиса `vui_parameters()`, описанная в Приложении E, представлена. **`vui_parameters_present_flag`**, равное 0, определяет, что структура синтаксиса `vui_parameters()`, описанная в Приложении E, не представлена.

7.4.2.1.1 Семантика списка масштабирования

`delta_scale` используется для определения `j`-го элемента списка масштабирования для `j` в диапазоне от 0 до `sizeofScalingList - 1` включительно. Значение `delta_scale` должно находиться в диапазоне от -128 до +127 включительно.

Если `useDefaultScalingMatrixFlag` определяется равным 1, в качестве списка масштабирования должен приниматься список масштабирования по умолчанию, определенный в таблице 7-2.

7.4.2.1.2 Семантика RBSP расширения набора параметров последовательности

seq_parameter_set_id определяет набор параметров последовательности, связанный с расширением набора параметров последовательности. Значение **seq_parameter_set_id** должно находиться в диапазоне от 0 до 31 включительно.

aux_format_idc, равное 0, указывает, что в кодированной видеопоследовательности отсутствуют дополнительно кодированные изображения. **aux_format_idc**, равное 1, указывает, что в каждом блоке доступа данной видеопоследовательности представлено точно одно дополнительно кодированное изображение и что для целей альфа-смешения декодированные образцы связанного первично кодированного изображения в каждом блоке доступа должны умножаться на значения образцов интерпретации дополнительно кодированного изображения в блоке доступа в процессе отображения после выхода из процесса декодирования. **aux_format_idc**, равное 2, указывает, что в каждом блоке доступа данной видеопоследовательности представлено точно одно дополнительно кодированное изображение и что для целей альфа-смешения декодированные образцы связанного первично кодированного изображения в каждом блоке доступа не должны умножаться на значения образцов интерпретации дополнительно кодированного изображения в блоке доступа в процессе отображения после выхода из процесса декодирования. **aux_format_idc**, равное 3, указывает, что в каждом блоке доступа данной видеопоследовательности представлено точно одно дополнительно кодированное изображение и что использование дополнительно кодированных изображений не определено. Значение **aux_format_idc** должно находиться в диапазоне от 0 до 3 включительно. Значения, превышающие 3, зарезервированы для **aux_format_idc** с целью указания наличия точно одного дополнительно кодированного изображения в каждом блоке доступа кодированной видеопоследовательности для целей, которые в дальнейшем будут определены МСЭ-Т | ИСО/МЭК. Если **aux_format_idc** не представлено, его значение должно приниматься равным 0.

ПРИМЕЧАНИЕ 1. – Не требуется, чтобы декодеры, соответствующие настоящей Рекомендации | Международному стандарту, декодировали дополнительно кодированные изображения.

bit_depth_aux_minus8 определяет глубину цвета образцов массива образцов дополнительно кодированного изображения. Значение **bit_depth_aux_minus8** должно находиться в диапазоне от 0 до 4 включительно.

alpha_incr_flag, равное 0, указывает, что значение образца интерпретации для каждого значения декодированного образца дополнительно кодированного изображения равно значению декодированного образца дополнительно кодированного изображения для целей альфа-смешения. **alpha_incr_flag**, равное 1, указывает, что для целей альфа-смешения после декодирования образцов дополнительно кодированного изображения любое значение образца дополнительно кодированного изображения, превышающее $\text{Min}(\text{alpha_opaque_value}, \text{alpha_transparent_value})$, должно быть увеличено на единицу для получения значения образца интерпретации для образца избыточно кодированного изображения, и что любое значение образца дополнительно кодированного изображения, меньшее или равное $\text{Min}(\text{alpha_opaque_value}, \text{alpha_transparent_value})$, должно использоваться без изменения как значение образца интерпретации для значения декодированного образца дополнительно кодированного изображения.

alpha_opaque_value определяет значение образца интерпретации образца дополнительно кодированного изображения, для которого связанные образцы яркости и цветности того же блока доступа для целей альфа-смешения рассматриваются непрозрачными. Количество битов, используемых для представления элемента синтаксиса **alpha_opaque_value**, составляет $\text{bit_depth_aux_minus8} + 9$ битов.

alpha_transparent_value определяет значение образца интерпретации образца дополнительно кодированного изображения, для которого связанные образцы яркости и цветности для того же блока доступа для целей альфа-смешения рассматриваются прозрачными. Количество битов, используемых для представления элемента синтаксиса **alpha_transparent_value**, составляет $\text{bit_depth_aux_minus8} + 9$ битов.

Если **alpha_incr_flag** равно 1, **alpha_transparent_value** не должно быть равным **alpha_opaque_value**, а значение $\text{Log}_2(\text{Abs}(\text{alpha_opaque_value} - \text{alpha_transparent_value}))$ должно быть целочисленным. Значение **alpha_transparent_value**, которое равно **alpha_opaque_value**, показывает, что дополнительно кодированное изображение не предназначено для целей альфа-смешения.

ПРИМЕЧАНИЕ 2. – Для целей альфа-смешения **alpha_opaque_value** может превышать **alpha_transparent_value** или может быть меньше **alpha_transparent_value**. Значения образцов интерпретации должны ограничиваться диапазоном от **alpha_opaque_value** до **alpha_transparent_value** включительно.

Для соответствия настоящей Рекомендации | Международному стандарту не требуется декодирования расширения набора параметров последовательности и декодирования дополнительно кодированных изображений.

В отношении синтаксиса всех кодированных секций дополнительно кодированного изображения должны действовать те же ограничения, что и для кодированной секции избыточного изображения, со следующими различиями в ограничениях.

- Следующее применяется в зависимости от того, является ли первично кодированное изображение изображением IDR:
 - если первично кодированное изображение является изображением IDR, синтаксис дополнительно кодированной секции должен соответствовать синтаксису секции, значение **nal_unit_type**, которой равно 5 (секция изображения IDR);
 - иначе (первично кодированное изображение не является изображением IDR) синтаксис дополнительно кодированной секции должен соответствовать синтаксису секции, значение **nal_unit_type** которой равно 1 (секция изображения не IDR).
- Секции дополнительно кодированного изображения (если представлено) должны содержать все макроблоки, соответствующие макроблокам первично кодированного изображения.
- **redundant_pic_cnt** должно быть равным 0 во всех дополнительно кодированных секциях.

Процесс декодирования (необязательный) для декодирования дополнительно кодированных изображений является таким, как если бы дополнительно кодированные изображения были первично кодированными изображениями в отдельном кодированном видеопотоке, который отличается от процесса для первично кодированных изображений в текущем кодированном видеопотоке следующим.

- Статус IDR или не IDR всех дополнительно кодированных изображений должен приниматься таким же, как статус IDR или не IDR первичного изображения в том же блоке доступа, а не определяться по значению `nal_ref_idc`.
- Значение `chroma_format_idc` для декодирования дополнительно кодированных изображений должно приниматься равным 0.
- Значение `bit_depth_luma_minus8` для декодирования дополнительно кодированных изображений должно приниматься равным `bit_depth_aux_minus8`.

ПРИМЕЧАНИЕ 3. – Построение альфа-смешения обычно выполняется с изображением заднего плана В, изображением переднего плана F и декодированным дополнительно кодированным изображением А, размеры которых одинаковы. Предположим, для иллюстрации примера, что разрешающая способность по цветности В и F была подстроена аналогичной разрешающей способности по яркости. Обозначим соответствующие образцы В, F и А как *b*, *f* и *a*, соответственно. Обозначим образцы яркости и цветности с помощью индексов Y, Cb и Cr.

Определим переменные `alphaRange`, `alphaFwt` и `alphaBwt` следующим образом:

```
alphaRange = Abs( alpha_opaque_value - alpha_transparent_value );
alphaFwt = Abs( a - alpha_transparent_value );
alphaBwt = Abs( a - alpha_opaque_value );
```

Далее в построении альфа-смешения образцы *d* отображаемого изображения D можно вычислить следующим образом:

```
dY = ( alphaFwt*fY + alphaBwt*bY + alphaRange/2 ) / alphaRange;
dCB = ( alphaFwt*fCB + alphaBwt*bCB + alphaRange/2 ) / alphaRange;
dCR = ( alphaFwt*fCR + alphaBwt*bCR + alphaRange/2 ) / alphaRange.
```

Образцы изображения D, F и В могут также представлять значения красного, зеленого и синего компонентов (см. п. E.2.1). В данном примере принимаем значения компонентов Y, Cb и Cr. Каждый компонент, например Y, принимается для иллюстрации вышеупомянутого примера имеющим одинаковую глубину цвета в каждом из изображений D, F и В. Вместе с тем разные компоненты, например Y и Cb, в данном примере необязательно должны иметь одинаковую глубину цвета.

Если `aux_format_idc` равно 1, F является декодированным изображением, полученным по декодированным яркости и цвету, а А является декодированным изображением, полученным из декодированного дополнительно кодированного изображения. В этом случае построение альфа-смешения указанного примера включает умножение образцов F на коэффициенты, полученные из образцов А.

Формат изображения, пригодный для редактирования или прямого просмотра и широко используемый, называется изображением, предварительно умноженным с черным цветом. Если изображением переднего плана является F, тогда изображение S, предварительно умноженное с черным цветом, определяется следующими выражениями:

```
sY = ( alphaFwt*fY ) / alphaRange;
sCB = ( alphaFwt*fCB ) / alphaRange;
sCR = ( alphaFwt*fCR ) / alphaRange.
```

Видео, предварительно умноженное с черным цветом, характеризуется тем, что изображение S будет корректным, если отображается с черным задним планом. Для В без черного заднего плана построение отображаемого изображения D можно рассчитать следующим образом:

```
dY = sY + ( alphaBwt*bY + alphaRange/2 ) / alphaRange;
dCB = sCB + ( alphaBwt*bCB + alphaRange/2 ) / alphaRange;
dCR = sCR + ( alphaBwt*bCR + alphaRange/2 ) / alphaRange.
```

Если `aux_format_idc` равно 2, S является декодированным изображением, полученным по декодированным яркости и цвету, а А опять является декодированным изображением, полученным из декодированного дополнительно кодированного изображения. В этом случае построение альфа-смешения не включает умножения образцов S на коэффициенты, полученные из образцов А.

additional_extension_flag, равное 0, указывает, что в синтаксической структуре расширения набора параметров последовательности более не следует данных до конечных битов RBSP. Значение `additional_extension_flag` должно быть равным 0. Значение 1 для `additional_extension_flag` зарезервировано для будущего использования МСЭ-Т | ИСО/МЭК. Декодеры, соответствующие настоящей Рекомендации | Международному стандарту, должны игнорировать все данные, которые следуют за значением 1 флага `additional_extension_flag` в блоке NAL расширения набора параметров последовательности.

7.4.2.2 Семантика RBSP набора параметров изображения

pic_parameter_set_id указывает на набор параметров изображения, на который есть ссылки в заголовке секции. Значение `pic_parameter_set_id` должно быть в диапазоне от 0 до 255 включительно.

seq_parameter_set_id ссылается на активный набор параметров последовательности. Значение `seq_parameter_set_id` должно быть в диапазоне от 0 до 31 включительно.

entropy_coding_mode_flag выбирает энтропию метода декодирования, использованного для элементов синтаксиса, в таблицах которых два дескриптора включены следующим образом:

- если `entropy_coding_mode_flag` равен 0, используют метод, описанный левым дескриптором в таблице синтаксиса (кодированный Exp-Golomb, см. п. 9.1, или CAVLC, см. п. 9.2);
- иначе (`entropy_coding_mode_flag` равен 1) используют метод, описанный правым дескриптором в таблице синтаксиса (CABAC, см. п. 9.3).

pic_order_present_flag, равный 1, определяет, что отсчеты порядка изображений относительно элементов синтаксиса находятся в заголовке секций, как описано в п. 7.3.3. **pic_order_present_flag**, равный 0, определяет, что отсчеты порядка изображений относительно элементов синтаксиса отсутствуют в заголовке секций.

num_slice_groups_minus1 плюс 1 определяет число групп секций для изображения. Если **num_slice_groups_minus1** равно 0, все секции изображения принадлежат той же группе секций. Разрешенный диапазон **num_slice_groups_minus1** описан в Приложении А.

slice_group_map_type определяет характеристики кодирования отображения группы секции и отображения блоков в группах секций. Значение **slice_group_map_type** должно быть в диапазоне от 0 до 6 включительно.

slice_group_map_type, равное 0, определяет чередование групп секций.

slice_group_map_type, равное 1, определяет распределенное отображение группы секций.

slice_group_map_type, равное 2, определяет одну или более "приоритетных" групп секций и "остаток" групп секций.

значения **slice_group_map_type**, равные 3, 4, и 5, определяют изменение групп секций. Если **num_slice_groups_minus1** на равно 1, то значения **slice_group_map_type** не должны быть равны 3, 4 или 5.

slice_group_map_type, равное 6, определяет явно заданное присвоение группы секций каждому блоку отображения группы секций.

Блоки отображения группы секции характеризуют следующим образом:

- если **frame_mbs_only_flag** равен 0, а **mb_adaptive_frame_field_flag** равен 1 и кодированное изображение – это кадр, то блоки отображения группы представляют пару блоков из макроблоков;
- иначе, если **frame_mbs_only_flag** равен 1, или если кодированное изображение – это поле, то блоки отображения группы секций представляют блоки из макроблоков;
- иначе (**frame_mbs_only_flag** равен 0, а **mb_adaptive_frame_field_flag** равен 0, а кодированное изображение – это кадр) блоки отображения группы секций представляют блоки из двух макроблоков, которые являются вертикально-смежными, как пара макроблоков кадра MBAFF.

run_length_minus1[i] используют для спецификации числа последовательных блоков отображения группы секций, которые должны быть присвоены *i*-ой группе секций в порядке растрового сканирования блоков отображения группы секций. Значение **run_length_minus1[i]** должно быть в диапазоне от 0 до **PicSizeInMapUnits** – 1 включительно.

top_left[i] и **bottom_right[i]** определяют левый верхний и нижний правый углы прямоугольника соответственно. **top_left[i]** и **bottom_right[i]** – это местоположения блоков отображения группы секций при растровом сканировании изображения блоков отображения группы секций. Для каждого прямоугольника *i* все следующие ограничения должны быть выдержаны значениями элементов синтаксиса **top_left[i]** и **bottom_right[i]**:

- **top_left[i]** должен быть меньше или равен **bottom_right[i]**, а **bottom_right[i]** должен быть меньше чем **PicSizeInMapUnits**;
- $(\text{top_left}[i] \% \text{PicWidthInMbs})$ должен быть меньше или равен значению $(\text{bottom_right}[i] \% \text{PicWidthInMbs})$.

slice_group_change_direction_flag используют вместе с **slice_group_map_type** для спецификации уточненного типа отображения, если **slice_group_map_type** равен 3, 4 или 5.

slice_group_change_rate_minus1 используют для спецификации переменной **SliceGroupChangeRate**. **SliceGroupChangeRate** определяет кратное число блоков отображения группы секций, при которых размер группы секций может изменяться от одного изображения к другому. Значение **slice_group_change_rate_minus1** должно быть в диапазоне от 0 до **PicSizeInMapUnits** – 1 включительно. Переменную **SliceGroupChangeRate** описывают следующим образом:

$$\text{SliceGroupChangeRate} = \text{slice_group_change_rate_minus1} + 1. \quad (7-20)$$

pic_size_in_map_units_minus1 используют для спецификации числа блоков отображения группы секций в изображении. **pic_size_in_map_units_minus1** должно быть равно **PicSizeInMapUnits** – 1.

slice_group_id[i] определяет группу секций *i*-ого блока отображения группы секций в порядке растрового сканирования. Размер элемента синтаксиса **slice_group_id[i]** – это $\text{Ceil}(\text{Log}_2(\text{num_slice_groups_minus1} + 1))$ битов. Значение **slice_group_id[i]** должно быть в диапазоне от 0 до **num_slice_groups_minus1** включительно.

num_ref_idx_l0_active_minus1 определяет максимум индекса контроля для контрольного изображения списка 0, который должен использоваться для декодирования каждой секции изображения. Для этого изображения используют предсказание списка 0, если для этой секции **num_ref_idx_active_override_flag** равно 0. Если **MbaffFrameFlag** равно 1, то **num_ref_idx_l0_active_minus1** – это максимум индекса для декодирования макроблоков кадра, а $2 * \text{num_ref_idx_l0_active_minus1} + 1$ – это максимум индекса для декодирования макроблоков поля. Значение **num_ref_idx_l0_active_minus1** должно быть в диапазоне от 0 до 31 включительно.

num_ref_idx_l1_active_minus1 имеет ту же семантику, что и **num_ref_idx_l0_active_minus1** с заменой 10 и список 0 на 11 и список 1 соответственно.

weighted_pred_flag, равное 0, определяет, что взвешенное предсказание не должно использоваться в секциях P и SP. Флаг **weighted_pred_flag**, равный 1, определяет, что взвешенное предсказание должно использоваться в секциях P и SP.

weighted_bipred_idc, равное 0, определяет, что взвешенное предсказание по умолчанию должно использоваться в секциях B. Значение **weighted_bipred_idc**, равное 2, определяет, что подразумеваемое взвешенное предсказание должно использоваться в секциях B. Значение **weighted_bipred_idc** должно быть в диапазоне от 0 до 2 включительно.

pic_init_qp_minus26 определяет начальное значение минус 26 $SliceQP_Y$ для каждой секции. Начальное значение модифицируют в слое секции, если декодировано ненулевое значение **slice_qp_delta**, и модифицируют в дальнейшем, если декодировано ненулевое значение **mb_qp_delta** в слое макроблока. Значение **pic_init_qp_minus26** должно быть в диапазоне от $-(26 + QpBdOffset_Y)$ до +25 включительно.

pic_init_qs_minus26 определяет начальное значение минус 26 $SliceQS_Y$ для всех макроблоков в секциях SP или SI. Начальное значение модифицируют в слое секции, если декодировано ненулевое значение **slice_qs_delta**. Значение **pic_init_qs_minus26** должно быть в диапазоне от -26 до +25 включительно.

chroma_qp_index_offset определяет сдвиг, который должен быть добавлен к QP_Y и QS_Y для адресации в таблице значений QP_C для компонента цвета Cb. Значение **chroma_qp_index_offset** должно быть в диапазоне от -12 до +12 включительно.

deblocking_filter_control_present_flag, равное 1, определяет, что установка элементов синтаксиса, контролирующих характеристики фильтра устранения блочности, представлена в заголовках секций. **deblocking_filter_control_present_flag**, равный 0, определяет, что установка элементов синтаксиса, контролирующих характеристики фильтра устранения блочности, не представлена в заголовках секций, а действуют их предполагаемые значения.

constrained_intra_pred_flag, равное 0, определяет, что внутреннее предсказание допускает использование оставшихся данных и декодированных образцов смежных макроблоков, кодированных с использованием режимов внешнего предсказания макроблоков, для предсказания макроблоков, кодированных с использованием внутренних режимов предсказания. **constrained_intra_pred_flag**, равный 1, определяет ограниченное внутреннее предсказание. В этом случае предсказание макроблоков, кодированных с использованием внутренних режимов предсказания макроблоков, будет использовать только оставшиеся данные и декодированные образцы от I или SI типов макроблоков.

redundant_pic_cnt_present_flag, равное 0, определяет, что элемент синтаксиса **redundant_pic_cnt** не представлен в заголовках секций, разделенных участках данных B и разделенных участках данных C, которые считают (либо непосредственно, либо по ассоциации с соответствующим разделением данных A) набором параметров изображения. Значение **redundant_pic_cnt_present_flag**, равное 1, определяет, что элемент синтаксиса **redundant_pic_cnt** представлен во всех заголовках секций, разделенных участков данных B и разделенных участках данных C, которые считают (либо непосредственно, либо по ассоциации с соответствующим разделением данных A) набором параметров изображения.

transform_8x8_mode_flag, равное 1, показывает, что может использоваться процесс декодирования преобразования 8x8 (см. п. 8.5). **transform_8x8_mode_flag**, равное 0, показывает, что процесс декодирования преобразования 8x8 не используется. Если флаг **transform_8x8_mode_flag** не представлен, его значение должно приниматься равным 0.

pic_scaling_matrix_present_flag, равное 1, указывает на наличие параметров для изменения списков масштабирования, определенных в наборе параметров последовательности. **pic_scaling_matrix_present_flag**, равное 0, указывает, что списки масштабирования, используемые для изображения, должны приниматься такими, которые определены набором параметров последовательности. Если флаг **pic_scaling_matrix_present_flag** не представлен, его значение должно приниматься равным 0.

pic_scaling_list_present_flag[i], равное 1, указывает на наличие синтаксической структуры списка масштабирования для определения списка масштабирования для индекса *i*. **pic_scaling_list_present_flag[i]**, равное 0, показывает, что синтаксическая структура для списка масштабирования *i* не представлена в наборе параметров изображения и что в зависимости от значения **seq_scaling_matrix_present_flag** применяется следующее:

- если флаг **seq_scaling_matrix_present_flag** равен 0, для получения списка масштабирования уровня изображения для индекса *i* должен использоваться набор правил обращения к списку масштабирования A, определенный в таблице 7-2;
- иначе (значением **seq_scaling_matrix_present_flag** является 1) для получения списка масштабирования уровня изображения для индекса *i* должен использоваться набор правил обращения к списку масштабирования B, определенный в таблице 7-2.

second_chroma_qp_index_offset определяет сдвиг, который должен быть добавлен к QP_Y и QS_Y для адресации к таблице значений QP_C для компонента цвета Cr. Значение **second_chroma_qp_index_offset** должно находиться в диапазоне от -12 до +12 включительно.

Если **second_chroma_qp_index_offset** не представлено, оно должно приниматься равным **chroma_qp_index_offset**.

7.4.2.3 Семантика дополнительного расширения информации RBSP

Дополнительное расширение информации (SEI) содержит информацию, которая не является необходимой для декодирования образцов кодированных изображений из блоков NAL VCL.

7.4.2.3.1 Семантика сообщения дополнительного расширения информации

Блок NAL с SEI содержит одно или более сообщений SEI. Каждое сообщение SEI состоит из переменных, определяющих тип, payloadType, и размер payloadSize полезной нагрузки SEI. Полезная нагрузка SEI описана в Приложении D. Полученный размер полезной нагрузки, payloadSize, определен в байтах и должен быть равен числу байтов в полезной нагрузке SEI.

ff_byte – байт, равный 0xFF, определяет необходимость дальнейшего представления использованной структуры синтаксиса.

last_payload_type_byte – последний байт типа полезной нагрузки сообщения SEI.

last_payload_size_byte – последний байт размера сообщения SEI.

7.4.2.4 Семантика разграничителя RBSP блока доступа

Разграничитель блока доступа может быть использован для указания типа секций, представленных в первично кодированном изображении, и для упрощения определения границы между блоками доступа. Не существует нормативного процесса декодирования, связанного с разграничителем блока доступа.

primary_pic_type указывает, что значения slice_type для всех секций первично кодированного изображения являются элементами набора, который перечислен в таблице 7-2 для всех заданных значений primary_pic_type.

Таблица 7-5 – Значения primary_pic_type

primary_pic_type	Значения slice_type, которые могут присутствовать в первично кодированном изображении
0	I
1	I, P
2	I, P, B
3	SI
4	SI, SP
5	I, SI
6	I, SI, P, SP
7	I, SI, P, SP, B

7.4.2.5 Семантика конца последовательности RBSP

Конец последовательности RBSP определяет, что последующий блок доступа в потоке битов в порядке декодирования (если такой имеется) должен быть блоком доступа IDR. Содержание синтаксиса SODB и RBSP для конца последовательности RBSP пусто. Для конца последовательности RBSP отсутствует нормативный процесс декодирования.

7.4.2.6 Семантика конца потока битов RBSP

Конец потока битов RBSP указывает, что никаких дополнительных блоков NAL не должно присутствовать в потоке битов, который продвигается к концу потока RBSP в порядке декодирования. Содержание синтаксиса SODB RBSP конца потока RBSP является пустым. Для конца потока RBSP нормативный процесс декодирования не определен.

7.4.2.7 Семантика данных заполнителя RBSP

Данные заполнителя RBSP содержат байты, значение которых должно быть равным 0xFF. Для данных заполнителя RBSP нормативный процесс декодирования не определен.

ff_byte – байт, равный 0xFF.

7.4.2.8 Семантика RBSP слоя без разделения

Слой секции без разделения RBSP состоит из заголовка секции и данных секции.

7.4.2.9 Семантика RBSP деления на части данных секции

7.4.2.9.1 Семантика RBSP деления на части данных А секции

Если используют деление на части данных секции, кодированные данные для простой секции делят на три разные части. Часть А содержит все элементы синтаксиса категории 2.

Элементы синтаксиса категории 2 включают все элементы синтаксиса в заголовке секции, и структуры синтаксиса данных секции иные, чем элементы синтаксиса в оставшейся() структуре синтаксиса.

slice_id идентифицирует секцию, связанную с делением данных. Каждая секция должна иметь особое значение **slice_id** в кодированном изображении, которое содержит эта секция. Если произвольный порядок секции не допустим, как это определено в Приложении А, первая секция кодированного изображения в порядке декодирования должна иметь **slice_id** равно 0, значение **slice_id** должно возрастать на единицу для каждой последующей секции кодированного изображения в порядке декодирования.

Диапазон **slice_id** описан следующим образом:

- если **MbaffFrameFlag** равно 0, **slice_id** должно быть в диапазоне от 0 до **PicSizeInMbs** – 1 включительно;
- иначе (**MbaffFrameFlag** равно 1) **slice_id** должно быть в диапазоне от 0 до **PicSizeInMbs** / 2 – 1 включительно.

7.4.2.9.2 Семантика RBSP деления на части данных В секции

Если используют деление на части данных секции, кодированные данные для простой секции делят на три разные части. Часть разделенных данных В содержит все элементы синтаксиса категории 3.

Элементы синтаксиса категории 4 включают все элементы синтаксиса в оставшейся() структуре синтаксиса и в структурах синтаксиса, использованных для общих типов макроблоков I и SI, как описано в таблице 7-10.

slice_id имеет такую же семантику, которая описана в п. 7.4.2.9.1.

redundant_pic_cnt должен быть равным 0 для секций и секций с разделенными данными, принадлежащими к первично кодированному изображению. Значение **redundant_pic_cnt** должно быть больше 0 для кодированных секций и разделенных частей данных кодированных секций в оставшихся кодированных изображениях. Если **redundant_pic_cnt** отсутствует, это значение должно считаться равным 0. Значение **redundant_pic_cnt** должно быть в диапазоне от 0 до 127 включительно.

Наличие секции RBSP с разделенными данными В определено следующим образом:

- если элементы синтаксиса секции RBSP с разделенными данными А указывают на наличие любого элемента синтаксиса категории 3 в данных секции, секция RBSP с разделенными данными В должна присутствовать с тем же самым значением **slice_id** и **redundant_pic_cnt**, что и в секции RBSP с разделенными данными А;
- иначе (элементы синтаксиса секции RBSP с разделенными данными А не указывают на наличие любого элемента синтаксиса категории 3 в данных секции) секция RBSP с разделенными данными В не должна присутствовать с тем же самым значением **slice_id** и **redundant_pic_cnt**, что и в секции RBSP с разделенными данными А.

7.4.2.9.3 Семантика RBSP деления на части данных С секции

Если используют деление на части данных секции, кодированные данные для простой секции делят на три разные части. Часть разделенных данных С содержит все элементы синтаксиса категории 4.

Элементы синтаксиса категории 4 включают все элементы синтаксиса в оставшейся() структуре синтаксиса и в структурах синтаксиса, использованных для общих типов макроблоков P и B, как описано в таблице 7-10.

slice_id имеет такую же семантику, которая описана в п. 7.4.2.9.1.

redundant_pic_cnt имеет такую же семантику, которая описана в п. 7.4.2.9.2.

Присутствие секции RBSP с разделенными данными С определено следующим образом:

- если элементы синтаксиса секции RBSP с разделенными данными А указывают на наличие любого элемента синтаксиса категории 4 в данных секции, секция RBSP с разделенными данными С должна присутствовать с тем же самым значением **slice_id** и **redundant_pic_cnt**, что и в секции RBSP с разделенными данными А;
- иначе (элементы синтаксиса секции RBSP с разделенными данными А не указывают на наличие любого элемента синтаксиса категории 4 в данных секции) секция RBSP с разделенными данными С не должна присутствовать с тем же самым значением **slice_id** и **redundant_pic_cnt**, что и в секции RBSP с разделенными данными А.

7.4.2.10 Семантика секции концевых битов RBSP

cabac_zero_word – это байт-синхронизированная последовательность из двух байтов, равных 0x0000.

Допустим NumBytesInVclNALunits – сумма значений NumBytesInNALunit для всех блоков NAL VCL кодированного изображения.

Допустим BinCountsInNALunits – число раз, с которым процесс анализа функции DecodeBin(), описанный в п. 9.3.3.2, запускают для декодирования содержания всех блоков NAL VCL кодированного изображения. Если entropy_coding_mode_flag равен 1, то BinCountsInNALunits не должно превышать $(32 \div 3) * \text{NumBytesInVclNALunits} + (\text{RawMbits} * \text{PicSizeInMbs}) \div 32$.

ПРИМЕЧАНИЕ. – Ограничение по максимуму числа бинов, возникающих при декодировании содержания слоя секций блоков NAL, может быть выполнено введением числа элементов синтаксиса cabac_zero_word для увеличения значения NumBytesInVclNALunits. Каждое значение cabac_zero_word представлено в блоке NAL трехбайтовыми последовательностями 0x000003 (как результат ограничения на содержание блока NAL, которое возникает при требовании включения значения emulation_prevention_three_byte для каждого значения cabac_zero_word).

7.4.2.11 Семантика Rbsp концевых битов

rbsp_stop_one_bit – должно быть равно 1.

rbsp_alignment_zero_bit должно быть равно 0.

7.4.3 Семантика заголовка секции

В случае присутствия значения элементов синтаксиса заголовка секции pic_parameter_set_id значения frame_num, field_pic_flag, bottom_field_flag, idr_pic_id, pic_order_cnt_lsb, delta_pic_order_cnt_bottom, delta_pic_order_cnt[0], delta_pic_order_cnt[1], sp_for_switch_flag и slice_group_change_cycle должны быть одинаковыми во всех заголовках секций кодированного изображения.

first_mb_in_slice определяет адрес первого макроблока в секции. Если произвольный порядок секций не разрешен, как это определено в Приложении А, значение first_mb_in_slice не должно быть меньше значения first_mb_in_slice для любой другой секции текущего изображения, которое предшествует текущей секции в порядке декодирования.

Первый адрес макроблока секции находят следующим образом:

- если MbaffFrameFlag равно 0, first_mb_in_slice – это адрес макроблока, первого в секции, а значение first_mb_in_slice должно быть в диапазоне от 0 до PicSizeInMbs – 1 включительно;
- иначе (MbaffFrameFlag равно 1) first_mb_in_slice * 2 – это адрес макроблока, первого в секции, который является верхним макроблоком из первой пары макроблоков в этой секции, а значение first_mb_in_slice должно быть в диапазоне от 0 до PicSizeInMbs / 2 – 1 включительно.

slice_type определяет тип кодирования секции, согласно таблице 7-6.

Таблица 7-6 – Наименование связи с slice_type

slice_type	Наименование связи slice_type
0	P (P секция)
1	B (B секция)
2	I (I секция)
3	SP (SP секция)
4	SI (SI секция)
5	P (P секция)
6	B (B секция)
7	I (I секция)
8	SP (SP секция)
9	SI (SI секция)

Кроме типа кодирования текущей секции, значения slice_type в диапазоне 5..9 определяют, что все другие секции текущего кодированного изображения должны иметь значение slice_type, равное текущему значению slice_type или текущему значению slice_type – 5.

Если nal_unit_type равно 5 (IDR изображение), slice_type должен быть равен 2, 4, 7 или 9.

Если num_ref_frames равно 0, slice_type должен быть равен 2, 4, 7 или 9.

pic_parameter_set_id определяет используемый набор параметров изображения. Значение pic_parameter_set_id должно быть в диапазоне от 0 до 255 включительно.

frame_num используют как идентификатор изображения, и это значение должно быть представлено $\log_2 \max_frame_num - 4 + 4$ битами в потоке битов. Значение **frame_num** ограничено следующим образом.

Переменную **PrevRefFrameNum** определяют следующим образом.

- Если текущее изображение – это IDR изображение, то **PrevRefFrameNum** устанавливают равным 0.
- Иначе (текущее изображение – не IDR изображение) **PrevRefFrameNum** устанавливают следующим образом.
 - если процесс декодирования для промежутков в **frame_num**, определенный в п. 8.2.5.2, активирован процессом декодирования для блока доступа, который содержит неконтрольное изображение, следующее за предыдущим блоком доступа в порядке декодирования, содержащим контрольное изображение, значение **PrevRefFrameNum** устанавливается равным значению **frame_num** последнего из "несуществующих" контрольных кадров, полученных процессом декодирования для промежутков в **frame_num**, определенным в п. 8.2.5.2;
 - иначе **PrevRefFrameNum** устанавливается равным значению **frame_num** для предыдущего блока доступа в порядке декодирования, который содержал контрольное изображение.

Значение **frame_num** ограничено следующим образом.

- Если текущее изображение – это IDR изображение, то **frame_num** должно быть равным 0.
- Иначе (текущее изображение – это IDR изображение) ссылаясь на первично кодированное изображение в предыдущем блоке доступа в порядке декодирования, который содержит контрольное изображение в качестве предыдущего контрольного изображения, значение **frame_num** текущего изображения не должно быть равно **PrevRefFrameNum** до тех пор, пока все следующие условия не будут истинными.
 - Текущее изображение и предыдущее контрольное изображение принадлежат последующим блокам доступа в порядке декодирования.
 - Текущее изображение и предыдущее контрольное изображение – это контрольные поля, имеющие инверсное равенство.
 - Одно или более из следующих условий является истинным:
 - предыдущее контрольное изображение – это IDR изображение,
 - предыдущее контрольное изображение включает значение элемента синтаксиса **memory_management_control_operation**, равное 5;
ПРИМЕЧАНИЕ 1. – Если предыдущее контрольное изображение включает значение элемента синтаксиса **memory_management_control_operation**, равное 5, то **PrevRefFrameNum** равно 0.
 - имеется первично кодированное изображение, которое предшествует предыдущему контрольному изображению, а первично кодированное изображение, которое предшествует предыдущему контрольному изображению, не содержит **frame_num**, равное **PrevRefFrameNum**;
 - имеется первично кодированное изображение, которое предшествует предыдущему контрольному изображению, а первично кодированное изображение, которое предшествует предыдущему контрольному изображению, не является контрольным изображением.

Если значение **frame_num** не равно **PrevRefFrameNum**, то применяют следующее.

- В порядке декодирования не должно быть никаких предыдущих полей или кадров, которые отмечены как "используется для укороченного контроля" со значением **frame_num**, равным любому значению, полученному переменной **UnusedShortTermFrameNum** в следующих случаях:

$$\begin{aligned} \text{UnusedShortTermFrameNum} &= (\text{PrevRefFrameNum} + 1) \% \text{MaxFrameNum} \\ \text{while}(\text{UnusedShortTermFrameNum} \neq \text{frame_num}) & \\ \text{UnusedShortTermFrameNum} &= (\text{UnusedShortTermFrameNum} + 1) \% \text{MaxFrameNum}. \end{aligned} \quad (7-21)$$

- Значение **frame_num** ограничено следующим образом.
 - Если **gaps_in_frame_num_value_allowed_flag** равно 0, значение **frame_num** для текущего изображения должно быть равно $(\text{PrevRefFrameNum} + 1) \% \text{MaxFrameNum}$.
 - Иначе (**gaps_in_frame_num_value_allowed_flag** равно 1) используют следующее.
 - Если **frame_num** больше чем **PrevRefFrameNum**, то не должно быть никаких неконтрольных изображений в потоке битов, который следует за предыдущим контрольным изображением и предшествует текущему изображению в порядке декодирования, в котором любое из следующих условий является истинным:
 - значение **frame_num** неконтрольного изображения меньше, чем **PrevRefFrameNum**;
 - значение **frame_num** неконтрольного изображения больше, чем значение **frame_num** текущего изображения.

- Иначе ($frame_num$ меньше чем $PrevRefFrameNum$) не должно быть никаких неконтрольных изображений в потоке битов, который следует за предыдущим контрольным изображением и предшествует текущему изображению в порядке декодирования, в котором любое из следующих условий является истинным:
 - значение $frame_num$ неконтрольного изображения меньше, чем $PrevRefFrameNum$;
 - значение $frame_num$ неконтрольного изображения больше, чем $frame_num$ текущего изображения.

Изображение, включающее $memory_management_control_operation$, равное 5, должно иметь ограничения $frame_num$, как описано выше, а после декодирования текущего изображения и обработки операций управления памятью считают, что изображение должно иметь значение $frame_num$, равное 0, для всех последующих использований в процессе декодирования, за исключением описанных в п. 7.4.1.2.4.

ПРИМЕЧАНИЕ 2. – Если первично кодированное изображение не является IDR изображением и не содержит элемент синтаксиса $memory_management_control_operation$, равный 5, значение $frame_num$ соответствующего избыточно кодированного изображения будет таким же, что и значение $frame_num$ первично кодированного изображения. В противном случае, дополнительно кодированное изображение включает элемент синтаксиса $memory_management_control_operation$, равный 5, а соответствующее первично кодированное изображение является IDR изображением.

field_pic_flag, равный 1, определяет, что данная секция – это секция кодированного поля. **field_pic_flag**, равный 0, определяет, что данная секция – это секция кодированного кадра. Если отсутствует **field_pic_flag**, то считают, что это значение должно быть равным 0.

Переменную $MbaffFrameFlag$ определяют следующим образом:

$$MbaffFrameFlag = (mb_adaptive_frame_field_flag \&\& !field_pic_flag). \quad (7-22)$$

Переменную для изображения высоты в блоках макроблоков определяют следующим образом:

$$PicHeightInMbs = FrameHeightInMbs / (1 + field_pic_flag). \quad (7-23)$$

Переменную для изображения высоты компонента яркости определяют следующим образом:

$$PicHeightInSamples_L = PicHeightInMbs * 16. \quad (7-24)$$

Переменную для изображения высоты компонента цветности определяют следующим образом:

$$PicHeightInSamples_C = PicHeightInMbs * MbHeightC. \quad (7-25)$$

Переменную $PicSizeInMbs$ для текущего изображения определяют согласно уравнению:

$$PicSizeInMbs = PicWidthInMbs * PicHeightInMbs. \quad (7-26)$$

Переменную $MaxPicNum$ определяют следующим образом:

- если **field_pic_flag** равен 0, $MaxPicNum$ устанавливают равным $MaxFrameNum$;
- иначе (**field_pic_flag** равен 1) $MaxPicNum$ устанавливают равным $2 * MaxFrameNum$.

Переменную $CurrPicNum$ определяют следующим образом:

- если **field_pic_flag** равен 0, $CurrPicNum$ устанавливают равным $frame_num$;
- иначе (**field_pic_flag** равен 1) $CurrPicNum$ устанавливают равным $2 * frame_num + 1$.

bottom_field_flag, равный 1, определяет, что эта секция составляет часть нижнего кодированного поля. **bottom_field_flag**, равный 0, определяет, что изображение – это кодированное верхнее поле. Если этот элемент синтаксиса не представлен в текущей секции, он должен считаться равным 0.

idr_pic_id определяет IDR изображение. Значение **idr_pic_id** во всех секциях IDR изображения должно оставаться неизменным. Если оба из двух последующих блоков доступа в порядке декодирования являются IDR блоками доступа, значение **idr_pic_id** в секциях первого такого IDR блока доступа должно отличаться от **idr_pic_id** второго такого IDR блока доступа. Значение **idr_pic_id** должно быть в диапазоне от 0 до 65535 включительно.

pic_order_cnt_lsb определяет модуль счета порядка изображений $MaxPicOrderCntLsb$ для верхнего поля кодированного кадра или кодированного поля. Размер элемента синтаксиса **pic_order_cnt_lsb** – это $\log_2_max_pic_order_cnt_lsb_minus4 + 4$ битов. Значение **pic_order_cnt_lsb** должно быть в диапазоне от 0 до $MaxPicOrderCntLsb - 1$ включительно.

delta_pic_order_cnt_bottom определяет разность счета в порядке изображения между нижним и верхним полями кодированного кадра следующим образом:

- если текущее изображение включает `memory_management_control_operation`, равное 5, значение `delta_pic_order_cnt_bottom` должно быть в диапазоне от $(1 - \text{MaxPicOrderCntLsb})$ до $2^{31} - 1$ включительно;
- иначе (текущее изображение не включает `memory_management_control_operation`, равное 5) значение `delta_pic_order_cnt_bottom` должно быть в диапазоне от -2^{31} до $2^{31} - 1$ включительно.

Если этот элемент синтаксиса не представлен в потоке битов для текущей секции, его полагают равным 0.

delta_pic_order_cnt[0] определяет разность счета в порядке изображения от ожидаемого порядка изображений для верхнего поля кодированного кадра или кодированного поля, как описано в п. 8.2.1. Значение `delta_pic_order_cnt[0]` должно быть в диапазоне от -2^{31} до $2^{31} - 1$ включительно. Если этот элемент синтаксиса не представлен в потоке битов для текущей секции, его полагают равным 0.

delta_pic_order_cnt[1] определяет разность счета в порядке изображения от ожидаемого порядка изображений для нижнего поля кодированного кадра или кодированного поля, как описано в п. 8.2.1. Значение `delta_pic_order_cnt[1]` должно быть в диапазоне от -2^{31} до $2^{31} - 1$ включительно. Если этот элемент синтаксиса не представлен в потоке битов для текущей секции, его полагают равным 0.

redundant_pic_cnt должен быть равным 0 для секций и частей секций с разделенными данными, принадлежащими первично кодированному изображению. Значение `redundant_pic_cnt` должно быть больше, чем 0 для кодированных секций или кодированных секций с разделенными данными избыточно кодированного изображения. Если `redundant_pic_cnt` не представлен в потоке битов, его полагают равным 0. Значение `redundant_pic_cnt` должно быть в диапазоне от 0 до 127 включительно.

ПРИМЕЧАНИЕ 3. – Любая область декодированного первичного изображения и соответствующая область, которая является результатом применения процесса декодирования, описанного в главе 8, для любого избыточно кодированного изображения в том же блоке доступа, должны быть визуально похожи друг на друга.

Значение `pic_parameter_set_id` в кодированной секции или кодированной секции с разделенными данными избыточно кодированного изображения должно быть таким, чтобы значение `pic_order_present_flag` в наборе параметров изображения при использовании избыточно кодированного изображения было равно значению `pic_order_present_flag` при текущем наборе параметров изображения, соответствующему первично кодированному изображению.

В представленном первично кодированном изображении и любом избыточно кодированном изображении, следующий за элементами синтаксиса элемент должен иметь те же значения: `field_pic_flag`, `bottom_field_flag` и `idr_pic_id`.

Если значение `nal_ref_idc` в блоке NAL VCL блока доступа равно 0, значение `nal_ref_idc` во всех других блоках NAL VCL того же блока доступа должно быть равным 0.

ПРИМЕЧАНИЕ 4. – Вышеприведенное ограничение имеет также следующий смысл. Если значение `nal_ref_idc` для блоков NAL VCL первично кодированного изображения равно 0, то значения `nal_ref_idc` для блоков NAL VCL любого соответствующего избыточно кодированного изображения равны 0; иначе (значение `nal_ref_idc` для блоков NAL VCL первично кодированного изображения больше 0) значения `nal_ref_idc` для блоков NAL VCL любого соответствующего избыточно кодированного изображения также больше 0.

Статус разметки контрольных изображений и значение `frame_numafter` после того, как процесс разметки декодированного контрольного изображения (согласно п. 8.2.5) был запущен для первично кодированного изображения или любого избыточно кодированного изображения того же блока доступа, должен оставаться тем же самым независимо от того, были ли декодированы первично кодированное изображение или избыточно кодированное изображение (вместо первично кодированного изображения) этого блока доступа.

ПРИМЕЧАНИЕ 5. – Вышеприведенное ограничение имеет также следующий смысл.

Если первично кодированное изображение – это не IDR изображение, содержание структуры синтаксиса `dec_ref_pic_marking()` должно быть идентичным во всех заголовках секций первично кодированного изображения и всех избыточно кодированных изображений, соответствующих этому первично кодированному изображению.

Иначе (первично кодированное изображение – это IDR изображение) применяют следующее.

Если избыточно кодированное изображение, соответствующее первично кодированному изображению, – это не IDR изображение, то содержание структуры синтаксиса `dec_ref_pic_marking()` должно быть идентичным во всех заголовках секций первично кодированного изображения и во всех избыточно кодированных изображениях, соответствующих первично кодированному изображению.

Иначе (избыточно кодированное изображение, соответствующее первично кодированному изображению, – это IDR изображение) все заголовки секций избыточного изображения должны содержать структуру синтаксиса(), включающую элемент синтаксиса `memory_management_control_operation`, равный 5, а далее применяют следующее.

Если значение `long_term_reference_flag` в первично кодированном изображении равно 0, то структура синтаксиса `dec_ref_pic_marking` избыточно кодированного изображения не должна включать элемент синтаксиса `memory_management_control_operation`, равный 6.

Иначе (значение `long_term_reference_flag` в первично кодированном изображении равно 1) структура синтаксиса `dec_ref_pic_marking` избыточно кодированного изображения должна включать элемент синтаксиса `memory_management_control_operation`, равный 5, 4 и 6 в порядке декодирования, значение `max_long_term_frame_idx_plus1` должно быть равно 1, а значение `long_term_frame_idx` должно быть равно 0.

Значения `TopFieldOrderCnt` и `BottomFieldOrderCnt` (если они применимы), которые являются результатом завершения процесса декодирования любого избыточно кодированного изображения или первично кодированного изображения того же блока доступа должны быть идентичными независимо от того, будут ли декодированы первично кодированное изображение или любое избыточно кодированное изображение (вместо первично кодированного изображения) блока доступа.

Не существует обязательного процесса декодирования для кодированной секции (или для кодированной секции с разделенными данными) избыточно кодированного изображения. Если значение `redundant_pic_cnt` заголовка кодированной секции больше 0, декодер может отбросить эту кодированную секцию. Однако кодированная секция (или кодированная секция с разделенными данными) любого избыточно кодированного изображения должна подчиняться тем же ограничениям, что и кодированная секция (или кодированной секции с разделенными данными) исходного изображения.

ПРИМЕЧАНИЕ 6. – Если некоторые образцы в первично декодированном изображении не могут быть корректно декодированы из-за ошибок или потерь в передаче последовательности, а кодированная дополнительная секция может быть декодирована корректно, декодер должен заменить образцы декодированного исходного изображения соответствующими образцами декодированной дополнительной секции. Если более чем одна дополнительная секция перекрывает существенную область исходного изображения, должна быть использована дополнительная секция, имеющая наименьшее значение `redundant_pic_cnt`.

Дополнительные секции и секции разделенных данных, имеющих такое же значение `redundant_pic_cnt`, принадлежат к этому же избыточному изображению. Декодированные секции в этом же избыточном изображении не должны покрывать и перекрывать всю площадь изображения.

direct_spatial_mv_pred_flag определяет метод, использованный в процессе декодирования для получения векторов движения и индексов контроля для внешнего предсказания следующим образом:

- если `direct_spatial_mv_pred_flag` равен 1, процесс вывода векторов движения яркости для `B_Skip`, `B_Direct_16x16` и `B_Direct_8x8` в п. 8.4.1.2 должен использовать режим прямого пространственного предсказания, как описано в п. 8.4.1.2.2;
- иначе (`direct_spatial_mv_pred_flag` равен 0) процесс вывода векторов движения яркости для `B_Skip`, `B_Direct_16x16` и `B_Direct_8x8` в п. 8.4.1.2 должен использовать режим временного прямого предсказания, как описано в п. 8.4.1.2.3.

num_ref_idx_active_override_flag, равный 0, определяет, что действуют значения элементов синтаксиса `num_ref_idx_l0_active_minus1` и `num_ref_idx_l1_active_minus1`, которые определены в наборе параметров изображения. `num_ref_idx_active_override_flag`, равный 1, определяет, что значения `num_ref_idx_l0_active_minus1` и `num_ref_idx_l1_active_minus1`, которые определены в наборе параметров изображения, отменены для текущей секции (и только для текущей секции) следующими значениями в заголовке секции:

Если текущая секция – это секция P, SP или B, а `field_pic_flag` равен 0 и значение `num_ref_idx_l0_active_minus1` в наборе параметров изображения превышает 15, то `num_ref_idx_active_override_flag` должен быть равен 1.

Если текущая секция – это секция B, а `field_pic_flag` равен 0 и значение `num_ref_idx_l1_active_minus1` в наборе параметров изображения превышает 15, то `num_ref_idx_active_override_flag` должен быть равен 1.

num_ref_idx_l0_active_minus1 определяет максимум индекса контроля для контрольного изображения списка 0, который должен использоваться для декодирования секции.

Диапазон `num_ref_idx_l0_active_minus1` описан следующим образом:

- если `field_pic_flag` равен 0, `num_ref_idx_l0_active_minus1` должен быть в диапазоне от 0 до 15 включительно. Если `MbaffFrameFlag` равен 1, то `num_ref_idx_l0_active_minus1` – это максимальное значение индекса для декодирования кадров макроблоков, а $2 * \text{num_ref_idx_l0_active_minus1} + 1$ – это максимальное значение индекса для декодирования полей макроблоков;
- иначе (`field_pic_flag` равен 1) `num_ref_idx_l0_active_minus1` должен быть в диапазоне от 0 до 31 включительно.

num_ref_idx_l1_active_minus1 имеет ту же семантику, что и `num_ref_idx_l0_active_minus1` с заменой l0 и список 0, на l1 и список 1 соответственно.

cabac_init_idc описывает индекс для определения таблицы инсталляции, использованной в процессе инсталляции переменных контекста. Значение `cabac_init_idc` должно быть в диапазоне от 0 до 2 включительно.

slice_qp_delta определяет начальное значение QP_Y , которое следует использовать для всех макроблоков в секции, которые до тех пор модифицировали значением `mb_qp_delta` в слое макроблока. Начальный параметр квантования QP_Y для секции вычисляют следующим образом:

$$\text{SliceQP}_Y = 26 + \text{pic_init_qp_minus26} + \text{slice_qp_delta}. \quad (7-27)$$

Значение `slice_qp_delta` должно быть ограничено таким образом, чтобы SliceQP_Y находилось в диапазоне от $-\text{QpBdOffset}_Y$ до +51 включительно.

sp_for_switch_flag определяет процесс декодирования, который следует использовать для декодирования макроблоков P в секции SP следующим образом:

- если `sp_for_switch_flag` равен 0, макроблоки P в секции SP должны быть декодированы, используя процесс декодирования SP для не переключающихся изображений, как описано в п. 8.6.1;
- иначе (`sp_for_switch_flag` равен 1) макроблоки P в секции SP должны быть декодированы, используя процесс декодирования SP и SI для переключающихся изображений, как описано в п. 8.6.2.

slice_qs_delta определяет значение QS_Y для всех макроблоков в секциях SP и SI. Параметр квантования QS_Y для секции вычисляют следующим образом:

$$QS_Y = 26 + pic_init_qs_minus26 + slice_qs_delta. \quad (7-28)$$

Значение **slice_qs_delta** должно быть ограничено таким образом, чтобы QS_Y находилось в диапазоне от 0 до 51 включительно. Значение QS_Y используют для декодирования всех макроблоков в секциях SI с **mb_type**, равным SI, и для всех макроблоков в секциях SP с режимом внешнего предсказания (**inter** предсказания).

disable_deblocking_filter_idc определяет, действительно ли операция разблокирования (разделения на блоки) фильтра должна быть аннулирована для некоторых краев блока секции, а также определяет, для каких краев аннулируют фильтрацию. Если **disable_deblocking_filter_idc** отсутствует в заголовке секции, значение **disable_deblocking_filter_idc** должно быть принято равным 0.

Значение **disable_deblocking_filter_idc** должно быть в диапазоне от 0 до 2 включительно.

slice_alpha_c0_offset_div2 определяет сдвиг, использованный для доступа к таблицам разблокирования фильтра α и t_{c0} , при операциях фильтрации, которыми управляют макроблоки в секции. Исходя из этого значения следует использовать сдвиг в тех случаях, когда адресацию к этим таблицам вычисляют как:

$$\text{FilterOffsetA} = \text{slice_alpha_c0_offset_div2} \ll 1. \quad (7-29)$$

Значение **slice_alpha_c0_offset_div2** должно быть в диапазоне от -6 до +6 включительно. Если **slice_alpha_c0_offset_div2** отсутствует в заголовке секции, значение **slice_alpha_c0_offset_div2** должно считаться равным 0.

slice_beta_offset_div2 определяет сдвиг, использованный для доступа к таблицам разблокирования фильтра β при операциях фильтрации, которыми управляют макроблоки в секции. Исходя из этого значения, следует использовать сдвиг в тех случаях, когда адресацию таблице β вычисляют как:

$$\text{FilterOffsetB} = \text{slice_beta_offset_div2} \ll 1. \quad (7-30)$$

Значение **slice_beta_offset_div2** должно быть в диапазоне от -6 до +6 включительно. Если **slice_beta_offset_div2** отсутствует в заголовке секции, значение **slice_beta_offset_div2** должно считаться равным 0.

slice_group_change_cycle используют для вывода числа блоков отображения группы секции в группу секций 0, если тип **slice_group_map_type** равен 3, 4 или 5, как это определено уравнением:

$$\text{MapUnitsInSliceGroup0} = \text{Min}(\text{slice_group_change_cycle} * \text{SliceGroupChangeRate}, \text{PicSizeInMapUnits}). \quad (7-31)$$

Значение **slice_group_change_cycle** представлено в потоке битов следующим числом битов:

$$\text{Ceil}(\text{Log2}(\text{PicSizeInMapUnits} \div \text{SliceGroupChangeRate} + 1)). \quad (7-32)$$

Значение **slice_group_change_cycle** должно быть в диапазоне от 0 до $\text{Ceil}(\text{PicSizeInMapUnits} \div \text{SliceGroupChangeRate})$ включительно.

7.4.3.1 Семантика изменения порядка в списке контрольного изображения

Элементы синтаксиса **reordering_of_pic_nums_idc**, **abs_diff_pic_num_minus1** и **long_term_pic_num** определяют изменение списка от первоначального контрольного изображения на список контрольных изображений, который следует использовать для декодирования секции.

ref_pic_list_reordering_flag_l0, равный 1, определяет, что элемент синтаксиса **reordering_of_pic_nums_idc** представлен для описания списка 0 контрольного изображения. **ref_pic_list_reordering_flag_l0**, равный 0, определяет, что этот элемент синтаксиса отсутствует.

Если **ref_pic_list_reordering_flag_l0** равен 1, то число раз, при котором значение **reordering_of_pic_nums_idc**, не равное 3, следующее за значением **ref_pic_list_reordering_flag_l0**, не должно превышать **num_ref_idx_l0_active_minus1 + 1**.

Если значение **RefPicList0[num_ref_idx_l0_active_minus1]** в полученном исходном списке контрольных изображений (как описано в п. 8.2.4.2) равно "неконтрольному изображению", то **ref_pic_list_reordering_flag_l0** должен быть равен 1, а **reordering_of_pic_nums_idc** не должно быть равно 3 до тех пор, пока значение **RefPicList0[num_ref_idx_l0_active_minus1]** в преобразованном списке, полученное как описано в п. 8.2.4.3, не станет равным "неконтрольному изображению".

ref_pic_list_reordering_flag_11, равное 1, определяет, что элемент синтаксиса **reordering_of_pic_nums_idc** представлен для описания списка 1 контрольного изображения. **ref_pic_list_reordering_flag_11**, равное 0, определяет, что этот элемент синтаксиса отсутствует.

Если **ref_pic_list_reordering_flag_11** равен 1, то число раз, при котором значение **reordering_of_pic_nums_idc**, равное 3, следующее за значением **ref_pic_list_reordering_flag_11**, не должно превышать **num_ref_idx_11_active_minus1 + 1**.

Если декодируют секцию В и значение **RefPicList1[num_ref_idx_11_active_minus1]** в полученном исходном списке контрольных изображений (как описано в п. 8.2.4.2) равно "неконтрольному изображению", то **ref_pic_list_reordering_flag_11** должен быть равен 1, а **reordering_of_pic_nums_idc** не должно быть равно 3 до тех пор, пока значение **RefPicList1[num_ref_idx_11_active_minus1]** в списке с измененным порядком, как описано в п. 8.2.4.3, не станет равным "неконтрольному изображению".

reordering_of_pic_nums_idc вместе с **abs_diff_pic_num_minus1** или **long_term_pic_num** определяет, какое из контрольных изображений отображается повторно. Значения **reordering_of_pic_nums_idc** описаны в таблице 7-7. Первое значение **reordering_of_pic_nums_idc**, которое следует после **ref_pic_list_reordering_flag_10** или **ref_pic_list_reordering_flag_11**, не должно быть равно 3.

Таблица 7-7 – Списки контрольных изображений изменения порядка операций **reordering_of_pic_nums_idc**

reordering_of_pic_nums_idc	Характеристики изменения порядка
0	abs_diff_pic_num_minus1 представлен и соответствует разности вычитания из номера изображения значения предсказания
1	abs_diff_pic_num_minus1 представлен и соответствует разности сложения значения предсказания с номером изображения
2	long_term_pic_num представлен и определяет номер долгосрочного изображения для контрольного изображения
3	Конец цикла преобразования порядка в списке первоначального контрольного изображения

abs_diff_pic_num_minus1 плюс 1 определяет абсолютную разность между номером изображения, индекс которого изменяют в списке на текущий, и предсказанным значением номера. Значение **abs_diff_pic_num_minus1** должно быть в диапазоне от 0 до **MaxPicNum - 1**. Допустимые значения **abs_diff_pic_num_minus1** далее ограничены, как это описано в п. 8.2.4.3.1.

long_term_pic_num определяет долгосрочный номер изображения из тех, индекс которых изменяют в списке на текущий. При декодировании кодированного кадра **long_term_pic_num** должен быть равен значению **LongTermPicNum**, присвоенному одному из контрольных кадров или паре дополнительных контрольных полей, помеченных как "используется для долгосрочного контроля". При декодировании кодированного поля **long_term_pic_num** должен быть равен значению **LongTermPicNum**, присвоенному одному из контрольных полей, "используется для долгосрочного контроля".

7.4.3.2 Семантика таблицы веса предсказания

luma_log2_weight_denom – это логарифм по основанию 2 от знаменателя для всех факторов взвешивания яркости. Значение **luma_log2_weight_denom** должно быть в диапазоне от 0 до 7 включительно.

chroma_log2_weight_denom – это логарифм по основанию 2 от знаменателя для всех факторов взвешивания цветности. Значение **chroma_log2_weight_denom** должно быть в диапазоне от 0 до 7 включительно.

luma_weight_10_flag равно 1, определяет, что присутствуют факторы взвешивания для компонента яркости списка предсказаний 0. Значение **luma_weight_10_flag**, равное 0, определяет, что факторы взвешивания отсутствуют.

luma_weight_10[i] – это фактор взвешивания для предсказания значения яркости по списку предсказаний 0 с использованием **RefPicList0[i]**. Если **luma_weight_10_flag** равно 1, значение **luma_weight_10[i]** должно быть в диапазоне от -128 до 127 включительно. Если **luma_weight_10_flag** равно 0, **luma_weight_10[i]** должно быть принято равным $2^{\text{luma_log2_weight_denom}}$ для **RefPicList0[i]**.

luma_offset_10[i] – это аддитивный сдвиг для предсказания значения яркости по списку предсказаний 0 с использованием **RefPicList0[i]**. Значение **luma_offset_10[i]** должно быть в диапазоне от -128 до 127 включительно. Если **luma_weight_10_flag** равно 0, **luma_offset_10[i]** должно быть принято равным 0 для **RefPicList0[i]**.

chroma_weight_10_flag, равное 1, определяет, что присутствуют факторы взвешивания для предсказания значения цветности по списку предсказаний 0. Значение **chroma_weight_10_flag**, равное 0, определяет, что эти факторы взвешивания отсутствуют.

chroma_weight_10[i][j] – это фактор взвешивания для предсказания значения цветности по списку предсказаний 0 с использованием RefPicList0[i] при j равным 0 для Cb и j равным 1 для Cr. Если chroma_weight_10_flag равно 1, значение chroma_weight_10[i][j] должно быть в диапазоне от –128 до 127 включительно. Если chroma_weight_10_flag равно 0, chroma_weight_10[i][j] должно быть принято равным $2^{\text{chroma_log2_weight_denom}}$ для RefPicList0[i].

chroma_offset_10[i][j] – это аддитивный сдвиг для предсказания значения цветности по списку предсказаний 0 с использованием RefPicList0[i] при j равным 0 для Cb и j равным 1 для Cr. Значение chroma_offset_10[i][j] должно быть в диапазоне от –128 до 127 включительно. Если chroma_weight_10_flag равно 0, chroma_offset_10[i][j] должно быть принято равным 0 для RefPicList0[i].

luma_weight_11_flag, luma_weight_11, luma_offset_11, chroma_weight_11_flag, chroma_weight_11, chroma_offset_11 имеют ту же семантику, что и luma_weight_10_flag, luma_weight_10, luma_offset_10, chroma_weight_10_flag, chroma_weight_10, chroma_offset_10, соответственно, с заменой 10, список 0 и List0 на 11, список 1 и List1.

7.4.3.3 Семантика разметки декодированного контрольного изображения

Элементы синтаксиса no_output_of_prior_pics_flag, long_term_reference_flag, adaptive_ref_pic_marking_mode_flag, memory_management_control_operation, difference_of_pic_nums_minus1, long_term_frame_idx, long_term_pic_num, и max_long_term_frame_idx_plus1 определяют разметку контрольных изображений.

Разметками контрольного изображения могут быть: "не используется для контроля", "используется для краткосрочного контроля" или "используется для долгосрочного контроля", но только одно из трех. Если контрольное изображение считают помеченным как "используется для контроля", его в собирательном смысле считают изображением, помеченным как "используется для краткосрочного контроля" или "используется для долгосрочного контроля" (но не оба вместе). Контрольное изображение, помеченное как "используется для краткосрочного контроля", считается краткосрочным контрольным изображением. Контрольное изображение, помеченное как "используется для долгосрочного контроля" считается долгосрочным контрольным изображением.

Элемент синтаксиса adaptive_ref_pic_marking_mode_flag и содержание структуры синтаксиса разметки декодированного контрольного изображения должны быть идентичны для всех кодированных секций кодированного изображения.

Категория синтаксиса разметки декодированного контрольного изображения должна присваиваться следующим образом:

- если структура синтаксиса разметки декодированного контрольного изображения находится в заголовке секции, категория синтаксиса для структуры синтаксиса разметки декодированного контрольного изображения должна полагаться равной 2;
- иначе (структура синтаксиса разметки декодированного контрольного изображения находится в повторном сообщении SEI о разметке декодированного контрольного изображения, как это определено в Приложении D) категория синтаксиса для структуры синтаксиса разметки декодированного контрольного изображения должна полагаться равной 5.

no_output_of_prior_pics_flag определяет, как обрабатывают ранее декодированные изображения в буфере декодированного изображения после декодирования IDR изображения. См. Приложение C. Если IDR изображение – это первое IDR изображение в потоке битов, значение no_output_of_prior_pics_flag не оказывает влияния на процесс декодирования. Если IDR изображение – это не первое IDR изображение в потоке битов, значения PicWidthInMbs, FrameHeightInMbs или max_dec_frame_buffering, которые находят из действующего набора параметров последовательности, отличаются от значений PicWidthInMbs, FrameHeightInMbs или max_dec_frame_buffering, которые находят из предыдущего действующего набора параметров последовательности, а no_output_of_prior_pics_flag равно 1, может быть интерпретировано декодером независимо от действующего значения no_output_of_prior_pics_flag.

long_term_reference_flag равно 0, определяет, что переменная MaxLongTermFrameIdx установлена в положение, равное "индексы недолгосрочных кадров", и что IDR изображение помечено как "используется для краткосрочного контроля". long_term_reference_flag, равно 1, определяет, что переменная MaxLongTermFrameIdx установлена в положение, равное 0, и что текущее IDR изображение помечено "используется для долгосрочного контроля" и задано значением LongTermFrameIdx, равным 0. Если num_ref_frames равно 0, значение long_term_reference_flag должно быть равно 0.

adaptive_ref_pic_marking_mode_flag выбирает режим разметки контрольного изображения текущего декодированного изображения так, как это описано в таблице 7-8. Значение adaptive_ref_pic_marking_mode_flag должно быть равно 1, если число кадров, пар дополнительных полей и непарных полей, которые в данный момент помечены как "используется для долгосрочного контроля", равно Max(num_ref_frames, 1).

Таблица 7-8 – Интерпретация значения `adaptive_ref_pic_marking_mode_flag`

<code>adaptive_ref_pic_marking_mode_flag</code>	Описание режима разметки контрольного изображения
0	Режим разметки контрольного изображения скользящим окном: Режим разметки краткосрочных контрольных изображений, основанный на механизме "первый пришел – первым обслужен" (first-in first-out).
1	Адаптивный режим разметки контрольного изображения: Режим разметки контрольного изображения, создающий элементы синтаксиса для спецификации разметки контрольных изображений как "не используется для контроля" и присваивающий долгосрочные индексы кадров.

`memory_management_control_operation` определяет, что для воздействия на разметку контрольного изображения следует использовать операцию управления. Элемент синтаксиса `memory_management_control_operation` следует за данными, необходимыми для описания операции `memory_management_control_operation`. Значения и операции управления, связанные с `memory_management_control_operation`, описаны в таблице 7-9. Элементы синтаксиса `memory_management_control_operation` обрабатывают процесс декодирования в том порядке, в котором они появляются в заголовке секции, а семантику ограничения, выраженную для каждого значения `memory_management_control_operation`, используют в особой позиции в том порядке, в котором обрабатывают отдельные значения `memory_management_control_operation`.

Для интерпретации `memory_management_control_operation` термин "контрольное изображение" интерпретируется следующим образом:

- если текущим изображением является кадр, термин "контрольное изображение" относится либо к контрольному кадру, либо к паре дополнительных контрольных полей;
- иначе (текущее изображение является полем) термин "контрольное изображение" относится либо к контрольному полю, либо к полю контрольного кадра..

Значение `memory_management_control_operation` в заголовке секции не должно быть равным 1, если заданное контрольное изображение не помечено как "используется для краткосрочного контроля", при обработке `memory_management_control_operation` процессом декодирования.

Значение `memory_management_control_operation` в заголовке секции не должно быть равным 2, если заданный номер долгосрочного изображения не указывает на контрольное изображение, которое помечено как "используется для долгосрочного контроля", при обработке `memory_management_control_operation` процессом декодирования.

Значение `memory_management_control_operation` в заголовке секции не должно быть равным 3, если заданное контрольное изображение не помечено как "используется для краткосрочного контроля" при обработке значения `memory_management_control_operation` процессом декодирования.

Значение `memory_management_control_operation` не должно быть равным 3 или 6, если значение переменной `MaxLongTermFrameIdx` равно значению "нет долгосрочных индексов кадра" и если `memory_management_control_operation` обрабатывают процессом декодирования.

В заголовке секции должно быть не более одного значения `memory_management_control_operation`, равного 4.

В заголовке секции должно быть не более одного значения `memory_management_control_operation`, равного 5.

В заголовке секции должно быть не более одного значения `memory_management_control_operation`, равного 6.

В заголовке секции значение `memory_management_control_operation` не должно быть равно 5, если в диапазоне от 1 до 3 отсутствует значение `memory_management_control_operation` в той же самой структуре синтаксиса разметки декодированного контрольного изображения.

`memory_management_control_operation`, равное 5, не должно следовать за `memory_management_control_operation`, равным 6, в том же заголовке секции.

Если имеется значение `memory_management_control_operation`, равное 6, любые значения `memory_management_control_operation`, равные 2, 3 или 4, которые следуют за `memory_management_control_operation`, равным 6 в том же заголовке секции, не должны описывать текущее изображение, которое следует пометить как "не используется для контроля".

ПРИМЕЧАНИЕ 1. – Эти ограничения запрещают любые комбинации многочисленных элементов синтаксиса `memory_management_control_operation`, которые могли бы определить текущее изображение, помеченное как "не используется для контроля". Однако разрешены некоторые другие комбинации элементов синтаксиса `memory_management_control_operation`, которые могут влиять на статус разметки других контрольных изображений более одного раза в том же заголовке секции. В частности, это разрешено для значения `memory_management_control_operation`, равного 3, определяющего индекс долгосрочного кадра, который следует присвоить конкретному краткосрочному контрольному изображению, следующему в том же заголовке секции за `memory_management_control_operation`, равным 2, 3, 4 или 6, которые определяют это же контрольное изображение, чтобы в дальнейшем пометить его как "не используется для контроля".

Таблица 7-9 – Значения операций управления памятью (memory_management_control_operation)

memory_management_control_operation	Операции управления памятью
0	Конец цикла элементов синтаксиса memory_management_control_operation
1	Разметка краткосрочного контрольного изображения как "не используется для контроля"
2	Разметка долгосрочного контрольного изображения как "не используется для контроля"
3	Разметка краткосрочного контрольного изображения как "используется для долгосрочного контроля" и присвоение ему индекса долгосрочного кадра
4	Определение максимального индекса долгосрочного кадра и разметка всех долгосрочных контрольных изображений, имеющих индексы долгосрочного кадра, превышающие максимальное значение, как "не используется для контроля"
5	Разметка всех контрольных изображений как "не используется для контроля" и установка переменной MaxLongTermFrameIdx в "индексы не долгосрочных кадров"
6	Разметка текущего изображения как "используется для долгосрочного контроля" и присвоение ему индекса долгосрочного кадра

Если выполняется декодирование поля и представлена команда memory_management_control_operation, равная 3, которая присваивает индекс долгосрочного кадра полю, являющемуся частью краткосрочного контрольного кадра или частью пары дополнительных краткосрочных контрольных полей, в структуре синтаксиса разметки того же декодированного контрольного изображения должна быть представлена другая команда memory_management_control_operation для присвоения того же индекса долгосрочного кадра другому полю этого же кадра или пары дополнительных контрольных полей.

ПРИМЕЧАНИЕ 2. – Вышеуказанное требование должно выполняться, даже если поле, на которое указывает команда memory_management_control_operation, значение которой равно 3, размечается далее как "не используется для контроля" (например, если представлена команда memory_management_control_operation, значение которой равно 2, в том же заголовке секции, что вызывает разметку поля как "не используется для контроля").

Если первое поле (в порядке декодирования) пары дополнительных контрольных полей включает значение long_term_reference_flag, равное 1, или команду memory_management_control_operation, равную 6, то структура синтаксиса разметки декодированного контрольного изображения для другого поля пары дополнительных контрольных полей должна содержать команду memory_management_control_operation, равную 6, которая присваивает тот же индекс долгосрочного кадра другому полю.

ПРИМЕЧАНИЕ 3. – Вышеуказанное требование должно выполняться, даже если первое поле пары дополнительных контрольных полей размечается далее как "не используется для контроля" (например, если представлена команда memory_management_control_operation, значение которой равно 2, в заголовке секции второго поля, что вызывает разметку первого поля как "не используется для контроля").

difference_of_pic_nums_minus1 используют (со значением memory_management_control_operation, равным 3 или 1), чтобы присвоить индекс долгосрочного кадра краткосрочному контрольному изображению или чтобы пометить краткосрочное контрольное изображение как "не используется для контроля". Если значение memory_management_control_operation обрабатывают процессом декодирования, то номер результирующего изображения, который находят из difference_of_pic_nums_minus1, должен быть номером изображения, присвоенным одному из контрольных изображений, помеченных как "используется для контроля", а не ранее присвоенному индексу долгосрочного кадра.

Номер результирующего изображения ограничен следующим образом:

- если field_pic_flag равно 0, номер результирующего изображения должен быть одним из установки номеров изображений, присвоенных контрольным кадрам или парам дополнительных контрольных полей;

ПРИМЕЧАНИЕ 4. – Если field_pic_flag равно 0, номер результирующего изображения должен быть номером изображения, присвоенным паре дополнительных контрольных полей, в которых оба поля помечены как "используется для контроля", или кадра, в котором оба поля помечены как "используется для контроля". В частности, если field_pic_flag равно 0, на разметку непарных полей или кадра, в котором единственное поле помечено как "используется для контроля", не может влиять значение memory_management_control_operation, равное 1.

- иначе (field_pic_flag равно 1) номер результирующего изображения должен быть одним из установки номеров изображений, присвоенных контрольным полям.

long_term_pic_num используют (со значением `memory_management_control_operation`, равным 2), чтобы пометить долгосрочное контрольное изображение как "не используется для контроля". Если значение `memory_management_control_operation` обрабатывают процессом декодирования, `long_term_pic_num` должно быть равно номеру долгосрочного изображения, присвоенному одному из контрольных изображений, которое в данный момент помечено как "используется для долгосрочного контроля".

Номер результирующего долгосрочного изображения ограничен следующим образом:

- если `field_pic_flag` равно 0, номер результирующего долгосрочного изображения должен быть одним из установки номеров изображений, присвоенных контрольным кадрам или парам дополнительных контрольных полей;

ПРИМЕЧАНИЕ 5. – Если `field_pic_flag` равно 0, номер результирующего долгосрочного изображения должен быть номером изображения, присвоенным паре дополнительных контрольных полей, в которых оба поля помечены как "используется для контроля", или кадра, в котором оба поля помечены как "используется для контроля". В частности, если `field_pic_flag` равно 0, на разметку непарных полей или кадра, в котором единственное поле помечено как "используется для контроля", не может влиять значение `memory_management_control_operation`, равное 2.

- иначе (`field_pic_flag` равно 1) номер результирующего долгосрочного изображения должен быть одним из установки номеров изображений, присвоенных контрольным полям.

long_term_frame_idx используют (со значением `memory_management_control_operation`, равным 3 или 6), чтобы присвоить индекс долгосрочного кадра изображению. Если `memory_management_control_operation` обрабатывают процессом декодирования, значение `long_term_frame_idx` должно быть в диапазоне от 0 до `MaxLongTermFrameIdx` включительно.

max_long_term_frame_idx_plus1 минус 1 определяет максимальное значение индекса долгосрочного кадра, разрешенного для долгосрочных контрольных изображений (до получения другого значения `max_long_term_frame_idx_plus1`). Значение `max_long_term_frame_idx_plus1` должно быть в диапазоне от 0 до `num_ref_frames` включительно.

7.4.4 Семантика данных секции

cabac_alignment_one_bit – это бит, равный 1.

mb_skip_run определяет число последовательных пропущенных макроблоков, для которых при декодировании секции P или SP значение `mb_type` должно быть обозначено `P_Skip`, а тип макроблока обозначен как макроблок типа P или для которых при декодировании секции B значение `mb_type` должно быть обозначено `B_Skip`, а тип макроблока обозначен как макроблок типа B. Значение `mb_skip_run` должно быть в диапазоне от 0 до `PicSizeInMbs - CurrMbAddr` включительно.

mb_skip_flag, равное 1, определяет, что для данного макроблока при декодировании секции P или SP значение `mb_type` должно быть обозначено `P_Skip`, а тип макроблока обозначен как макроблок типа P, или при декодировании секции B значение `mb_type` должно быть обозначено `B_Skip`, а тип макроблока обозначен как макроблок типа B. Значение `mb_skip_flag`, равное 0, указывает, что текущий макроблок не пропущен.

mb_field_decoding_flag, равное 0, определяет, что текущая пара макроблоков – это пара макроблоков кадра. Значение `mb_field_decoding_flag`, равное 1, определяет, что эта пара макроблоков – пара макроблоков поля. Оба макроблока из пары макроблоков кадра в данном тексте считают макроблоками кадра, несмотря на то что оба макроблока из пары макроблоков поля в данном тексте считают макроблоками поля.

Если значение `mb_field_decoding_flag` отсутствует в любом из пары макроблоков, значение `mb_field_decoding_flag` находят следующим образом:

- если имеется смежная пара макроблоков непосредственно слева от текущей пары в той же секции, значение `mb_field_decoding_flag` должно быть принято равным значению `mb_field_decoding_flag` для смежной пары макроблоков непосредственно слева от текущей пары макроблоков;
- иначе, если в той же секции нет смежной пары макроблоков непосредственно слева от текущей пары, но есть смежная пара макроблоков непосредственно над текущей парой, значение `mb_field_decoding_flag` должно быть принято равным значению `mb_field_decoding_flag` для смежной пары макроблоков непосредственно над текущей парой макроблоков;
- иначе (если в этой же секции нет смежной пары макроблоков непосредственно слева и над текущей парой макроблоков) значение `mb_field_decoding_flag` должно быть принято равным 0.

end_of_slice_flag, равное 0, указывает, что другой макроблок – это следующий в данной секции. Значение `end_of_slice_flag`, равное 1, указывает на конец секции и отсутствие последующих макроблоков.

Функция `NextMbAddress()`, которую используют в таблице синтаксиса данных секции, описана в п. 8.2.2.

7.4.5 Семантика слоя макроблока

mb_type определяет тип макроблока. Семантика `mb_type` зависит от типа секции.

Таблицы и семантика описаны для разных типов макроблоков секций I, SI, P, SP и B. Каждая таблица представляет: значение mb_type, наименование mb_type, число используемых разделенных частей макроблоков (которые дает функция NumMbPart(mb_type), режим предсказания макроблока (если он не разделен) или первого разделения (которые дает функция MbPartPredMode(mb_type, 0) и режим предсказания второго разделения (которые дает функция MbPartPredMode(mb_type, 1). Если значение не применимо, его обозначают "na". В данном тексте значение mb_type можно считать типом макроблока, а значение X в MbPartPredMode() в данном тексте можно считать "режимом предсказания (разделения) макроблока X" или "X предсказанием макроблоков".

В таблице 7-10 показаны разрешенные типы макроблоков для каждого значения slice_type.

ПРИМЕЧАНИЕ 1. – Существуют некоторые типы макроблоков с режимом предсказания Pred_L0, которые относят к типам B макроблоков.

Таблица 7-10 – Разрешенные типы макроблоков для значения slice_type

slice_type	Разрешенные типы макроблоков
I (секция)	I (см. таблицу 7-11) (типы макроблоков)
P (секция)	P (см. таблицу 7-13) и I (см. таблицу 7-11) (типы макроблоков)
B (секция)	B (см. таблицу 7-14) и I (см. таблицу 7-11) (типы макроблоков)
SI (секция)	SI (см. таблицу 7-12) и I (см. таблицу 7-11) (типы макроблоков)
SP (секция)	P (см. таблицу 7-13) и I (см. таблицу 7-11) (типы макроблоков)

transform_size_8x8_flag, равное 1, указывает, что для текущего макроблока должны быть активированы для образцов яркости процесс декодирования коэффициентов преобразования и процесс построения изображения до процесса фильтра устранения блочности для блоков 8x8 остатка. **transform_size_8x8_flag**, равное 0, указывает, что для текущего макроблока должны быть активированы для образцов яркости процесс декодирования коэффициентов преобразования и процесс построения изображения до процесса фильтра устранения блочности для блоков 4x4 остатка. Если в потоке битов **transform_size_8x8_flag** не представлен, его значение должно приниматься равным 0.

ПРИМЕЧАНИЕ 2. – Если режимом предсказания текущего макроблока MbPartPredMode(mb_type, 0) является Intra_16x16, флаг **transform_size_8x8_flag** не присутствует в этом потоке битов и его значение, таким образом, принимается равным 0.

Если sub_mb_type[mbPartIdx] (см. п. 7.4.5.2) представлен в потоке битов для всех блоков 8x8, индексированных mbPartIdx = 0..3, переменная noSubMbPartSizeLessThan8x8Flag показывает, равны ли 8 значения соответствующих SubMbPartWidth(sub_mb_type[mbPartIdx]) и SubMbPartHeight(sub_mb_type[mbPartIdx]) для всех четырех блоков 8x8.

ПРИМЕЧАНИЕ 3. – Если значением noSubMbPartSizeLessThan8x8Flag является 0 и тип текущего макроблока не I_NxN, флаг **transform_size_8x8_flag** не представляется в этом потоке битов и его значение, таким образом, принимается равным 0.

Типы макроблоков, которые можно отнести к макроблокам типа I, показаны в таблице 7-8.

Типы макроблоков для секций I – это все типы макроблоков I.

Таблица 7-11– Типы макроблоков для секций I

mb_type	Наименование mb_type	transform_size_8x8_flag	MbPartPredMode (mb_type, 0)	Intra16x16PredMode	CodedBlockPatternChroma	CodedBlockPatternLuma
0	I_NxN	0	Intra_4x4	na	Уравнение 7-33	Уравнение 7-33
0	I_NxN		Intra_8x8	na	Уравнение 7-33	Уравнение 7-33
1	I_16x16_0_0_0	1	Intra_16x16	0	0	0
2	I_16x16_1_0_0	na	Intra_16x16	1	0	0
3	I_16x16_2_0_0	na	Intra_16x16	2	0	0
4	I_16x16_3_0_0	na	Intra_16x16	3	0	0
5	I_16x16_0_1_0	na	Intra_16x16	0	1	0
6	I_16x16_1_1_0	na	Intra_16x16	1	1	0
7	I_16x16_2_1_0	na	Intra_16x16	2	1	0
8	I_16x16_3_1_0	na	Intra_16x16	3	1	0
9	I_16x16_0_2_0	na	Intra_16x16	0	2	0
10	I_16x16_1_2_0	na	Intra_16x16	1	2	0
11	I_16x16_2_2_0	na	Intra_16x16	2	2	0
12	I_16x16_3_2_0	na	Intra_16x16	3	2	0
13	I_16x16_0_0_1	na	Intra_16x16	0	0	15
14	I_16x16_1_0_1	na	Intra_16x16	1	0	15
15	I_16x16_2_0_1	na	Intra_16x16	2	0	15
16	I_16x16_3_0_1	na	Intra_16x16	3	0	15
17	I_16x16_0_1_1	na	Intra_16x16	0	1	15
18	I_16x16_1_1_1	na	Intra_16x16	1	1	15
19	I_16x16_2_1_1	na	Intra_16x16	2	1	15
20	I_16x16_3_1_1	na	Intra_16x16	3	1	15
21	I_16x16_0_2_1	na	Intra_16x16	0	2	15
22	I_16x16_1_2_1	na	Intra_16x16	1	2	15
23	I_16x16_2_2_1	na	Intra_16x16	2	2	15
24	I_16x16_3_2_1	na	Intra_16x16	3	2	15
25	I_PCM	na	na	na	na	na

Следующая семантика присвоена типам макроблоков в таблице 7-11.

I_NxN: мнемоническое имя mb_type, равного 0 с MbPartPredMode(mb_type, 0), равным Intra_4x4 или Intra_8x8.

I_16x16_0_0_0, I_16x16_1_0_0, I_16x16_2_0_0, I_16x16_3_0_0, I_16x16_0_1_0, I_16x16_1_1_0, I_16x16_2_1_0, I_16x16_3_1_0, I_16x16_0_2_0, I_16x16_1_2_0, I_16x16_2_2_0, I_16x16_3_2_0, I_16x16_0_0_1, I_16x16_1_0_1, I_16x16_2_0_1, I_16x16_3_0_1, I_16x16_0_1_1, I_16x16_1_1_1, I_16x16_2_1_1, I_16x16_3_1_1, I_16x16_0_2_1, I_16x16_1_2_1, I_16x16_2_2_1, I_16x16_3_2_1: макроблок кодирован как макроблок режима предсказания Intra_16x16.

Каждому макроблоку режима предсказания Intra_16x16 присваивают значение Intra16x16PredMode, которое определяет режим предсказания Intra_16x16. CodedBlockPatternChroma содержит значение образца кодированного блока цветности, как описано в таблице 7-15. Если значением chroma_format_idc является 0, значением CodedBlockPatternChroma должен быть 0. CodedBlockPatternLuma определяет, присутствуют ли ненулевые уровни AC коэффициентов преобразования для компонента яркости. CodedBlockPatternLuma, равное 0, указывает, что все уровни AC коэффициентов преобразования в компоненте яркости макроблока равны 0. CodedBlockPatternLuma, равное 15, указывает, что по крайней мере один из уровней AC коэффициентов преобразования в компоненте яркости макроблока – ненулевой, и требуется сканирование уровней AC коэффициентов преобразования для всех 16 блоков 4x4 в блоке 16x16.

Intra_4x4 определяет режим предсказания макроблока, а также указывает, что запущен процесс предсказания Intra_4x4, описанный в п. 8.3.1. Intra_4x4 – это режим Intra предсказания макроблока.

Intra_8x8 определяет режим предсказания макроблока, а также указывает, что запущен процесс предсказания Intra_8x8, описанный в п. 8.3.2. Intra_8x8 – это режим Intra предсказания макроблока.

Intra_16x16 определяет режим предсказания макроблока, а также указывает, что запущен процесс предсказания Intra_16x16, описанный в п. 8.3.3. Intra_16x16 – это режим Intra предсказания макроблока.

Для макроблока, кодированного значением mb_type, равным I_PCM, должен быть принят режим Intra предсказания макроблока.

Тип макроблока, который можно отнести к типу макроблока SI, определяется в таблице 7-12.

Типы макроблоков секций SI определяются в таблице 7-12 и таблице 7-11. Нулевое значение mb_type определяется в таблице 7-12, а значения mb_type от 1 до 26 определены в таблице 7-11, индексированной вычитанием 1 из значения mb_type.

Таблица 7-12 – Нулевые значения типа макроблока для секций SI

mb_type	Наименование mb_type	MbPartPredMode (mb_type, 0)	Intra16x16PredMode	CodedBlockPatternChroma	CodedBlockPatternLuma
0	SI	Intra_4x4	na	Уравнение 7-23	Уравнение 7-23

Следующая семантика присвоена типу макроблока в таблице 7-12. Макроблок SI кодирован как макроблок предсказания Intra_4x4.

Типы макроблоков, которые можно отнести к типу P, показаны в таблице 7-13.

Типы макроблоков для секций P и SP показаны в таблице 7-13 и таблице 7-11. Значения mb_type от 0 до 4 показаны в таблице 7-13, значения mb_type от 5 до 30 показаны в таблице 7-11, индексированной вычитанием 5 из значения mb_type.

Таблица 7-13 – Значения типа макроблока от 0 до 4 для секций P и SP

mb_type	Наименование mb_type	NumMbPart (mb_type)	MbPartPredMode (mb_type, 0)	MbPartPredMode (mb_type, 1)	MbPartWidth (mb_type)	MbPartHeight (mb_type)
0	P_L0_16x16	1	Pred_L0	na	16	16
1	P_L0_L0_16x8	2	Pred_L0	Pred_L0	16	8
2	P_L0_L0_8x16	2	Pred_L0	Pred_L0	8	16
3	P_8x8	4	na	na	8	8
4	P_8x8ref0	4	na	na	8	8
полученные	P_Skip	1	Pred_L0	na	16	16

Следующая семантика присвоена типам макроблоков в таблице 7-13.

- P_L0_16x16: образцы макроблоков предсказаны для одного разделенного макроблока образцов яркости размером 16x16, объединенных с образцами цветности.
- P_L0_L0_MxN, с заменой MxN на 16x8 или 8x16: образцы макроблоков предсказаны с использованием двух разделенных частей яркости размером MxN, равным 16x8, или двух разделенных частей яркости размером MxN, равным 8x16, и соответственно объединенных с образцами цветности.
- P_8x8: представлен в потоке битов для каждого субмакроблока дополнительного элемента синтаксиса (sub_mb_type), который определяет тип соответствующего субмакроблока (см. п. 7.4.5.2).
- P_8x8ref0: представлен в потоке битов. Имеет ту же семантику, что и P_8x8, но без элементов синтаксиса для контроля индекса (ref_idx_10), а ref_idx_10[mbPartIdx] должен считаться равным 0 для всех субмакроблоков этого макроблока (с индексами mbPartIdx, равными 0..3).
- P_Skip: в потоке битов отсутствуют дополнительные данные для этого макроблока.

Следующая семантика присвоена режимам предсказания макроблоков (MbPartPredMode()) из таблицы 7-13.

- Pred_L0: определяет, что запущен процесс Inter предсказания (внешнего предсказания) с использованием списка предсказаний 0. Pred_L0 – это режим предсказания макроблока Inter.

Типы макроблоков, которые в общем можно отнести к типу макроблоков В, описаны в таблице 7-14.

Типы макроблоков для секций В описаны в таблице 7-14 и таблице 7-11. Значения mb_type от 0 до 22 описаны в таблице 7-14, а значения mb_type от 23 до 48 описаны в таблице 7-11, проиндексированной вычитанием 23 из значения mb_type.

Таблица 7-14 – Значения от 0 до 22 типов макроблоков для секций В

mb_type	Наименование mb_type	NumMbPart (mb_type)	MbPartPredMode (mb_type, 0)	MbPartPredMode (mb_type, 1)	MbPartWidth (mb_type)	MbPartHeight (mb_type)
0	B_Direct_16x16	na	Direct	na	8	8
1	B_L0_16x16	1	Pred_L0	na	16	16
2	B_L1_16x16	1	Pred_L1	na	16	16
3	B_Bi_16x16	1	BiPred	na	16	16
4	B_L0_L0_16x8	2	Pred_L0	Pred_L0	16	8
5	B_L0_L0_8x16	2	Pred_L0	Pred_L0	8	16
6	B_L1_L1_16x8	2	Pred_L1	Pred_L1	16	8
7	B_L1_L1_8x16	2	Pred_L1	Pred_L1	8	16
8	B_L0_L1_16x8	2	Pred_L0	Pred_L1	16	8
9	B_L0_L1_8x16	2	Pred_L0	Pred_L1	8	16
10	B_L1_L0_16x8	2	Pred_L1	Pred_L0	16	8
11	B_L1_L0_8x16	2	Pred_L1	Pred_L0	8	16
12	B_L0_Bi_16x8	2	Pred_L0	BiPred	16	8
13	B_L0_Bi_8x16	2	Pred_L0	BiPred	8	16
14	B_L1_Bi_16x8	2	Pred_L1	BiPred	16	8
15	B_L1_Bi_8x16	2	Pred_L1	BiPred	8	16
16	B_Bi_L0_16x8	2	BiPred	Pred_L0	16	8
17	B_Bi_L0_8x16	2	BiPred	Pred_L0	8	16
18	B_Bi_L1_16x8	2	BiPred	Pred_L1	16	8
19	B_Bi_L1_8x16	2	BiPred	Pred_L1	8	16
20	B_Bi_Bi_16x8	2	BiPred	BiPred	16	8
21	B_Bi_Bi_8x16	2	BiPred	BiPred	8	16
22	B_8x8	4	na	na	8	8
полученные	B_Skip	na	Direct	na	8	8

Типам макроблоков в таблице 7-11 присваивают следующую семантику.

- B_Direct_16x16: в потоке битов отсутствуют разности векторов движения или индексов контроля для макроблока. Функции MbPartWidth(B_Direct_16x16) и MbPartHeight(B_Direct_16x16) используют в процессе вывода векторов движения и индексов контрольных кадров (согласно п. 8.4.1) для режима прямого предсказания.
- B_X_16x16 с заменой X на L0, L1 или Bi: образцы макроблока предсказаны с одним разделенным макроблоком яркости с размером образцов яркости 16x16 и объединенным с образцами цветности. Для макроблока типа B_X_16x16 с заменой X либо на L0, либо на L1 в потоке битов макроблока присутствуют одна разность векторов

движения и один индекс контроля. Для макроблока типа $V_X_{16 \times 16}$ с заменой X на V_i в потоке битов макроблока присутствуют две разности векторов движения и два индекса контроля.

- $V_{X0_X1_M \times N}$ с заменой $X0$ и $X1$, которые считают первыми и вторыми разделенными частями макроблока, на $L0$, $L1$ или V_i , а также с заменой $M \times N$ на 16×8 или 8×16 : образцы макроблока предсказаны с использованием двух разделенных частей яркости размером $M \times N$, равным 16×8 , или двух разделенных частей яркости размером $M \times N$, равным 8×16 , и объединенных с образцами цветности соответственно. Для разделенных частей макроблока $X0$ или $X1$ с заменой $X0$ или $X1$ на $L0$ или $L1$ в потоке битов представлены одна разность векторов движения и один индекс контроля. Для разделенных частей макроблока $X0$ или $X1$ с заменой $X0$ или $X1$ на V_i в потоке битов представлены две разности векторов движения и два индекса контроля.
- $V_{8 \times 8}$: в потоке битов для каждого субмакроблока представлен дополнительный элемент синтаксиса (sub_mb_type), который определяет тип соответствующего субмакроблока (см. п. 7.4.5.2).
- V_Skip : в потоке битов отсутствуют данные для этого макроблока. Функции $MbPartWidth(V_Skip)$ и $MbPartHeight(V_Skip)$ используют в процессе вывода векторов движения и индексов контрольных кадров (согласно п. 8.4.1) для режима прямого предсказания.

Следующие элементы семантики присваивают режимам предсказания макроблоков ($MbPartPredMode()$) из таблицы 7-14.

- $Direct$: в потоке битов отсутствуют разности векторов движения или индексов контроля для макроблока (случай V_Skip или $V_Direct_{16 \times 16}$). $Direct$ – это режим $Inter$ предсказания макроблока.
- $Pred_L0$: см. семантику для таблицы 7-13.
- $Pred_L1$: определяет, что запущен процесс $Inter$ предсказания с использованием списка предсказаний 1. $Pred_L1$ – это режим $Inter$ предсказания макроблока.
- $BiPred$: определяет, что запущен процесс $Inter$ предсказания с использованием списков предсказаний 0 и 1. $BiPred$ – это режим $Inter$ предсказания макроблока.

pcm_alignment_zero_bit – бит, равный 0.

pcm_sample_luma[i [i]] – значение образца. Первые значения pcm_sample_luma [i [i]] представляют значения образцов яркости при растровом сканировании макроблока. Число битов, используемых для представления каждого из этих образцов, равно $BitDepth_Y$. Если $profile_idc$ не равно 100, 110, 122 или 144, pcm_sample_luma [i] не должно быть равно 0.

pcm_sample_chroma[i] – значение образца. Первые значения $MbWidthC * MbHeightC$ pcm_sample_chroma [i] представляют значения образцов Cb при растровом сканировании макроблока, а остальные значения $MbWidthC * MbHeightC$ pcm_sample_chroma [i] представляют значения образцов Cr при растровом сканировании макроблока. Число битов, используемых для представления каждого из этих образцов, равно $BitDepth_C$. Если $profile_idc$ не равно 100, 110, 122 или 144, pcm_sample_chroma [i] не должно быть равно 0.

coded_block_pattern определяет, какой из четырех блоков яркости 8×8 и связанных блоков цветности макроблока может содержать ненулевые уровни коэффициентов преобразования. Для макроблоков с режимом предсказания, не равным $Intra_{16 \times 16}$, образец $coded_block_pattern$ представлен в потоке битов, а переменные $CodedBlockPatternLuma$ и $CodedBlockPatternChroma$ находят следующим образом:

$$\begin{aligned} CodedBlockPatternLuma &= coded_block_pattern \% 16, \\ CodedBlockPatternChroma &= coded_block_pattern / 16. \end{aligned} \quad (7-33)$$

Если $coded_block_pattern$ представлен, переменная $CodedBlockPatternLuma$ выбирает для каждого из четырех блоков яркости 8×8 макроблока один из следующих случаев:

- все уровни коэффициентов преобразования четырех блоков яркости 4×4 в блоке яркости 8×8 равны нулю;
- один или более уровней коэффициентов преобразования одного из четырех блоков яркости 4×4 в блоке яркости 8×8 должны иметь ненулевые значения.

Значения $CodedBlockPatternChroma$ показаны в таблице 7-15.

Таблица 7-15 – Спецификация значений CodedBlockPatternChroma

CodedBlockPatternChroma	Описание
0	Все уровни коэффициентов преобразования цветности равны 0.
1	Один или более уровней коэффициентов преобразования цветности DC должны иметь ненулевые значения. Все уровни коэффициентов преобразования цветности AC равны 0.
2	Ноль или более уровней коэффициентов преобразования цветности DC имеют ненулевые значения. Один или более уровней коэффициентов преобразования цветности AC должны иметь ненулевые значения.

mb_qp_delta может менять значение QP_Y в слое макроблока. Декодированное значение **mb_qp_delta** должно быть в диапазоне от $-(26 + QpBdOffset_Y / 2)$ до $+(25 + QpBdOffset_Y / 2)$ включительно. Значение **mb_qp_delta** должно быть принято равным 0, если оно не представлено ни для одного макроблока (включая типы макроблоков P_Skip и B_Skip).

Значение QP_Y находят как:

$$QP_Y = ((QP_{Y,PREV} + mb_qp_delta + 52 + 2 * QpBdOffset_Y) \% (52 + QpBdOffset_Y)) - QpBdOffset_Y, \quad (7-34)$$

где $QP_{Y,PREV}$ – параметр квантования яркости, а QP_Y взято из предыдущего макроблока в порядке декодирования в текущей секции. Для первого макроблока в секции $QP_{Y,PREV}$ представляет первоначальную установку, равную $SliceQP_Y$, найденную из уравнения 7-27 в начале каждой секции.

Значение QP'_Y определяют следующим образом:

$$QP'_Y = QP_Y + QpBdOffset_Y. \quad (7-35)$$

7.4.5.1 Семантика предсказания макроблока

Все образцы макроблока предсказаны. Режимы предсказания находят, используя следующие элементы синтаксиса.

prev_intra4x4_pred_mode_flag[luma4x4BlkIdx] и **rem_intra4x4_pred_mode[luma4x4BlkIdx]** определяют предсказание Intra_4x4 блока яркости 4x4 с индексом $luma4x4BlkIdx = 0..15$.

prev_intra8x8_pred_mode_flag[luma8x8BlkIdx] и **rem_intra8x8_pred_mode[luma8x8BlkIdx]** определяют предсказание Intra_8x8 блока яркости 8x8 с индексом $luma8x8BlkIdx = 0..3$.

intra_chroma_pred_mode определяет тип пространственного предсказания, использованного для цветности в макроблоках с предсказанием Intra_4x4 или Intra_16x16, как показано в таблице 7-16. Значение **intra_chroma_pred_mode** должно находиться в диапазоне от 0 до 3 включительно.

Таблица 7-16 – Взаимосвязь между режимами пространственных предсказаний **intra_chroma_pred_mode**

intra_chroma_pred_mode	Режим Intra предсказания цветности
0	DC
1	Горизонтальное
2	Вертикальное
3	Плоское

ref_idx_10[mbPartIdx], если представлено, определяет индекс в списке контрольных изображений 0 контрольного изображения, которое используют для предсказания.

Диапазон **ref_idx_10[mbPartIdx]**, индекс в списке 0 контрольных изображений и, если применимо, четность поля в контрольном изображении, использованного для предсказания, описаны следующим образом:

- если **MbaffFrameFlag** равно 0 или **mb_field_decoding_flag** равно 0, то значение **ref_idx_10[mbPartIdx]** должно быть в диапазоне от 0 до **num_ref_idx_10_active_minus1** включительно;

- иначе (MbaffFrameFlag равно 1 и mb_field_decoding_flag равно 1) значение ref_idx_10[mbPartIdx] должно быть в диапазоне от 0 до 2 * num_ref_idx_10_active_minus1 + 1 включительно.

Если для внешнего предсказания использовано только одно контрольное изображение, то значение ref_idx_10[mbPartIdx] должно быть принято равным 0.

ref_idx_11[mbPartIdx] имеет ту же семантику, что и ref_idx_10 с заменой 10 и списка 0 на 11 и список 1 соответственно.

mvd_10[mbPartIdx][0][compIdx] определяет разность между компонентом вектора, который должен использоваться, и его предсказанием. Индекс mbPartIdx определяет, какой разделенной части макроблока присвоено значение mvd_10. Разделение макроблока описано значением mb_type. Разность горизонтальных компонентов вектора движения декодируют первой в порядке декодирования и присваивают значение CompIdx = 0. Вертикальный компонент вектора движения декодируют вторым в порядке декодирования и присваивают значение CompIdx = 1. Диапазон компонентов mvd_10[mbPartIdx][0][compIdx] определен ограничениями на вектор движения переменным значением, полученным из этого диапазона, как это указано в Приложении А.

mvd_11[mbPartIdx][0][compIdx] имеет ту же семантику, что и mvd_10 с заменой 10 и L0 на 11 и L1 соответственно.

7.4.5.2 Семантика предсказания субмакроблока

sub_mb_type[mbPartIdx] определяет подтип макроблоков.

Таблицы и семантика описаны для различных подтипов макроблоков секций макроблоков типа P и B. Каждая таблица представляет значение sub_mb_type, наименование sub_mb_type, число используемых разделенных частей субмакроблока (заданных функцией NumSubMbPart(sub_mb_type) и режимом предсказания субмакроблока (заданным функцией SubMbPredMode(sub_mb_type)). В данном тексте значение sub_mb_type можно рассматривать как "подтип макроблока", а значение SubMbPredMode() – как "субрежим предсказания макроблока".

Интерпретация sub_mb_type[mbPartIdx] для макроблока типа P определена в таблице 7-17, где строка "принимается равным" определяет значения, которые принимаются, если не представлен тип sub_mb_type[mbPartIdx].

Таблица 7-17 – Подтип макроблоков для макроблоков типа P

sub_mb_type[mbPartIdx]	Наименование sub_mb_type[mbPartIdx]	NumSubMbPart (sub_mb_type[mbPartIdx])	SubMbPredMode (sub_mb_type[mbPartIdx])	SubMbPartWidth (sub_mb_type[mbPartIdx])	SubMbPartHeight (sub_mb_type[mbPartIdx])
Принимается равным	na	na	na	na	na
0	P_L0_8x8	1	Pred_L0	8	8
1	P_L0_8x4	2	Pred_L0	8	4
2	P_L0_4x8	2	Pred_L0	4	8
3	P_L0_4x4	4	Pred_L0	4	4

Следующую семантику присваивают подтипам макроблоков в таблице 7-17.

- P_L0_MxN, с заменой MxN на 8x8, 8x4, 4x8 и 4x4: образцы субмакроблока, предсказанные с использованием одной разделенной части яркости размером MxN, равным 8x8, двух разделенных частей яркости размером MxN, равным 8x4, или двух разделенных частей яркости размером MxN, равным 4x8, или четырех разделенных частей яркости размером MxN, равным 4x4, и соответственно объединенных с образцами цветности.

Следующую семантику присваивают субрежимам предсказания макроблоков (SubMbPredMode()) в таблице 7-17.

- Pred_L0: см. семантику для таблицы 7-13.

Интерпретация sub_mb_type[mbPartIdx] для макроблоков типа B определяется в таблице 7-18, где строка "принимается равным" определяет значения, которые принимаются, если не представлен sub_mb_type[mbPartIdx],

а принятое значение "mb_type" указывает, что для этого случая имя sub_mb_type[mbPartIdx] то же, что и имя mb_type.

Таблица 7-18 – Подтипы макроблоков для макроблоков типа В

sub_mb_type[mbPartIdx]	Наименование sub_mb_type[mbPartIdx]	NumSubMbPart (sub_mb_type[mbPartIdx])	SubMbPredMode (sub_mb_type[mbPartIdx])	SubMbPartWidth (sub_mb_type[mbPartIdx])	SubMbPartHeight (sub_mb_type[mbPartIdx])
Принимается равным	mb_type	4	Direct	4	4
0	B_Direct_8x8	4	Direct	4	4
1	B_L0_8x8	1	Pred_L0	8	8
2	B_L1_8x8	1	Pred_L1	8	8
3	B_Bi_8x8	1	BiPred	8	8
4	B_L0_8x4	2	Pred_L0	8	4
5	B_L0_4x8	2	Pred_L0	4	8
6	B_L1_8x4	2	Pred_L1	8	4
7	B_L1_4x8	2	Pred_L1	4	8
8	B_Bi_8x4	2	BiPred	8	4
9	B_Bi_4x8	2	BiPred	4	8
10	B_L0_4x4	4	Pred_L0	4	4
11	B_L1_4x4	4	Pred_L1	4	4
12	B_Bi_4x4	4	BiPred	4	4

Следующую семантику присваивают типам субмакроблоков по таблице 7-18.

- B_Skip и B_Direct_16x16: в потоке битов отсутствуют разности векторов движения или индексов контроля для субмакроблока. Функции SubMbPartWidth() и SubMbPartHeight() используют в процессе вывода векторов движения и индексов контрольных кадров (согласно п. 8.4.1) для режима прямого предсказания.
- B_Direct_8x8: в потоке битов отсутствуют разности векторов движения или индексов контроля для субмакроблока. Функции SubMbPartWidth(B_Direct_8x8) и SubMbPartHeight(B_Direct_8x8) используют в процессе вывода векторов движения и индексов контрольных кадров (согласно п. 8.4.1) для режима прямого предсказания.
- B_X_MxN, с заменой X на L0, L1 и Bi, а MxN на 8x8, 8x4, 4x8 или 4x4: образцы субмакроблока предсказаны с использованием одной разделенной части яркости размером MxN, равным 8x8, или образцы субмакроблока предсказаны с использованием двух разделенных частей яркости размером MxN, равным 8x4, или образцы субмакроблока предсказаны с использованием двух разделенных частей яркости размером MxN, равным 4x8, или образцы субмакроблока предсказаны с использованием четырех разделенных частей яркости размером MxN, равным 4x4, и соответственно объединенных с образцами цветности. Все части субмакроблока имеют один и тот же индекс контроля. Для разделенной части субмакроблока MxN в субмакроблоке со значением sub_mb_type типа B_X_MxN с заменой X на L0 или L1 в потоке битов существует одна разность векторов движения. Для разделенной части субмакроблока MxN в субмакроблоке со значением sub_mb_type типа B_Bi_MxN в потоке битов существует две разности векторов движения.

Следующую семантику присваивают субрежимам предсказания макроблоков (SubMbPredMode()) в таблице 7-18.

- Direct: см. семантику для таблицы 7-14.
- Pred_L0: см. семантику для таблицы 7-13.
- Pred_L1: см. семантику для таблицы 7-14.
- BiPred: см. семантику для таблицы 7-14.

ref_idx_10[mbPartIdx] имеет ту же семантику, что и ref_idx_10 в п. 7.4.5.1.

ref_idx_11[mbPartIdx] имеет ту же семантику, что и ref_idx_11 в п. 7.4.5.1.

mvd_10[mbPartIdx][subMbPartIdx][compIdx] имеет ту же семантику, что и mvd_10 в п. 7.4.5.1, за исключением того, что это применяют к индексам разделения субмакроблока со значением subMbPartIdx. Индексы mbPartIdx и subMbPartIdx определяют, каким разделенным частям макроблока и субмакроблока присвоено значение mvd_10.

mvd_11[mbPartIdx][subMbPartIdx][compIdx] имеет ту же семантику, что и mvd_11 в п. 7.4.5.1.

7.4.5.3 Семантика данных остатка

Структуру синтаксиса residual_block (), которую используют для анализа уровней коэффициентов преобразования, присваивают следующим образом:

- если значение entropy_coding_mode_flag равно 0, то residual_block устанавливают равным значению residual_block_cavlc, которое используют для выделения элементов синтаксиса для уровней коэффициентов преобразования;
- иначе (entropy_coding_mode_flag равно 1) residual_block устанавливают равным значению residual_block_cabac, которое используют для выделения элементов синтаксиса для уровней коэффициентов преобразования.

В зависимости от mb_type, яркости или цветности и формата цветности структуру синтаксиса residual_block (coeffLevel, maxNumCoeff) используют с аргументом coeffLevel, который представляет список, содержащий уровни коэффициентов преобразования maxNumCoeff, которые выделяются в блок residual_block() и в maxNumCoeff следующим образом.

- В зависимости от MbPartPredMode(mb_type, 0) применяют следующее.
 - Если MbPartPredMode(mb_type, 0) равно Intra_16x16, то уровни коэффициентов преобразования выделяются в список Intra16x16DCLevel и в 16 списков Intra16x16ACLevel[i]. Intra16x16DCLevel содержит 16 уровней коэффициентов преобразования DC с уровнями для каждого блока яркости 4x4. Для каждого из 16 блоков яркости 4x4, индексированных с $i = 0..15$, 15 уровней коэффициентов преобразования AC i -ого блока выделяются в i -ый список Intra16x16ACLevel[i].
 - Иначе (MbPartPredMode(mb_type, 0) не равно Intra_16x16) применяется следующее.
 - Если значение transform_size_8x8_flag равно 0, для каждого из 16 блоков яркости 4x4, индексированных с $i = 0..15$, 16 уровней коэффициентов преобразования i -ого блока выделяются в i -ый список LumaLevel[i].
 - Иначе (transform_size_8x8_flag равно 1) для каждого из 4 блоков яркости 8x8, индексированных с $i8x8 = 0..3$, применяется следующее:
 - если значение entropy_coding_mode_flag равно 0, сначала для каждого из 4 блоков яркости 4x4, индексированных с $i4x4 = 0..3$, 16 уровней коэффициентов преобразования $i4x4$ -ого блока выделяются в $(i8x8 * 4 + i4x4)$ -ый список LumaLevel[$i8x8 * 4 + i4x4$]. Затем 64 уровня коэффициентов преобразования $i8x8$ -ого блока яркости 8x8, индексированного с $4 * i + i4x4$, где $i = 0..15$ и $i4x4 = 0..3$, определяются как $LumaLevel8x8[i8x8][4 * i + i4x4] = LumaLevel[i8x8 * 4 + i4x4][i]$;
- ПРИМЕЧАНИЕ.– Предполагается, что блоки яркости 4x4 с luma4x4BlkIdx = $i8x8 * 4 + i4x4$, содержащие каждый четвертый уровень коэффициентов преобразования соответствующего $i8x8$ -ого блока яркости 8x8 со сдвигом $i4x4$, представляют пространственное местоположение, выдаваемое процессом инверсного сканирования блока яркости 4x4, см. п. 6.4.3.
- иначе (entropy_coding_mode_flag равно 1) 64 уровня коэффициентов преобразования $i8x8$ -ого блока выделяются в $i8x8$ -ый список LumaLevel8x8[$i8x8$].
- Для каждого компонента цветности, индексированного с iCbCr = 0..1, уровни коэффициентов преобразования $4 * NumC8x8$ блоков цветности 4x4 выделяются в iCbCr-ый список ChromaDCLevel[iCbCr].
- Для каждого из блоков цветности 4x4, индексированных с $i4x4 = 0..3$ и $i8x8 = 0..NumC8x8 - 1$, каждого компонента цветности, индексированного с iCbCr = 0..1, 15 уровней коэффициентов преобразования AC выделяются в $(i8x8 * 4 + i4x4)$ -ый список iCbCr-ого компонента цветности ChromaACLevel[iCbCr][$i8x8 * 4 + i4x4$].

7.4.5.3.1 Семантика остатка блока CAVLC

Функция `TotalCoeff(coeff_token)`, которую используют в п. 7.3.5.3.1, возвращает число ненулевых уровней коэффициентов преобразования, полученных из `coeff_token`.

Функция `TrailingOnes(coeff_token)`, которую используют в п. 7.3.5.3.1, возвращает концевые единицы, полученные из `coeff_token`.

coeff_token определяет общее число ненулевых уровней коэффициентов преобразования и число концевых единиц уровней коэффициентов преобразования при сканировании уровней коэффициентов преобразования. Уровень коэффициентов преобразования концевой единицы – это один из трех последовательных ненулевых уровней коэффициентов преобразования, которые имеют абсолютное значение, равное 1, в конце сканированных ненулевых уровней коэффициентов преобразования. Диапазон `coeff_token` описан в п. 9.2.1.

trailing_ones_sign_flag определяет знак уровня коэффициента преобразования концевой единицы следующим образом:

- если `trailing_ones_sign_flag` равно 0, то соответствующий уровень коэффициентов преобразования будет декодирован как +1;
- иначе (`trailing_ones_sign_flag` равно 1) соответствующий уровень коэффициентов преобразования будет декодирован как –1.

level_prefix и **level_suffix** определяет значение ненулевого уровня коэффициентов преобразования. Диапазон `level_prefix` и `level_suffix` описан в п. 9.2.2.

total_zeros определяет общее число уровней коэффициентов преобразования с нулевыми значениями, которые расположены перед позицией последнего ненулевого уровня коэффициентов преобразования при сканировании уровней коэффициентов преобразования. Диапазон `total_zeros` описан в п. 9.2.3.

run_before определяет число последовательных уровней коэффициентов преобразования при сканировании с нулевыми значениями перед уровнем коэффициентов преобразования с ненулевыми значениями. Диапазон `run_before` описан в п. 9.2.3.

coeffLevel содержит `maxNumCoeff` уровней коэффициентов преобразования для текущего списка уровней коэффициентов преобразования.

7.4.5.3.2 Семантика остатка блока CABAC

coded_block_flag определяет, содержит ли блок ненулевые уровни коэффициентов преобразования, следующим образом:

- если `coded_block_flag` равно 0, блок не содержит ненулевые уровни коэффициентов преобразования;
- иначе (`coded_block_flag` равно 1) блок содержит по крайней мере один ненулевой уровень коэффициентов преобразования.

significant_coeff_flag[i] определяет, является ли ненулевым уровень коэффициентов преобразования на позиции сканирования `i`, следующим образом:

- если `significant_coeff_flag[i]` равно 0, уровень коэффициентов преобразования на позиции сканирования `i` установлен равным 0;
- иначе (`significant_coeff_flag[i]` равно 1) уровень коэффициентов преобразования на позиции сканирования `i` имеет ненулевое значение.

last_significant_coeff_flag[i] определяет для позиции сканирования `i`, являются ли ненулевыми уровни коэффициентов преобразования для последующих позиций сканирования от `i + 1` до `maxNumCoeff – 1`, следующим образом:

- если `last_significant_coeff_flag[i]` равно 1, все уровни, следующие за уровнями коэффициентов преобразования (в порядке сканирования) блока, имеют значения, равные 0;
- иначе (`last_significant_coeff_flag[i]` равно 0) на пути сканирования имеются дополнительные ненулевые уровни коэффициентов преобразования.

coeff_abs_level_minus1[i] – это абсолютное значение уровня коэффициентов преобразования минус 1. Значение `coeff_abs_level_minus1` ограничено пределами, указанными в п. 8.5.

coeff_sign_flag[i] определяет знак уровня коэффициентов преобразования следующим образом:

- если `coeff_sign_flag` равно 0, соответствующий уровень коэффициентов преобразования имеет положительное значение;
- иначе (`coeff_sign_flag` равно 1) соответствующий уровень коэффициентов преобразования имеет отрицательное значение.

coeffLevel содержит `maxNumCoeff` уровней коэффициентов преобразования для текущего списка уровней коэффициентов преобразования.

8 Процесс декодирования

Выходами этого процесса являются декодированные образцы текущего изображения (иногда ссылаются на переменную `CurrPic`).

Этот раздел описывает процесс декодирования, заданные элементы синтаксиса и обозначенные заглавными буквами переменные из раздела 7.

Процесс декодирования определен таким образом, что все декодеры должны обеспечить идентичные результаты. Любой процесс декодирования, который обеспечивает идентичные результаты для описываемого здесь процесса, соответствует требованиям к процессу декодирования настоящей Рекомендации | Международного стандарта.

Каждое изображение в этом разделе считают исходным. Каждую секцию в этом разделе считают секцией исходного изображения. Каждую секцию разделенных данных в этом разделе считают секцией разделенных данных исходного изображения.

Обзор процесса декодирования задан следующим образом.

- Декодирование блоков NAL определено в п. 8.1.
- Процессы в п. 8.2 определяют процессы декодирования, используя элементы синтаксиса в этом слое секции и выше.
 - Переменные и функции, связанные с порядком счета изображений находят в п. 8.2.1. (необходимые только для активизации одной секции изображения).
 - Переменные и функции, связанные с макроблоком при отображении группы секции, находят в п. 8.2.2. (необходимые только для активизации одной секции изображения).
 - Метод объединения различных разделенных частей в тех случаях, когда секцию разделенных данных используют, как описано в п. 8.2.3.
 - Если `frame_num` текущего изображения не равно `PrevRefFrameNum` и не равно $(PrevRefFrameNum + 1) \% MaxFrameNum$, процесс декодирования для промежутков в `frame_num` выполняется согласно п. 8.2.5.2 до декодирования какой-либо секции текущего изображения.
 - В начале процесса декодирования каждой секции P, SP или B выполняется процесс декодирования для создания списков контрольных изображений, описанный в п. 8.2.4, для определения списка контрольных изображений 0 (`RefPicList0`), а в случае декодирования секции B – списка контрольных изображений 1 (`RefPicList1`).
 - Если текущее изображение является контрольным, то после того, как все секции текущего изображения декодированы, процесс разметки декодированного контрольного изображения (согласно п. 8.2.5) определит, как использовано в процессе декодирования текущее изображение для внешнего предсказания последующих декодированных изображений.
- Процессы в пп. 8.3, 8.4, 8.5, 8.6 и 8.7 определяют процессы декодирования, используя элементы синтаксиса в слое этого макроблока и выше.
 - Процесс внутреннего предсказания для макроблоков I и SI, за исключением макроблоков I_PCM, который описан в п. 8.3, содержит на выходе образцы `intra` предсказания (внутреннего предсказания). Для макроблоков I_PCM в п. 8.3 непосредственно определен процесс построения изображения. Выходом являются образцы, построенные до процесса фильтра устранения блочности.
 - Процесс `inter` предсказания (внешнего предсказания) для макроблоков P и B определен в п. 8.4 с образцами внешнего предсказания в качестве выхода.
 - Процесс декодирования коэффициентов преобразования и процесс построения изображения до процесса фильтра устранения блочности (разблокирования) определены в п. 8.5. С помощью этих процессов находят образцы для макроблоков I и B и для макроблоков P и секций P. Выходом являются образцы, построенные до процесса фильтра устранения блочности.
 - Процесс декодирования для макроблоков P в секциях SP или для макроблоков SI определен в п. 8.6. С помощью этих процессов находят образцы для макроблоков P в секциях SP и для макроблоков SI. Выходом являются образцы, построенные до процесса фильтра устранения блочности.
 - Образцы, построенные до процесса фильтра устранения блочности, который является следующим для краев блоков и макроблоков, обрабатывают фильтром устранения блочности, который описан в п. 8.7, с декодированными образцами на выходе.

8.1 Процесс декодирования блока NAL

Входами в этот процесс являются блоки NAL.

Выходами этого процесса являются структуры синтаксиса RBSP, инкапсулированные в блоки NAL.

Процесс декодирования каждого блока NAL извлекает структуру синтаксиса RBSP из блока NAL и затем выполняет процесс декодирования, определенный для структуры синтаксиса RBSP в блоке NAL следующим образом.

Пункт 8.2 описывает процесс декодирования для блоков NAL с `nal_unit_type`, равным от 1 до 5.

Пункт 8.3 описывает процесс декодирования для макроблока или части макроблока, кодированных в блоках NAL с `nal_unit_type`, равным 1, 2 и 5.

Пункт 8.4 описывает процесс декодирования для макроблока или части макроблока, кодированных в блоках NAL с `nal_unit_type`, равным 1 и 2.

Пункт 8.5 описывает процесс декодирования для макроблока или части макроблока, кодированных в блоках NAL с `nal_unit_type`, равным 1 и от 3 до 5.

Пункт 8.6 описывает процесс декодирования для макроблока или части макроблока, кодированных в блоках NAL с `nal_unit_type`, равным 1 и от 3 до 5.

Пункт 8.7 описывает процесс декодирования для макроблока или части макроблока, кодированных в блоках NAL с `nal_unit_type`, равным от 1 до 5.

Блоки NAL с `nal_unit_type`, равным 7 и 8, содержат наборы параметров последовательности и наборы параметров изображения соответственно. Наборы параметров изображения используют в процессах декодирования других блоков NAL, как это определено ссылкой на набор параметров изображения в заголовках секций каждого изображения. Наборы параметров последовательности используют в процессах декодирования других блоков NAL, как это определено ссылкой на набор параметров последовательности в наборах параметров изображения каждой последовательности.

Не определен нормативный процесс декодирования для блоков NAL с `nal_unit_type`, равным 6, 9, 10, 11 и 12.

8.2 Процесс декодирования секции

8.2.1 Процесс декодирования для вычисления порядка изображений

Выходами этого процесса являются `TopFieldOrderCnt` (если применяют) и `BottomFieldOrderCnt` (если применяют).

Вычисления порядков изображений используют, чтобы определить для контрольных изображений порядки первоначальных изображений при декодировании В секций (см. пп. 8.2.4.2.3 и 8.2.4.2.4), представить разности порядков изображений между кадрами и полями для вывода вектора движения во временном прямом режиме (см. п. 8.4.1.2.3), для предполагаемого режима взвешивания предсказания в В секциях (см. п. 8.4.2.3.2) и для контроля согласованности декодера (см. п. С.4).

Информацию о вычислении порядка изображений находят для каждого кадра, поля (либо декодированного из кодированного поля, либо как часть декодированного кадра) или пары дополнительных полей следующим образом:

- каждый кодированный кадр связан с двумя вычислениями порядков изображения, которые называют `TopFieldOrderCnt` и `BottomFieldOrderCnt` для верхнего и нижнего поля соответственно;
- каждое кодированное поле связано с вычислением порядков изображения, которые называют `TopFieldOrderCnt` для верхнего кодированного поля и `BottomFieldOrderCnt` для нижнего поля;
- каждая пара дополнительных полей связана с двумя вычислениями порядков изображения, которые называют `TopFieldOrderCnt` для верхнего кодированного поля и `BottomFieldOrderCnt` для нижнего поля соответственно.

`TopFieldOrderCnt` и `BottomFieldOrderCnt` указывают на порядок изображения соответствующего верхнего или нижнего поля относительно первого поля на выходе от предыдущего IDR изображения или от предыдущего контрольного изображения, включая значение `memory_management_control_operation`, равное 5, в порядке декодирования.

`TopFieldOrderCnt` и `BottomFieldOrderCnt` находят, активируя процессы декодирования для вычисления типа порядка изображений 0, 1 и 2 согласно пп. 8.2.1.1, 8.2.1.2 и 8.2.1.3, соответственно. Если текущее изображение включает операцию управления памятью, равную 5, то после декодирования текущего изображения `tempPicOrderCnt` устанавливают равным `PicOrderCnt(CurrPic)`, `TopFieldOrderCnt` текущего изображения (если имеется) устанавливают равным `TopFieldOrderCnt – tempPicOrderCnt`, а `BottomFieldOrderCnt` текущего изображения (если имеется) устанавливают равным `BottomFieldOrderCnt – tempPicOrderCnt`.

Поток битов не должен содержать данных, которые возникают в `Min(TopFieldOrderCnt, BottomFieldOrderCnt)`, не равном 0 для кодированного IDR кадра, `TopFieldOrderCnt – не равном 0` для кодированного верхнего IDR поля или `BottomFieldOrderCnt – не равном 0` для кодированного нижнего IDR поля. Следовательно, по крайней мере одно из значений `TopFieldOrderCnt` и `BottomFieldOrderCnt` должно быть равным 0 для полей кодированного IDR кадра.

Если текущее изображение – не IDR изображение, используют следующие условия.

- Рассмотрим список переменных `listD`, содержащий значения `TopFieldOrderCnt` и `BottomFieldOrderCnt` как элементы, связанные со списком изображений, включая все из следующего.
 - Первое изображение в списке – это предыдущее изображение любого из следующих типов:
 - IDR изображение;
 - изображение, содержащее значение `memory_management_control_operation`, равное 5.
 - Следующие дополнительные изображения:
 - если `pic_order_cnt_type` равно 0, то все прочие изображения, которые следуют в порядке декодирования за первым изображением в списке и не являются "несуществующими" кадрами, которые процесс декодирования посчитал промежутками в значении `frame_num`, определенном в п. 8.2.5.2, либо предшествуют текущему изображению в порядке декодирования, либо являются текущими изображениями. Если `pic_order_cnt_type` равно 0, все прочие изображения, которые следуют в порядке

декодирования за первым изображением в списке и не являются "несуществующими" кадрами, которые процесс декодирования посчитал промежутками в значении `frame_num`, определенном в п. 8.2.5.2, то текущее изображение включают в `listD` до того, как начнет действовать процесс разметки декодированного контрольного изображения;

- иначе (`pic_order_cnt_type` не равно 0) все прочие изображения, которые следуют в порядке декодирования за первым изображением в списке и либо предшествуют текущему изображению в порядке декодирования, либо являются текущими изображениями. Если `pic_order_cnt_type` не равно 0, текущее изображение включают в `listD` до того, как начнет действовать процесс разметки декодированного контрольного изображения.
- Рассмотрим список переменной `listO`, который содержит элементы `listD`, отсортированные в возрастающем порядке. `listO` не должен содержать ничего из следующего:
 - пару `TopFieldOrderCnt` и `BottomFieldOrderCnt` для кадра или пары дополнительных полей, которые не занимают последовательных позиций в `listO`;
 - `TopFieldOrderCnt`, который принимает значение, равное другому `TopFieldOrderCnt`;
 - `BottomFieldOrderCnt`, который принимает значение, равное другому `BottomFieldOrderCnt`;
 - `BottomFieldOrderCnt`, который принимает значение, равное `TopFieldOrderCnt`, до тех пор, пока `BottomFieldOrderCnt` и `TopFieldOrderCnt` принадлежат тому же кодированному кадру или паре дополнительных полей.

Поток битов не должен содержать данные, полученные в значениях `TopFieldOrderCnt`, `BottomFieldOrderCnt`, `PicOrderCntMsb` или `FrameNumOffset`, использованные в процессе декодирования, который описан в пп. 8.2.1.1–8.2.1.3, и которые выходят за диапазон значений от -2^{31} до $2^{31} - 1$ включительно.

Функция `PicOrderCnt(picX)` определена следующим образом:

```

if ( picX – это кадр или пара дополнительных полей )
    PicOrderCnt( picX ) = Min( TopFieldOrderCnt, BottomFieldOrderCnt ) кадра или пары дополнительных
полей picX
else if( picX – это верхнее поле )
    PicOrderCnt( picX ) = TopFieldOrderCnt поля picX
else if( picX – это нижнее поле )
    PicOrderCnt( picX ) = BottomFieldOrderCnt поля picX.

```

(8-1)

Далее `DiffPicOrderCnt(picA, picB)` определено следующим образом:

$$\text{DiffPicOrderCnt}(\text{picA}, \text{picB}) = \text{PicOrderCnt}(\text{picA}) - \text{PicOrderCnt}(\text{picB}).$$

(8-2)

Поток битов не должен содержать данные, полученные в значениях `DiffPicOrderCnt(picA, picB)`, используемых в процессе декодирования, которые выходят за пределы диапазона от -2^{15} до $2^{15} - 1$ включительно.

ПРИМЕЧАНИЕ 1. – Примем, что `X` – текущее изображение, а `Y` и `Z` – два другие изображения в той же последовательности. `Y` и `Z` рассматриваем как имеющие тот же порядок на выходе от `X`, если оба значения, `DiffPicOrderCnt(X, Y)` и `DiffPicOrderCnt(X, Z)` – положительные или отрицательные.

ПРИМЕЧАНИЕ 2. – Многие приложения присваивают `PicOrderCnt(X)` значение, пропорциональное времени выборки изображения `X` относительно времени выборки IDR изображения.

Если текущее изображение включает `memory_management_control_operation`, равное 5, то `PicOrderCnt(CurrPic)` должно быть больше, чем `PicOrderCnt` (любого другого изображения в `listD`).

8.2.1.1 Процесс декодирования для вычисления порядка изображений типа 0

Этот процесс запускают, если `pic_order_cnt_type` равно 0.

Входом в этот процесс является значение `PicOrderCntMsb` предыдущего контрольного изображения в порядке декодирования, как определено в этом пункте.

Выходом этого процесса является одно или оба значения `TopFieldOrderCnt` или `BottomFieldOrderCnt`.

Переменные `prevPicOrderCntMsb` и `prevPicOrderCntLsb` находят следующим образом.

- Если текущее изображение – IDR изображение, `prevPicOrderCntMsb` устанавливают равным 0 и `prevPicOrderCntLsb` устанавливают равным 0.
- Иначе (текущее изображение – не IDR изображение) используют следующие условия.

- Если предыдущее контрольное изображение в порядке декодирования включало `memory_management_control_operation`, равное 5, используют следующие условия:
 - если предыдущее контрольное изображение в порядке декодирования – не нижнее поле, то `prevPicOrderCntMsb` устанавливают равным 0, а `prevPicOrderCntLsb` устанавливают равным значению `TopFieldOrderCnt` для предыдущего контрольного изображения в порядке декодирования;
 - иначе (предыдущее контрольное изображение в порядке декодирования – нижнее поле) `prevPicOrderCntMsb` устанавливают равным 0, и `prevPicOrderCntLsb` устанавливают равным 0.
- Иначе (предыдущее контрольное изображение в порядке декодирования не включало `memory_management_control_operation`, равное 5) `prevPicOrderCntMsb` устанавливают равным `PicOrderCntMsb` предыдущего контрольного изображения в порядке декодирования, а `prevPicOrderCntLsb` устанавливают равным значению `pic_order_cnt_lsb` предыдущего контрольного изображения в порядке декодирования.

`PicOrderCntMsb` текущего изображения находят следующим образом:

```

if( ( pic_order_cnt_lsb < prevPicOrderCntLsb ) &&
    ( ( prevPicOrderCntLsb - pic_order_cnt_lsb ) >= ( MaxPicOrderCntLsb / 2 ) ) )
  PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb
else if( ( pic_order_cnt_lsb > prevPicOrderCntLsb ) &&
         ( ( pic_order_cnt_lsb - prevPicOrderCntLsb ) > ( MaxPicOrderCntLsb / 2 ) ) )
  PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb
else
  PicOrderCntMsb = prevPicOrderCntMsb.

```

(8-3)

Если текущее изображение – это не нижнее поле, то `TopFieldOrderCnt` находят следующим образом:

```

if( !field_pic_flag || !bottom_field_flag )
  TopFieldOrderCnt = PicOrderCntMsb + pic_order_cnt_lsb.

```

8-4)

Если текущее изображение – не верхнее поле, то `BottomFieldOrderCnt` находят следующим образом:

```

if( !field_pic_flag )
  BottomFieldOrderCnt = TopFieldOrderCnt + delta_pic_order_cnt_bottom
else if( bottom_field_flag )
  BottomFieldOrderCnt = PicOrderCntMsb + pic_order_cnt_lsb.

```

(8-5)

8.2.1.2 Процесс декодирования для вычисления порядка изображений типа 1

Этот процесс запускают, если `pic_order_cnt_type` равно 1.

Входом в этот процесс является значение `FrameNumOffset` предыдущего контрольного изображения в порядке декодирования, как определено в этом пункте.

Выходами этого процесса является одно или оба значения `TopFieldOrderCnt` или `BottomFieldOrderCnt`.

Значения `TopFieldOrderCnt` и `BottomFieldOrderCnt` находят, как определено в этом пункте. Положим что `prevFrameNum` равно `frame_num` предыдущего изображения в порядке декодирования.

Если текущее изображение – не IDR изображение, то переменную `prevFrameNumOffset` находят следующим образом:

- если предыдущее изображение в порядке декодирования включало `memory_management_control_operation`, равное 5, то `prevFrameNumOffset` устанавливают равным 0;
- иначе (предыдущее изображение в порядке декодирования не включало `memory_management_control_operation`, равное 5) `prevFrameNumOffset` устанавливают равным значению `FrameNumOffset` предыдущего изображения в порядке декодирования.

ПРИМЕЧАНИЕ. – Если `gaps_in_frame_num_value_allowed_flag` равно 1, то предыдущее изображение в порядке декодирования может быть "несуществующим" кадром, найденным процессом декодирования для промежутков в `frame_num` и определенным в п. 8.2.5.2.

Отысканию предшествуют следующие обязательные шаги.

1. Переменную `FrameNumOffset` находят следующим образом:

```

if( nal_unit_type == 5 )
  FrameNumOffset = 0
else if( prevFrameNum > frame_num )
  FrameNumOffset = prevFrameNumOffset + MaxFrameNum

```

(8-6)

```

else
    FrameNumOffset = prevFrameNumOffset.

```

2. Переменную `absFrameNum` находят следующим образом:

```

if( num_ref_frames_in_pic_order_cnt_cycle != 0 )
    absFrameNum = FrameNumOffset + frame_num
else
    absFrameNum = 0
if( nal_ref_idc == 0 && absFrameNum > 0 )
    absFrameNum = absFrameNum - 1.

```

(8-7)

3. Если `absFrameNum > 0`, `picOrderCntCycleCnt` и `frameNumInPicOrderCntCycle` находят следующим образом:

```

if( absFrameNum > 0 ) {
    picOrderCntCycleCnt = ( absFrameNum - 1 ) / num_ref_frames_in_pic_order_cnt_cycle
    frameNumInPicOrderCntCycle = ( absFrameNum - 1 ) % num_ref_frames_in_pic_order_cnt_cycle
}.

```

(8-8)

4. Переменную `expectedDeltaPerPicOrderCntCycle` находят следующим образом:

```

expectedDeltaPerPicOrderCntCycle = 0
for( i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++ )
    expectedDeltaPerPicOrderCntCycle += offset_for_ref_frame[ i ].

```

(8-9)

5. Переменную `expectedPicOrderCnt` находят следующим образом:

```

if( absFrameNum > 0 ){
    expectedPicOrderCnt = picOrderCntCycleCnt * expectedDeltaPerPicOrderCntCycle
    for( i = 0; i <= frameNumInPicOrderCntCycle; i++ )
        expectedPicOrderCnt = expectedPicOrderCnt + offset_for_ref_frame[ i ]
} else
    expectedPicOrderCnt = 0
if( nal_ref_idc == 0 )
    expectedPicOrderCnt = expectedPicOrderCnt + offset_for_non_ref_pic.

```

(8-10)

6. Переменные `TopFieldOrderCnt` или `BottomFieldOrderCnt` находят следующим образом:

```

if( !field_pic_flag ) {
    TopFieldOrderCnt = expectedPicOrderCnt + delta_pic_order_cnt[ 0 ]
    BottomFieldOrderCnt = TopFieldOrderCnt +
        offset_for_top_to_bottom_field + delta_pic_order_cnt[ 1 ]
} else if( !bottom_field_flag )
    TopFieldOrderCnt = expectedPicOrderCnt + delta_pic_order_cnt[ 0 ]
else
    BottomFieldOrderCnt = expectedPicOrderCnt + offset_for_top_to_bottom_field + delta_pic_order_cnt[ 0 ].

```

(8-11)

8.2.1.3 Процесс декодирования для вычисления порядка изображений типа 2

Этот процесс запускают, если `pic_order_type` равно 2.

Выходами этого процесса является одно или оба значения `TopFieldOrderCnt` или `BottomFieldOrderCnt`.

Положим `prevFrameNum` равным `frame_num` из предыдущего изображения в порядке декодирования.

Если текущее изображение – не IDR изображение, то переменную `prevFrameNumOffset` находят следующим образом:

- если предыдущее изображение в порядке декодирования включало `memory_management_control_operation`, равное 5, то `prevFrameNumOffset` устанавливают равным 0;
- иначе (предыдущее изображение в порядке декодирования не включало `memory_management_control_operation`, равное 5) `prevFrameNumOffset` устанавливают равным значению `FrameNumOffset` из предыдущего изображения в порядке декодирования.

ПРИМЕЧАНИЕ 1. – Если `gaps_in_frame_num_value_allowed_flag` равно 1, то предыдущее изображение в порядке декодирования может быть "несуществующим" кадром, найденным процессом декодирования для промежутков в `frame_num`, определенным в п. 8.2.5.2.

Переменную `FrameNumOffset` находят следующим образом:

```

if( nal_unit_type == 5 )
    FrameNumOffset = 0
else if( prevFrameNum > frame_num )
    FrameNumOffset = prevFrameNumOffset + MaxFrameNum
else
    FrameNumOffset = prevFrameNumOffset.
    
```

(8-12)

Переменную `tempPicOrderCnt` находят следующим образом:

```

if( nal_unit_type == 5 )
    tempPicOrderCnt = 0
else if( nal_ref_idc == 0 )
    else
        tempPicOrderCnt = 2 * ( FrameNumOffset + frame_num ).
    
```

(8-13) tempPi

Переменные `TopFieldOrderCnt` или `BottomFieldOrderCnt` находят следующим образом:

```

if( !field_pic_flag ) {
    TopFieldOrderCnt = tempPicOrderCnt
    BottomFieldOrderCnt = tempPicOrderCnt
} else if( bottom_field_flag )
    BottomFieldOrderCnt = tempPicOrderCnt
else
    TopFieldOrderCnt = tempPicOrderCnt.
    
```

(8-14)

ПРИМЕЧАНИЕ 2. – Вычисление порядка изображений типа 2 нельзя использовать в кодированной видеопоследовательности, содержащей непрерывно следующие неконтрольные изображения, которые могли появиться более чем в одном из этих изображений, имеющих то же значение `TopFieldOrderCnt`, или более чем в одном из этих изображений, имеющих то же значение `BottomFieldOrderCnt`.

ПРИМЕЧАНИЕ 3. – На выходе результатами вычисления порядка изображений типа 2 является порядок такой же, как порядок декодирования.

8.2.2 Процесс декодирования для отображения макроблока в группу секции

Входами в этот процесс являются набор параметров действующего изображения и заголовок той секции, из которой должна быть декодирована заданная секция.

Выходом этого процесса является отображение макроблока в группу секции `MbToSliceGroupMap`.

Этот процесс активируют в начале каждой секции.

ПРИМЕЧАНИЕ. – Выход этого процесса равен всем секциям изображения.

Если `num_slice_groups_minus1` равно 1 и `slice_group_map_type` равно 3, 4 или 5, то группы секций 0 и 1 имеют форму и размеры, которые определяются значением `slice_group_change_direction_flag`, как показано в таблице 8-1 и описано в пп. 8.2.2.4–8.2.2.6.

Таблица 8-1 – Детальный тип отображения группы секции

<code>slice_group_map_type</code>	<code>slice_group_change_direction_flag</code>	Детальный тип отображения группы секции
3	0	Вне блока по часовой стрелке
3	1	Вне блока против часовой стрелки
4	0	Растровое сканирование
4	1	Обратное растровое сканирование
5	0	Вытеснение направо
5	1	Вытеснение налево

В таком случае блоки отображения группы секции MapUnitsInSliceGroup0 в определенном возрастающем порядке располагают в группе секции 0, а оставшиеся блоки отображения группы секции изображения PicSizeInMapUnits – MapUnitsInSliceGroup0 располагают в группе секции 1.

Если num_slice_groups_minus1 равно 1, а slice_group_map_type равно 4 или 5, переменную sizeOfUpperLeftGroup определяют следующим образом:

$$\text{sizeOfUpperLeftGroup} = (\text{slice_group_change_direction_flag} ? (\text{PicSizeInMapUnits} - \text{MapUnitsInSliceGroup0}) : \text{MapUnitsInSliceGroup0}). \quad (8-15)$$

Переменную mapUnitToSliceGroupMap находят следующим образом.

- Если num_slice_groups_minus1 равно 0, отображение блока в отображенную группу секции производят для всех i в диапазоне от 0 до PicSizeInMapUnits – 1 включительно, как определено уравнением:

$$\text{mapUnitToSliceGroupMap}[i] = 0. \quad (8-16)$$

- Иначе (num_slice_groups_minus1 не равно 0) mapUnitToSliceGroupMap находят следующим образом.
 - Если slice_group_map_type равно 0, mapUnitToSliceGroupMap находят так, как это определено в п. 8.2.2.1.
 - Иначе, если slice_group_map_type равно 1, mapUnitToSliceGroupMap находят так, как это определено в п. 8.2.2.2.
 - Иначе, если slice_group_map_type равно 2, mapUnitToSliceGroupMap находят так, как это определено в п. 8.2.2.3.
 - Иначе, если slice_group_map_type равно 3, mapUnitToSliceGroupMap находят так, как это определено в п. 8.2.2.4.
 - Иначе, если slice_group_map_type равно 4, mapUnitToSliceGroupMap находят так, как это определено в п. 8.2.2.5.
 - Иначе, если slice_group_map_type равно 5, mapUnitToSliceGroupMap находят так, как это определено в п. 8.2.2.6.
 - Иначе (slice_group_map_type равно 6) mapUnitToSliceGroupMap находят так, как это определено в п. 8.2.2.7.

После вывода mapUnitToSliceGroupMap активируют процесс, определенный в п. 8.2.2.8 для преобразования отображенного блока в отображенную группу секции mapUnitToSliceGroupMap, а макроблока – в отображенную группу секции MbToSliceGroupMap. После того как отображение макроблока в группу секции найдено, как это определено в п. 8.2.2.8, функцию NextMbAddress(n) определяют как значение переменной nextMbAddress по алгоритму:

$$\begin{aligned} & i = n + 1 \\ & \text{while}(i < \text{PicSizeInMbs} \ \&\& \ \text{MbToSliceGroupMap}[i] \neq \text{MbToSliceGroupMap}[n]) \\ & \quad i++; \\ & \text{nextMbAddress} = i. \end{aligned} \quad (8-17)$$

8.2.2.1 Спецификация типа чередующейся группы секции

Спецификацию этого пункта применяют, если slice_group_map_type равно 0.

Отображенный блок в отображенной группе секции формируют, как определено ниже:

$$\begin{aligned} & i = 0 \\ & \text{do} \\ & \quad \text{for}(iGroup = 0; iGroup \leq \text{num_slice_groups_minus1} \ \&\& \ i < \text{PicSizeInMapUnits}; \\ & \quad \quad i += \text{run_length_minus1}[iGroup++] + 1) \\ & \quad \text{for}(j = 0; j \leq \text{run_length_minus1}[iGroup] \ \&\& \ i + j < \text{PicSizeInMapUnits}; j++) \\ & \quad \quad \text{mapUnitToSliceGroupMap}[i + j] = iGroup \\ & \text{while}(i < \text{PicSizeInMapUnits}). \end{aligned} \quad (8-18)$$

8.2.2.2 Спецификация типа отображения распределенной группы секции

Спецификацию этого пункта применяют, если slice_group_map_type равно 1.

Отображенный блок в отображенной группе секции формируют, как определено ниже:

```
for( i = 0; i < PicSizeInMapUnits; i++)
    mapUnitToSliceGroupMap[ i ] = ( ( i % PicWidthInMbs ) +
        ( ( i / PicWidthInMbs ) * ( num_slice_groups_minus1 + 1 ) ) / 2 )
        % ( num_slice_groups_minus1 + 1 ).
```

(8-19)

8.2.2.3 Спецификация переднего плана с типом отображения сдвига группы секции налево вверх

Спецификацию этого пункта применяют, если slice_group_map_type равно 2.

Отображенный блок в отображенной группе секции формируют, как определено ниже:

```
for( i = 0; i < PicSizeInMapUnits; i++)
    mapUnitToSliceGroupMap[ i ] = num_slice_groups_minus1
for( iGroup = num_slice_groups_minus1 - 1; iGroup >= 0; iGroup-- ) {
    yTopLeft = top_left[ iGroup ] / PicWidthInMbs
    xTopLeft = top_left[ iGroup ] % PicWidthInMbs
    yBottomRight = bottom_right[ iGroup ] / PicWidthInMbs
    xBottomRight = bottom_right[ iGroup ] % PicWidthInMbs
    for( y = yTopLeft; y <= yBottomRight; y++)
        for( x = xTopLeft; x <= xBottomRight; x++)
            mapUnitToSliceGroupMap[ y * PicWidthInMbs + x ] = iGroup
}
```

(8-20)

ПРИМЕЧАНИЕ. – Прямоугольники могут перекрываться. Группа секции 0 содержит макроблоки, которые находятся внутри прямоугольника, определенного верхней левой [0] и нижней правой [0] координатой. Группа секции, в которой ID группы секции больше 0 и меньше, чем num_slice_groups_minus1, содержит макроблоки, которые находятся внутри определенного прямоугольника для этой группы секции, и не находятся внутри прямоугольника, определенного для любой группы секции, имеющей меньшее значение ID группы секции. Группа секции с ID группы секции, равным num_slice_groups_minus1, содержит макроблоки, которых нет в других группах секций.

8.2.2.4 Спецификация типов отображения групп секций вне блока

Спецификацию этого пункта применяют, если slice_group_map_type равно 3.

Отображенный блок в отображенной группе секции формируют, как определено ниже:

```
for( i = 0; i < PicSizeInMapUnits; i++)
    mapUnitToSliceGroupMap[ i ] = 1
x = ( PicWidthInMbs - slice_group_change_direction_flag ) / 2
y = ( PicHeightInMapUnits - slice_group_change_direction_flag ) / 2
( leftBound, topBound ) = ( x, y )
( rightBound, bottomBound ) = ( x, y )
( xDir, yDir ) = ( slice_group_change_direction_flag - 1, slice_group_change_direction_flag )
for( k = 0; k < MapUnitsInSliceGroup0; k += mapUnitVacant ) {
    mapUnitVacant = ( mapUnitToSliceGroupMap[ y * PicWidthInMbs + x ] == 1 )
    if( mapUnitVacant )
        mapUnitToSliceGroupMap[ y * PicWidthInMbs + x ] = 0
    if( xDir == -1 && x == leftBound ) {
        leftBound = Max( leftBound - 1, 0 )
        x = leftBound
        ( xDir, yDir ) = ( 0, 2 * slice_group_change_direction_flag - 1 )
    } else if( xDir == 1 && x == rightBound ) {
        rightBound = Min( rightBound + 1, PicWidthInMbs - 1 )
        x = rightBound
        ( xDir, yDir ) = ( 0, 1 - 2 * slice_group_change_direction_flag )
    } else if( yDir == -1 && y == topBound ) {
        topBound = Max( topBound - 1, 0 )
        y = topBound
        ( xDir, yDir ) = ( 1 - 2 * slice_group_change_direction_flag, 0 )
    } else if( yDir == 1 && y == bottomBound ) {
        bottomBound = Min( bottomBound + 1, PicHeightInMapUnits - 1 )
        y = bottomBound
        ( xDir, yDir ) = ( 2 * slice_group_change_direction_flag - 1, 0 )
    } else
        ( x, y ) = ( x + xDir, y + yDir )
}
```

(8-21)

8.2.2.5 Спецификация типов отображения групп секций растрового сканирования

Спецификацию этого пункта применяют, если `slice_group_map_type` равно 4.

Отображенный блок в отображенной группе секции формируют, как определено ниже:

```
for( i = 0; i < PicSizeInMapUnits; i++ )
  if( i < sizeOfUpperLeftGroup )
    mapUnitToSliceGroupMap[ i ] = slice_group_change_direction_flag
  else
    mapUnitToSliceGroupMap[ i ] = 1 - slice_group_change_direction_flag. (8-22)
```

8.2.2.6 Спецификация типов отображения вытесненных групп секций

Спецификацию этого пункта применяют, если `slice_group_map_type` равно 5.

Отображенный блок в отображенной группе секции формируют, как определено ниже:

```
k = 0;
for( j = 0; j < PicWidthInMbs; j++ )
  for( i = 0; i < PicHeightInMapUnits; i++ )
    if( k++ < sizeOfUpperLeftGroup )
      mapUnitToSliceGroupMap[ i * PicWidthInMbs + j ] = slice_group_change_direction_flag
    else
      mapUnitToSliceGroupMap[ i * PicWidthInMbs + j ] = 1 - slice_group_change_direction_flag. (8-23)
```

8.2.2.7 Спецификация типа отображения группы секции, заданной в явной форме

Спецификацию этого пункта применяют, если `slice_group_map_type` равно 6.

Отображенный блок в отображенной группе секции формируют, как определено ниже:

```
mapUnitToSliceGroupMap[ i ] = slice_group_id[ i ]. (8-24)
```

Для всех i в диапазоне от 0 до `PicSizeInMapUnits - 1` включительно.

8.2.2.8 Спецификация для преобразования отображенного блока в отображенную группу секции и макроблока в отображенную группу секции

Для каждого значения i в диапазоне от 0 до `PicSizeInMbs - 1` включительно отображение макроблока в группу секции определено следующим образом.

- Если `frame_mbs_only_flag` равно 1 или `field_pic_flag` равно 1, отображение макроблока в группу секции определено как:

```
MbToSliceGroupMap[ i ] = mapUnitToSliceGroupMap[ i ]. (8-25)
```

- Иначе, если `MbaffFrameFlag` равно 1, отображение макроблока в группу секции определено как:

```
MbToSliceGroupMap[ i ] = mapUnitToSliceGroupMap[ i / 2 ]. (8-26)
```

- Иначе (`frame_mbs_only_flag` равно 0, `mb_adaptive_frame_field_flag` равно 0 и `field_pic_flag` равно 0) отображение макроблока в группу секции определено как:

```
MbToSliceGroupMap[ i ] = mapUnitToSliceGroupMap[ ( i / ( 2 * PicWidthInMbs ) ) * PicWidthInMbs + ( i % PicWidthInMbs ) ]. (8-27)
```

8.2.3 Процесс декодирования для разделения данных секции

Входами в этот процесс являются:

- RBSP слой разделенной части А данных секции;
- RBSP слой разделенной части В данных секции, имеющей тот же идентификатор `slice_id`, что и в RBSP слое разделенной части А данных секции, если в данных секции присутствуют элементы синтаксиса категории 3; и

- RBSP слой разделенной части С данных секции, имеющей тот же идентификатор slice_id, что и в RBSP слое разделенной части А данных секции, если в данных секции присутствуют элементы синтаксиса категории 4.

ПРИМЕЧАНИЕ 1. – Не обязательно должны быть представлены RBSP слой разделенной части В данных секции и RBSP слой разделенной части С данных секции.

Выход этого процесса – кодированная секция.

Если не используют разделение данных секции, то кодированные секции представляют слоем секции без RBSP разделения, который содержит заголовок секции и далее следует структура синтаксиса данных секции, содержащая все элементы синтаксиса категорий 2, 3 и 4 (см. столбец категорий в п. 7.3) данных макроблока для макроблоков этой секции.

Если используют разделение данных секции, то данные макроблока секции разделяют на части, от одной до трех, каждая в отдельном блоке NAL. Разделенная часть А содержит заголовок части А данных секции и все элементы синтаксиса категории 2. Разделенная часть В, если она присутствует, содержит заголовок части В данных секции и все элементы синтаксиса категории 3. Разделенная часть С, если она присутствует, содержит заголовок части С данных секции и все элементы синтаксиса категории 4.

При использовании разделения данных секции элементы синтаксиса каждой категории анализируют из отдельного блока NAL, который может не присутствовать, если отсутствуют символы соответствующей категории. В кодированной секции процесс декодирования должен обрабатывать разделенные части данных секции способом, эквивалентным обработке соответствующего слоя секция без разделения RBSP, с помощью извлечения каждого элемента синтаксиса из разделенной части данных секции, в которой элемент синтаксиса появляется в зависимости от назначения разделенной части в таблицах синтаксиса в п. 7.3.

ПРИМЕЧАНИЕ 2. – Элементы синтаксиса категории 3 существенны для декодирования остатков данных макроблоков типов I и SI. Элементы синтаксиса категории 4 существенны для декодирования остатков данных макроблоков типов P и B. Категория 2 включает все прочие элементы синтаксиса, связанные с декодированием макроблоков, и информацию о них часто обозначают как информационный заголовок. Заголовок части А данных секции содержит все элементы синтаксиса заголовка секции и дополнительно идентификатор slice_id секции, который используют для объединения разделенных частей данных секции В и С с разделенной частью А. Заголовки разделенных частей данных секции В и С содержат элемент синтаксиса slice_id, который устанавливает их объединение с разделенной частью А данных секции.

8.2.4 Процесс декодирования для создания списков контрольных изображений

Этот процесс активируют в начале декодирования каждой секции P, SP или B.

Декодированные контрольные изображения помечены как "используется для краткосрочного контроля" или "используется для долгосрочного контроля", как определено потоком битов в п. 8.2.5. Краткосрочные контрольные изображения идентифицированы значением frame_num. Долгосрочным контрольным изображениям присвоен долгосрочный индекс кадра, как это определено потоком битов и в п. 8.2.5.

Пункт 8.2.4.1 активируют, чтобы определить:

- присвоение переменным FrameNum, FrameNumWrap и PicNum каждого из краткосрочных контрольных изображений; и
- присвоение переменной LongTermPicNum каждого из долгосрочных контрольных изображений.

Контрольные изображения адресуют по их индексам контроля, как определено в п. 8.4.2.1. Индекс контроля – это индекс в списке контрольных изображений. Если декодирована секция P или SP, существует единственный список контрольных изображений RefPicList0. Если декодирована секция B, существует второй независимый список контрольных изображений RefPicList1 дополнительно к RefPicList0.

В начале декодирования каждой секции список контрольных изображений RefPicList0, а для секций B RefPicList1, находят следующим образом.

- Первоначальный список контрольных изображений RefPicList0, а для секций B RefPicList1, находят, как определено в п. 8.2.4.2.
- Первоначальный список контрольных изображений RefPicList0, а для секций B RefPicList1, модифицируют, как определено в п. 8.2.4.3.

ПРИМЕЧАНИЕ. – Процесс изменения порядка в списках контрольных изображений, описанный в п. 8.2.4.3, позволяет гибко изменять содержание RefPicList0 и RefPicList1 для секций B. В частности, возможно вставить изображение, помеченное как "используется для контроля" в RefPicList0 и в RefPicList1 для секций B, даже если это изображение не находится в исходном списке контрольных изображений, описание которого содержится в п. 8.2.4.2.

Число входов в модифицированном списке контрольных изображений RefPicList0 – это num_ref_idx_l0_active_minus1 + 1, а для секций B число входов в модифицированном списке контрольных изображений RefPicList1 – это num_ref_idx_l1_active_minus1 + 1. В модифицированных списках контрольных изображений RefPicList0 или RefPicList1 может появиться более одного индекса.

8.2.4.1 Процесс декодирования номеров изображений

Этот процесс активируют, если активируют процесс декодирования для создания списков контрольных изображений, определенный в п. 8.2.4, или процесс разметки декодированного контрольного изображения, определенный в п. 8.2.5.

Переменные FrameNum, FrameNumWrap, PicNum, LongTermFrameIdx и LongTermPicNum используют для процесса инициализации списков контрольных изображений (согласно п. 8.2.4.2), модификации процесса списков контрольных изображений (согласно п. 8.2.4.3) и для процесса разметки декодированного контрольного изображения (согласно п. 8.2.5).

Каждому краткосрочному контрольному изображению переменные FrameNum и FrameNumWrap присваивают следующим образом. Во-первых, FrameNum устанавливают равным элементу синтаксиса frame_num, который декодирован в заголовке(ax) секции соответствующего краткосрочного контрольного изображения. Затем переменную FrameNumWrap находят как:

```
if( FrameNum > frame_num )
    FrameNumWrap = FrameNum - MaxFrameNum
else
    FrameNumWrap = FrameNum,
```

(8-28)

где значение frame_num, использованное в уравнении 8-28, – это frame_num в заголовке(ax) секции текущего изображения.

Каждое долгосрочное контрольное изображение имеет связанное значение LongTermFrameIdx (которое было ему присвоено, как это определено в п. 8.2.5).

Каждому краткосрочному контрольному изображению присваивают переменную PicNum, а каждому долгосрочному контрольному изображению присваивают переменную LongTermPicNum. Значения этих переменных зависят от значений field_pic_flag и bottom_field_flag текущего изображения, и их устанавливают следующим образом.

- Если field_pic_flag равно 0, используют следующие условия.
 - Для каждого краткосрочного контрольного кадра или пары дополнительных контрольных полей:

$$\text{PicNum} = \text{FrameNumWrap}.$$
(8-29)

- Для каждого долгосрочного контрольного кадра или долгосрочной пары дополнительных контрольных полей:

$$\text{LongTermPicNum} = \text{LongTermFrameIdx}.$$
(8-30)

ПРИМЕЧАНИЕ. – При декодировании кадра значение MbaffFrameFlag не оказывает влияния на выводы формул в пп. 8.2.4.2, 8.2.4.3 и 8.2.5.

- Иначе (field_pic_flag равно 1) используют следующие условия.
 - Для каждого краткосрочного контрольного поля используют следующие условия.
 - Если контрольное поле имеет ту же четность, что и текущее поле:

$$\text{PicNum} = 2 * \text{FrameNumWrap} + 1.$$
(8-31)

- Иначе (контрольное поле имеет обратную четность текущего поля):

$$\text{PicNum} = 2 * \text{FrameNumWrap}.$$
(8-32)

- Для каждого долгосрочного контрольного поля используют следующие условия.
 - Если контрольное поле имеет ту же четность, что и текущее поле:

$$\text{LongTermPicNum} = 2 * \text{LongTermFrameIdx} + 1.$$
(8-33)

- Иначе (контрольное поле имеет обратную четность текущего поля):

$$\text{LongTermPicNum} = 2 * \text{LongTermFrameIdx}.$$
(8-34)

8.2.4.2 Процесс инициализации списков контрольных изображений

Процесс инициализации активируют, если декодирован заголовок секции P, SP или B.

RefPicList0 и RefPicList1 имеют начальные входы, как определено в пп. 8.2.4.2.1–8.2.4.2.5.

Если число входов в исходных значениях RefPicList0 или RefPicList1, созданных, как это определено в пп. 8.2.4.2.1–8.2.4.2.5, будет больше чем $\text{num_ref_idx_l0_active_minus1} + 1$ или $\text{num_ref_idx_l1_active_minus1} + 1$, соответственно, то позиции предыдущих дополнительных входов $\text{num_ref_idx_l0_active_minus1}$ или $\text{num_ref_idx_l1_active_minus1}$ аннулируют в исходном списке контрольных изображений.

Если число входов в исходных значениях RefPicList0 или RefPicList1, созданных, как это определено в пп. 8.2.4.2.1–8.2.4.2.5, будет меньше чем $\text{num_ref_idx_l0_active_minus1} + 1$ или $\text{num_ref_idx_l1_active_minus1} + 1$, соответственно, то оставшиеся входы в исходном списке контрольных изображений устанавливают равными "нет контрольного изображения".

8.2.4.2.1 Процесс инициализации списка контрольных изображений в кадрах секций P и SP

Процесс инициализации активируют при декодировании секции P или SP в кодированном кадре.

Если этот процесс активирован, то должен быть, по крайней мере, один контрольный кадр или пара дополнительных контрольных полей, которые помечены как "используется для краткосрочного контроля" или "используется для долгосрочного контроля".

Список контрольных изображений RefPicList0 составлен так, что краткосрочные контрольные кадры и краткосрочная пара дополнительных контрольных полей имеют меньшие индексы, чем долгосрочные контрольные кадры и долгосрочная пара дополнительных контрольных полей.

Краткосрочные контрольные кадры и пара дополнительных контрольных полей расположены так, что начинают с кадра или пары дополнительных полей с наивысшим значением PicNum и далее следуют в нисходящем порядке до кадра или пары дополнительных полей с самым низким значением PicNum.

Долгосрочные контрольные кадры и пара дополнительных контрольных полей расположены так, что начинают с кадра или пары дополнительных полей с самым низким значением LongTermPicNum и далее следуют в возрастающем порядке до кадра или пары дополнительных полей с наивысшим значением LongTermPicNum.

ПРИМЕЧАНИЕ. – Непарное контрольное поле не используют для внешнего предсказания при декодировании кадра, независимо от значения MbaffFrameFlag.

Например, если три контрольных кадра помечены как "используется для краткосрочного контроля" с PicNum, равным 300, 302 и 303, а два контрольных кадра помечены как "используется для долгосрочного контроля" с LongTermPicNum, равным 0 и 3, то исходный порядок индексов будет следующим:

- RefPicList0[0] устанавливают равным краткосрочному контрольному изображению с PicNum = 303;
- RefPicList0[1] устанавливают равным краткосрочному контрольному изображению с PicNum = 302;
- RefPicList0[2] устанавливают равным краткосрочному контрольному изображению с PicNum = 300;
- RefPicList0[3] устанавливают равным долгосрочному контрольному изображению с LongTermPicNum = 0; и
- RefPicList0[4] устанавливают равным долгосрочному контрольному изображению с LongTermPicNum = 3.

8.2.4.2.2 Процесс инициализации списка контрольных изображений в полях секций P и SP

Процесс инициализации активируют при декодировании секции P или SP в кодированном поле.

Каждое поле, включенное в список контрольных изображений RefPicList0, имеет отдельный индекс в списке контрольных изображений RefPicList0.

При декодировании поля каждое поле, включенное в список контрольных изображений, имеет в списке отличный индекс.

ПРИМЕЧАНИЕ. – При декодировании поля имеется, по крайней мере, вдвое больше действующих изображений, доступных для контроля точно так же, как при декодировании кадра в том же положении порядка декодирования.

Два упорядоченных списка контрольных кадров, refFrameList0ShortTerm и refFrameList0LongTerm, находят следующим образом. С целью формирования этого списка кадры, декодированные контрольные кадры, пары дополнительных контрольных полей, непарные контрольные поля и контрольные кадры, в которых единственное поле отмечено как "используется для краткосрочного контроля" или "используется для долгосрочного контроля", – все рассматривают как контрольные кадры.

- Все кадры, имеющие одно или более полей, помеченные как "используется для краткосрочного контроля", включают в список краткосрочных контрольных кадров refFrameList0ShortTerm. Если текущее поле – второе поле (в порядке декодирования) пары дополнительных контрольных полей, а первое поле помечено как "используется для краткосрочного контроля", то первое поле включают в список краткосрочных контрольных кадров refFrameList0ShortTerm. В списке refFrameList0ShortTerm порядок расположения начинается с контрольного кадра с наивысшим значением FrameNumWrap и далее следует в нисходящем порядке до контрольного кадра с наименьшим значением FrameNumWrap.
- Все кадры, имеющие одно или более полей, помеченные как "используется для долгосрочного контроля", включают в список долгосрочных контрольных кадров refFrameList0LongTerm. Если текущее поле – второе поле (в порядке декодирования) пары дополнительных контрольных полей, а первое поле помечено как "используется для долгосрочного контроля", то первое поле включают в список долгосрочных контрольных кадров refFrameList0LongTerm. В списке refFrameList0LongTerm порядок расположения начинается с контрольного

кадра с наименьшим значением LongTermFrameIdx и далее следует в возрастающем порядке до контрольного кадра с наибольшим значением LongTermFrameIdx.

Процесс, определенный в п. 8.2.4.2.5, активируют со значениями refFrameList0ShortTerm и refFrameList0LongTerm, заданными как входные, а выходные присвоены значению RefPicList0.

8.2.4.2.3 Процесс инициализации списка контрольных изображений в кадрах секций В

Процесс инициализации активируют при декодировании секции В в кодированном кадре.

Для целей формирования списков контрольных изображений RefPicList0 и RefPicList1 термин "контрольный вход" используется далее для обозначения декодированных контрольных кадров или пар дополнительных контрольных полей.

Если этот процесс активирован, тогда должен быть, по крайней мере, один контрольный вход, который помечен как "используется для краткосрочного контроля" или "используется для долгосрочного контроля".

Для секций В порядок краткосрочных контрольных входов в списках контрольных изображений RefPicList0 и RefPicList1 зависит от порядка на выходе, который задан PicOrderCnt(). Если pic_order_cnt_type равно 0, контрольные изображения, которые помечены как "несуществующие", как это определено в п. 8.2.5.2, не включают ни в RefPicList0, ни в RefPicList1.

ПРИМЕЧАНИЕ 1. – Если gaps_in_frame_num_value_allowed_flag равно 1, кодеры должны изменить порядок в списке контрольных изображений, чтобы обеспечить правильное выполнение процесса декодирования (в частности, если pic_order_cnt_type равно 0. В этом случае значение PicOrderCnt() не считают "несуществующими" кадрами).

Список контрольных изображений RefPicList0 составлен так, что краткосрочные контрольные входы имеют меньшие индексы, чем долгосрочные контрольные входы. Список составляется следующим образом.

- Пусть entryShortTerm – это переменная, диапазон значений которой образуют все контрольные входы, помеченные как "используется для краткосрочного контроля". Если представлены несколько значений entryShortTerm, у которых PicOrderCnt(entryShortTerm) меньше PicOrderCnt(CurrPic), эти значения переменной entryShortTerm помещаются в начале refPicList0 в убывающем порядке PicOrderCnt(entryShortTerm). Все остальные значения entryShortTerm (если представлены) затем добавляются к refPicList0 в возрастающем порядке PicOrderCnt(entryShortTerm).
- Долгосрочные контрольные входы начинаются с долгосрочного контрольного входа, который имеет наименьшее значение LongTermPicNum. Эта последовательность продолжается в возрастающем порядке до долгосрочного контрольного входа с наивысшим значением LongTermPicNum.

Список контрольных изображений RefPicList1 составлен так, что краткосрочные контрольные входы имеют меньшие индексы, чем долгосрочные контрольные входы. Список составляется следующим образом.

- Пусть entryShortTerm – переменная, диапазон значений которой образуют все контрольные входы, помеченные как "используется для краткосрочного контроля". Если представлены несколько значений entryShortTerm, у которых PicOrderCnt(entryShortTerm) больше PicOrderCnt(CurrPic), эти значения переменной entryShortTerm помещаются в начале refPicList1 в возрастающем порядке PicOrderCnt(entryShortTerm). Все остальные значения entryShortTerm (если представлены) затем добавляются к refPicList1 в убывающем порядке PicOrderCnt(entryShortTerm).
 - Долгосрочные контрольные входы начинаются с долгосрочного контрольного кадра или пары дополнительных контрольных полей, которые имеют наименьшее значение LongTermPicNum, и продолжают в возрастающем порядке до долгосрочного контрольного входа с наибольшим значением LongTermPicNum.
 - Если список контрольных изображений RefPicList1 имеет более одного входа, а RefPicList1 идентичен списку контрольных изображений RefPicList0, то первые два входа RefPicList1[0] и RefPicList1[1] переключают.
- ПРИМЕЧАНИЕ 2. – Непарное контрольное поле не используют для внешнего предсказания кадров, независимо от значения MbaffFrameFlag.

8.2.4.2.4 Процесс инициализации списка контрольных изображений в полях секций В

Процесс инициализации активируют при декодировании в кодированном поле секции В.

При декодировании поля каждое поле сохраненного контрольного кадра идентифицируют как отдельное контрольное изображение с особым индексом. Порядок краткосрочных контрольных изображений в списках контрольных изображений RefPicList0 и RefPicList1 зависит от порядка на выходе, заданного PicOrderCnt(). Если pic_order_cnt_type равно 0, то контрольные изображения, помеченные как "несуществующие", как это определено в п. 8.2.5.2, не включают ни в RefPicList0, ни в RefPicList1.

ПРИМЕЧАНИЕ 1. – Если gaps_in_frame_num_value_allowed_flag равно 1, кодеры должны изменить порядок в списке контрольных изображений, чтобы обеспечить правильное выполнение процесса декодирования (в частности, если pic_order_cnt_type равно 0. В этом случае значения PicOrderCnt() не считают "несуществующими" кадрами).

ПРИМЕЧАНИЕ 2. – При декодировании поля имеется, по крайней мере, вдвое больше действующих изображений, доступных для контроля точно так же, как при декодировании кадра в том же положении порядка декодирования.

Три упорядоченных списка контрольных кадров, `refFrameList0ShortTerm`, `refFrameList1ShortTerm` и `refFrameListLongTerm`, находят следующим образом. С целью формирования этого списка кадров термин контрольный вход относят к декодированным контрольным кадрам, парам дополнительных контрольных полей или к непарным контрольным полям. Если `pic_order_cnt_type` равно 0, термин контрольный вход не относят к кадрам, которые помечены как "несуществующие", как это определено в п. 8.2.5.2.

- Пусть `entryShortTerm` – это переменная, диапазон значений которой образуют все контрольные входы, помеченные как "используется для краткосрочного контроля". Если представлены несколько значений `entryShortTerm`, у которых `PicOrderCnt(entryShortTerm)` меньше или равно `PicOrderCnt(CurrPic)`, эти значения переменной `entryShortTerm` помещаются в начале `refFrameList0ShortTerm` в убывающем порядке `PicOrderCnt(entryShortTerm)`. Все остальные значения `entryShortTerm` (если представлены) затем добавляются к `refFrameList0ShortTerm` в возрастающем порядке `PicOrderCnt(entryShortTerm)`.

ПРИМЕЧАНИЕ 3. – Если в порядке декодирования текущее поле следует за кодированным полем `fldPrev`, с которым это поле образует пару дополнительных контрольных полей, `fldPrev` включают в список `refFrameList0ShortTerm` с помощью `PicOrderCnt(fldPrev)` и используют метод упорядочения, описанный в предыдущем предложении.

- Пусть `entryShortTerm` – переменная, диапазон значений которой образуют все контрольные входы, помеченные как "используется для краткосрочного контроля". Если представлены несколько значений `entryShortTerm`, у которых `PicOrderCnt(entryShortTerm)` больше `PicOrderCnt(CurrPic)`, эти значения переменной `entryShortTerm` помещаются в начале `refFrameList1ShortTerm` в возрастающем порядке `PicOrderCnt(entryShortTerm)`. Все остальные значения `entryShortTerm` (если представлены) затем добавляются к `refFrameList1ShortTerm` в убывающем порядке `PicOrderCnt(entryShortTerm)`.

ПРИМЕЧАНИЕ 4. – Если в порядке декодирования текущее поле следует за кодированным полем `fldPrev`, с которым это поле образует пару дополнительных контрольных полей, `fldPrev` включают в список `refFrameList1ShortTerm` с помощью `PicOrderCnt(fldPrev)` и используют метод упорядочения, описанный в предыдущем предложении.

- `refFrameListLongTerm` по порядку начинают с контрольного входа, имеющего наименьшее значение `LongTermFrameIdx`, и продолжают в возрастающем порядке до контрольного входа, имеющего наибольшее значение `LongTermFrameIdx`.

ПРИМЕЧАНИЕ 5. – Если дополнительное поле текущего изображения помечено как "используется для долгосрочного контроля", это поле включают в список `refFrameListLongTerm`. Контрольный вход, в котором только одно поле помечено как "используется для долгосрочного контроля", включают в список `refFrameListLongTerm`.

Процесс, определенный в п. 8.2.4.2.5, активируют с `refFrameList0ShortTerm` и `refFrameListLongTerm`, заданными в качестве входа и выхода, которые присвоены значению `RefPicList0`.

Процесс, определенный в п. 8.2.4.2.5, активируют с `refFrameList1ShortTerm` и `refFrameListLongTerm`, заданными как вход и выход, которые присвоены значению `RefPicList1`.

Если список контрольных изображений `RefPicList1` имеет более одного входа, а `RefPicList1` идентичен списку контрольных изображений `RefPicList0`, то первые два входа `RefPicList1[0]` и `RefPicList1[1]` переключают.

8.2.4.2.5 Процесс инициализации списков контрольных изображений в полях

Входами этого процесса являются списки контрольных кадров `refFrameListXShortTerm` (с `X` равным 0 или 1) и `refFrameListLongTerm`.

Список контрольных изображений `RefPicListX` – это список в таком порядке, при котором краткосрочные контрольные поля имеют более низкие индексы, чем долгосрочные контрольные поля. Заданные списки контрольных кадров `refFrameListXShortTerm` и `refFrameListLongTerm` находят следующим образом.

- Краткосрочные контрольные поля располагают, выбирая контрольные поля из упорядоченного списка кадров `refFrameListXShortTerm` переменной полей различной четности, начиная с поля, которое имеет ту же четность, что и текущее поле (если это поле присутствует). Если какое-нибудь поле контрольного кадра не было декодировано или отмечено как "используется для краткосрочного контроля", такое исчезающее поле игнорируют и вместо него из упорядоченного списка кадров `refFrameListXShortTerm` выбирают следующее доступное сохраненное контрольное поле той же четности и вводят его в `RefPicListX`. Если в упорядоченном списке кадров `refFrameListXShortTerm` отсутствует краткосрочное контрольное поле альтернативной четности, в `RefPicListX` вводят еще не индексированные поля соответствующей четности в том порядке, в котором они расположены в упорядоченном списке кадров `refFrameListXShortTerm`.
- Долгосрочные контрольные поля упорядочивают, выбирая контрольные поля из упорядоченного списка кадров `refFrameListLongTerm` переменной полей различной четности, начиная с поля, которое имеет ту же четность, что и текущее поле (если это поле присутствует). Если какое-нибудь поле контрольного кадра не было декодировано или отмечено как "используется для долгосрочного контроля", такое исчезающее поле игнорируют и вместо него из упорядоченного списка кадров `refFrameListLongTerm` выбирают следующее доступное сохраненное контрольное поле той же четности и вводят его в `RefPicListX`. Если в упорядоченном списке кадров `refFrameListLongTerm` отсутствует долгосрочное контрольное поле альтернативной четности, в `refFrameListLongTerm` вводят еще не индексированные поля соответствующей четности в том порядке, в котором они расположены в упорядоченном списке кадров `refFrameListLongTerm`.

8.2.4.3 Процесс изменения порядка в списках контрольных изображений

Если `ref_pic_list_reordering_flag_l0` равно 1, используют следующие условия.

- Допустим, что `refIdxL0` – индекс в списке контрольных изображений `RefPicList0`. Первоначально он установлен равным 0.

- Соответствующие элементы синтаксиса `reordering_of_pic_nums_idc` обработаны в порядке их поступления в поток битов. Для каждого из этих элементов синтаксиса используют следующие условия:
 - если `reordering_of_pic_nums_idc` равно 0 или 1, активируют процесс, определенный в п. 8.2.4.3.1, с `refIdxL0` в качестве входа, и выход присваивается индексу `refIdxL0`;
 - иначе, если `reordering_of_pic_nums_idc` равно 2, активируют процесс, определенный в п. 8.2.4.3.2, с `refIdxL0` в качестве входа, и выход присваивается индексу `refIdxL0`;
 - иначе (`reordering_of_pic_nums_idc` равно 3) процесс изменения порядка в списке контрольных изображений `RefPicList0` заканчивается.

Если `ref_pic_list_reordering_flag_l1` равно 1, используют следующие условия.

- Допустим, что `refIdxL1` – индекс в списке контрольных изображений `RefPicList1`. Первоначально он установлен равным 0.
- Соответствующие элементы синтаксиса `reordering_of_pic_nums_idc` обработаны в порядке их поступления в поток битов. Для каждого из этих элементов синтаксиса используют следующие условия:
 - если `reordering_of_pic_nums_idc` равно 0 или 1, активируют процесс, определенный в п. 8.2.4.3.1, с `refIdxL1` в качестве входа, и выход присваивается индексу `refIdxL1`;
 - иначе, если `reordering_of_pic_nums_idc` равно 2, активируют процесс, определенный в п. 8.2.4.3.2, с `refIdxL1` в качестве входа, и выход присваивается индексу `refIdxL1`;
 - иначе (`reordering_of_pic_nums_idc` равно 3) процесс изменения порядка в списке контрольных изображений `RefPicList1` заканчивается.

8.2.4.3.1 Процесс изменения порядка в списках контрольных изображений для краткосрочных контрольных изображений

Входом в этот процесс является индекс `refFrameListXShortTerm` (с `X` равным 0 или 1).

Выходом этого процесса является индекс `refIdxLX` с приращением.

Переменную `picNumLXNoWrap` находят следующим образом.

- Если `reordering_of_pic_nums_idc` равно 0:

$$\begin{aligned} & \text{if}(\text{picNumLXPred} - (\text{abs_diff_pic_num_minus1} + 1) < 0) \\ & \quad \text{picNumLXNoWrap} = \text{picNumLXPred} - (\text{abs_diff_pic_num_minus1} + 1) + \text{MaxPicNum} \\ & \text{else} \\ & \quad \text{picNumLXNoWrap} = \text{picNumLXPred} - (\text{abs_diff_pic_num_minus1} + 1). \end{aligned} \quad (8-35)$$

- Иначе (`reordering_of_pic_nums_idc` равно 1):

$$\begin{aligned} & \text{if}(\text{picNumLXPred} + (\text{abs_diff_pic_num_minus1} + 1) \geq \text{MaxPicNum}) \\ & \quad \text{picNumLXNoWrap} = \text{picNumLXPred} + (\text{abs_diff_pic_num_minus1} + 1) - \text{MaxPicNum} \\ & \quad \text{picNumLXNoWrap} = \text{picNumLXPred} + (\text{abs_diff_pic_num_minus1} + 1). \end{aligned} \quad (8-36) \quad \text{else}$$

Значение `picNumLXPred` – это предсказанное значение переменной `picNumLXNoWrap`. Если процесс, определенный в этом пункте, активируют первый раз для секции (при первом появлении которой `reordering_of_pic_nums_idc` равно 0 или 1 в синтаксисе `ref_pic_list_reordering()`), то `picNumL0Pred` и `picNumL1Pred` первоначально устанавливаются равным `CurrPicNum`. После каждого присвоения `picNumLXNoWrap` значение `picNumLXNoWrap` присваивают параметру `picNumLXPred`.

Переменную `picNumLX` находят следующим образом:

$$\begin{aligned} & \text{if}(\text{picNumLXNoWrap} > \text{CurrPicNum}) \\ & \quad \text{picNumLX} = \text{picNumLXNoWrap} - \text{MaxPicNum} \\ & \text{else} \\ & \quad \text{picNumLX} = \text{picNumLXNoWrap}. \end{aligned} \quad (8-37)$$

Значение `picNumLX` должно быть равным `PicNum` контрольного изображения, которое помечено как "используется для краткосрочного контроля" и не должно быть равным значению `PicNum` краткосрочного контрольного изображения, которое помечено как "несуществующее".

Следующая процедура выполняется, чтобы заменить изображение с номером краткосрочного изображения `picNumLX` на индекс позиции `refIdxLX`, сдвинуть позиции всех других оставшихся изображений дальше в списке и прибавить значение индекса `refIdxLX`.

```

for( cIdx = num_ref_idx_IX_active_minus1 + 1; cIdx > refIdxLX; cIdx-- )
    RefPicListX[ cIdx ] = RefPicListX[ cIdx - 1 ]
RefPicListX[ refIdxLX++ ] = значению краткосрочного контрольного изображения с PicNum, равным
picNumLX
nIdx = refIdxLX
for( cIdx = refIdxLX; cIdx <= num_ref_idx_IX_active_minus1 + 1; cIdx++ )
    if( PicNumF( RefPicListX[ cIdx ] ) != picNumLX )
        RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ],

```

(8-38)

где функцию PicNumF(RefPicListX[cIdx]) находят следующим образом:

- если изображение RefPicListX[cIdx] помечено как "используется для краткосрочного контроля", то PicNumF(RefPicListX[cIdx]) – это значение PicNum изображения RefPicListX[cIdx];
- иначе (изображение RefPicListX[cIdx] не помечено как "используется для краткосрочного контроля") PicNumF(RefPicListX[cIdx]) равно MaxPicNum.

ПРИМЕЧАНИЕ 1. – Значение MaxPicNum никогда не может быть равным picNumLX.

ПРИМЕЧАНИЕ 2. – В этой процедуре псевдокодирования длина списка RefPicListX временно сделана на один элемент длиннее, чем необходимая длина для окончательного списка. После выполнения этой процедуры следует сохранить только элементы этого списка от 0 до num_ref_idx_IX_active_minus1.

8.2.4.3.2 Процесс изменения порядка в списках контрольных изображений для долгосрочных контрольных изображений

Входом в этот процесс является индекс refIdxLX (с X равным 0 или 1).

Выходом этого процесса является индекс refIdxLX с приращением.

Следующая процедура выполняется, чтобы заменить изображение с номером долгосрочного изображения long_term_pic_num на индекс позиции refIdxLX, сдвинуть позиции всех других оставшихся изображений дальше в списке и прибавить значение индекса refIdxLX:

```

for( cIdx = num_ref_idx_IX_active_minus1 + 1; cIdx > refIdxLX; cIdx-- )
    RefPicListX[ cIdx ] = RefPicListX[ cIdx - 1 ]
RefPicListX[ refIdxLX++ ] = значению долгосрочного контрольного изображения с LongTermPicNum,
равным long_term_pic_num
nIdx = refIdxLX
for( cIdx = refIdxLX; cIdx <= num_ref_idx_IX_active_minus1 + 1; cIdx++ )
    if( LongTermPicNumF( RefPicListX[ cIdx ] ) != long_term_pic_num )
        RefPicListX[ nIdx++ ] = RefPicListX[ cIdx ],

```

(8-39)

где функцию LongTermPicNumF(RefPicListX[cIdx]) находят следующим образом:

- если изображение RefPicListX[cIdx] помечено как "используется для долгосрочного контроля", то LongTermPicNumF(RefPicListX[cIdx]) – это значение LongTermPicNum изображения RefPicListX[cIdx];
- иначе (изображение RefPicListX[cIdx] не помечено как "используется для долгосрочного контроля") LongTermPicNumF(RefPicListX[cIdx]) равно 2 * (MaxLongTermFrameIdx + 1).

ПРИМЕЧАНИЕ 1. – Значение 2 * (MaxLongTermFrameIdx + 1) никогда не может быть равным long_term_pic_num.

ПРИМЕЧАНИЕ 2. – В этой процедуре псевдокодирования длина списка RefPicListX временно сделана на один элемент длиннее, чем необходимая длина для окончательного списка. После выполнения этой процедуры следует сохранить только элементы этого списка от 0 до num_ref_idx_IX_active_minus1.

8.2.5 Процесс разметки декодированного контрольного изображения

Этот процесс активируют для декодирования изображений, если nal_ref_idc не равно 0.

ПРИМЕЧАНИЕ. – Процесс декодирования промежутков в frame_num, определенный в п. 0, может быть также активирован, если nal_ref_idc равно 0, как определено в разделе 8.

Декодированное изображение с nal_ref_idc, не равным 0, отнесенное к контрольному изображению, помечено как "используется для краткосрочного контроля" или "используется для долгосрочного контроля". Оба поля декодированного контрольного кадра помечены так же, как кадр. Пара дополнительных контрольных полей помечена так же, как эти поля. Изображение, которое помечено как "используется для краткосрочного контроля", идентифицировано его номером FrameNum и, если это – поле, то его четностью. Изображение, которое помечено как "используется для долгосрочного контроля", идентифицировано его индексом LongTermFrameIdx и, если это – поле, то его четностью.

Кадры или пара дополнительных полей, помеченные как "используется для краткосрочного контроля" или как "используется для долгосрочного контроля", можно использовать при декодировании кадра как контрольные для внешнего предсказания до появления кадра, пары дополнительных полей или одной из составляющих поля, помеченных как "не используется для контроля". Поле, помеченное как "используется для краткосрочного контроля"

или как "используется для долгосрочного контроля", можно использовать при декодировании поля как контрольное для внешнего предсказания до появления поля, помеченного как "не используется для контроля".

Изображение может быть помечено как "не используется для контроля" с помощью скользящего окна в процессе разметки контрольного изображения, механизмом "первым пришел – первым обслужен", описанным в п. 8.2.5.3 или процессом разметки контрольного изображения с адаптивным управлением памятью, приспособленной к адаптивной операции разметки (п. 8.2.5.4).

Краткосрочное контрольное изображение идентифицировано для использования в процессе декодирования переменными `FrameNum` и `FrameNumWrap` и изображения `PicNum`, а долгосрочное контрольное изображение идентифицировано для использования в процессе декодирования номером долгосрочного изображения `LongTermPicNum`. Если текущее изображение – это не IDR изображение, то активируют процедуру п. 8.2.4.1, чтобы определить присвоение переменных `FrameNum`, `FrameNumWrap`, `PicNum` и `LongTermPicNum`.

8.2.5.1 Последовательность операций для процесса разметки декодированного контрольного изображения

Разметку декодированного контрольного изображения производят следующими шагами.

1. Все секции текущего изображения декодированы.
2. В зависимости от того, является ли текущее изображение IDR изображением, используют следующие условия.
 - Если текущее изображение – это IDR изображение, используют следующие условия.
 - Все контрольные изображения помечаются как "не используется для контроля".
 - В зависимости от значения `long_term_reference_flag`, используют следующие условия:
 - если `long_term_reference_flag` равно 0, IDR изображение помечается как "используется для краткосрочного контроля", а `MaxLongTermFrameIdx` устанавливается равным "нет индексов долгосрочных кадров";
 - иначе (`long_term_reference_flag` равно 1) IDR изображение помечается как "используется для долгосрочного контроля", `LongTermFrameIdx` для IDR изображения устанавливается равным 0 и `MaxLongTermFrameIdx` устанавливается равным 0.
 - Иначе (текущее изображение – не IDR изображение) используют следующие условия:
 - если `adaptive_ref_pic_marking_mode_flag` равно 0, активируют процесс, определенный в п. 8.2.5.3;
 - иначе (`adaptive_ref_pic_marking_mode_flag` равно 1) активируют процесс, определенный в п. 8.2.5.4.
3. Если текущее изображение – не IDR изображение и не было помечено как "используется для долгосрочного контроля" значением `memory_management_control_operation`, равным 6, его отмечают как "используется для краткосрочного контроля".

После разметки текущего декодированного контрольного изображения общее число кадров, по крайней мере, с одним полем, помеченным как "используется для контроля", плюс число пар дополнительных полей, по крайней мере, с одним полем, помеченным как "используется для контроля", плюс число непарных полей, помеченных как "используется для контроля", не должно быть больше, чем $\text{Max}(\text{num_ref_frames}, 1)$.

8.2.5.2 Процесс декодирования для промежутков в `frame_num`

Этот процесс активируют, если `frame_num` не равно `PrevRefFrameNum` и не равно $(\text{PrevRefFrameNum} + 1) \% \text{MaxFrameNum}$.

ПРИМЕЧАНИЕ 1. – Хотя этот процесс определен в виде пункта внутри п. 8.2.5 (в котором описан процесс, который активируют, только если `nal_ref_idc` не равно 0), данный процесс можно также активировать, если `nal_ref_idc` равно 0 (как определено в разделе 8). Причины, по которым этот пункт помещен в структуру настоящей Рекомендации | Международного стандарта, – чисто исторические.

ПРИМЕЧАНИЕ 2. – Этот процесс можно активировать только для соответствующего потока битов, если `gaps_in_frame_num_value_allowed_flag` равно 1. Если `gaps_in_frame_num_value_allowed_flag` равно 0, а `frame_num` не равно `PrevRefFrameNum` и не равно $(\text{PrevRefFrameNum} + 1) \% \text{MaxFrameNum}$, процесс декодирования следует считать непреднамеренной потерей изображений.

Если этот процесс активируют, набор значений `frame_num`, принадлежащих к "несуществующим" изображениям, находят как все полученные в равнении 7-21 значения `UnusedShortTermFrameNum`, за исключением `frame_num` для текущего изображения.

Процесс декодирования создает и размечает кадр для каждого из его значений `frame_num`, принадлежащих к "несуществующим" изображениям в порядке, в котором значения `UnusedShortTermFrameNum` создаются уравнением 7-21, используя "скользящее окно" изображения процесса разметки, как определено в п. 8.2.5.3. Созданные кадры также помечаются как "несуществующие" и "используется для краткосрочного контроля". Образец значений созданных кадров можно установить на любое значение. Поток битов не должен содержать данных, которые образуют ссылку на эти созданные кадры, помеченные как "несуществующие" в процессе внешнего предсказания, ссылку на эти кадры в командах изменения порядка в списках контрольных изображений для краткосрочных контрольных изображений (п. 8.2.4.3.1) или ссылку на эти кадры в процессе присвоения `LongTermFrameIdx` краткосрочному контрольному изображению (п. 8.2.5.4.3).

Если `pic_order_cnt_type` не равно 0, `TopFieldOrderCnt` и `BottomFieldOrderCnt` находят для каждого "несуществующего" кадра, активируя процесс декодирования для вычисления порядка изображений в п. 8.2.1. Если активируют процесс в п. 8.2.1 для конкретного "несуществующего" кадра, текущее изображение считают изображением, имеющим значение `frame_num`, которое считают равным `UnusedShortTermFrameNum`, `nal_ref_idc` – равным 0, `nal_unit_type` – равным 5, `field_pic_flag` – равным 0, `adaptive_ref_pic_marking_mode_flag` – равным 0, `delta_pic_order_cnt[0]` (если необходимо) – равным 0 и `delta_pic_order_cnt[1]` (если необходимо) – равным 0.

ПРИМЕЧАНИЕ 3. – Процесс декодирования следует считать непреднамеренной потерей изображения, если любое из значений `frame_num`, относящиеся к "несуществующим" изображениям, считают процессом внешнего предсказания, изменением порядка команд в списках контрольных изображений для краткосрочных контрольных изображений (п. 8.2.4.3.1) или полагают принадлежащим к процессу присвоения `LongTermFrameIdx` краткосрочному контрольному изображению (п. 8.2.5.4.3). Процесс декодирования не следует считать непреднамеренной потерей изображения, если операцию управления памятью, не равную 3, применяют к кадру, помеченному как "несуществующий".

8.2.5.3 Скользящее окно процесса разметки декодированного контрольного изображения

Этот процесс активируют, если `adaptive_ref_pic_marking_mode_flag` равно 0.

В зависимости от свойств текущего изображения, как будет описано ниже, используют следующие условия.

- Если текущее изображение – кодированное поле, то есть второе поле в порядке декодирования пары дополнительных контрольных полей, а первое поле помечено как "используется для краткосрочного контроля", то текущее изображение также помечают как "используется для краткосрочного контроля".
- Иначе используют следующие условия.
 - Допустим, что `numShortTerm` – общее число контрольных кадров, пары дополнительных контрольных полей и непарных контрольных полей, для которых, по крайней мере, одно поле помечено как "используется для краткосрочного контроля". Допустим, что `numLongTerm` – общее число контрольных кадров, пары дополнительных контрольных полей и непарных контрольных полей, для которых, по крайней мере, одно поле помечено как "используется для долгосрочного контроля".
 - Если `numShortTerm + numLongTerm` равно $\text{Max}(\text{num_ref_frames}, 1)$, условие, что `numShortTerm` больше 0, должно быть выполнено, а краткосрочный контрольный кадр, пара дополнительных контрольных полей или непарное контрольное поле, которое имеет наименьшее значение `FrameNumWrap`, – помечают как "не используется для контроля". Если это – кадр или пара дополнительных полей, то оба эти поля также помечают как "не используется для контроля".

8.2.5.4 Процесс разметки декодированного контрольного изображения с адаптивным управлением памятью

Этот процесс активируют, если `adaptive_ref_pic_marking_mode_flag` равно 1.

Команды `memory_management_control_operation` со значениями от 1 до 6 обрабатывают в порядке их поступления в потоке битов после того, как декодировано текущее изображение. Для каждой из команд `memory_management_control_operation` активируют один из процессов, определенных в пп. 8.2.5.4.1–8.2.5.4.5, в зависимости от значения `memory_management_control_operation`. Команда `memory_management_control_operation` со значением 0 определяет конец команд `memory_management_control_operation`.

Операции управления памятью применяют к изображениям следующим образом:

- если `field_pic_flag` равно 0, команды `memory_management_control_operation` применяют к определенным кадрам или парам дополнительных контрольных полей;
- иначе (`field_pic_flag` равно 1) команды `memory_management_control_operation` применяют к отдельным определенным контрольным полям.

8.2.5.4.1 Процесс разметки краткосрочного контрольного изображения как "не используется для контроля"

Этот процесс активируют, если `memory_management_control_operation` равно 1.

Допустим, что `picNumX` определено как:

$$\text{picNumX} = \text{CurrPicNum} - (\text{difference_of_pic_nums_minus1} + 1). \quad (8-40)$$

В зависимости от `field_pic_flag` значение `picNumX` используют для разметки краткосрочных контрольных изображений как "не используется для контроля" следующим образом:

- если `field_pic_flag` равно 0, краткосрочный контрольный кадр или краткосрочную пару дополнительных контрольных полей, определенных `picNumX`, и оба этих поля помечают как "не используется для контроля";
- иначе (`field_pic_flag` равно 1) краткосрочное контрольное поле, определенное `picNumX`, помечают как "не используется для контроля". Если это контрольное поле – часть контрольного кадра или пары дополнительных контрольных полей, то кадр или пару дополнительных полей также помечают как "не используется для контроля", но разметку другого поля не меняют.

8.2.5.4.2 Процесс разметки долгосрочного контрольного изображения как "не используется для контроля"

Этот процесс активируют, если `memory_management_control_operation` равно 2.

В зависимости от `field_pic_flag` используют значение `LongTermPicNum` для разметки долгосрочного контрольного изображения как "не используется для контроля" следующим образом:

- если `field_pic_flag` равно 0, долгосрочный контрольный кадр или долгосрочную пару дополнительных контрольных полей, имеющих `LongTermPicNum`, равное `long_term_pic_num`, и оба этих поля помечают как "не используется для контроля";
- иначе (`field_pic_flag` равно 1) долгосрочное контрольное поле, определенное `LongTermPicNum`, равным `long_term_pic_num`, помечают как "не используется для контроля". Если это контрольное поле – часть контрольного кадра или пары дополнительных контрольных полей, то кадр или пару дополнительных полей также помечают как "не используется для контроля", но разметку другого поля не меняют.

8.2.5.4.3 Процесс присвоения значения `LongTermFrameIdx` краткосрочному контрольному изображению

Этот процесс активируют, если `memory_management_control_operation` равно 3.

Для заданного элемента синтаксиса `difference_of_pic_nums_minus1` переменную `picNumX` находят, как определено в п. 8.2.5.4.1. Значение `picNumX` должно относиться к кадру или к паре дополнительных контрольных полей или непарных контрольных полей, помеченных как "используется для краткосрочного контроля" и не помеченных как "несуществующие".

Если `LongTermFrameIdx`, равное `long_term_frame_idx`, уже присвоено долгосрочному контрольному кадру или долгосрочной паре дополнительных контрольных полей, то этот кадр или пара дополнительных полей и оба поля помечают как "не используется для контроля". Если `LongTermFrameIdx` уже присвоено непарному контрольному полю, и это поле не является дополнительным полем изображения, определенного `picNumX`, то это поле помечают как "не используется для контроля".

В зависимости от `field_pic_flag` значение `LongTermFrameIdx` используют для разметки изображения от "используется для краткосрочного контроля" до "используется для долгосрочного контроля" следующим образом:

- если `field_pic_flag` равно 0, разметку краткосрочного контрольного кадра или краткосрочной пары дополнительных контрольных полей, определенных `picNumX`, а также обоих этих полей меняют с "используется для краткосрочного контроля" на "используется для долгосрочного контроля" и присваивают `LongTermFrameIdx` значение `long_term_frame_idx`;
- иначе (`field_pic_flag` равно 1) разметку краткосрочного контрольного поля, определенного `picNumX`, меняют с "используется для краткосрочного контроля" на "используется для долгосрочного контроля" и присваивают `LongTermFrameIdx` значение `long_term_frame_idx`. Если поле является частью контрольного кадра или пары дополнительных контрольных полей, и другое поле этого же контрольного кадра или пары дополнительных контрольных полей тоже помечено как "используется для долгосрочного контроля", этот контрольный кадр или пара дополнительных контрольных полей также помечается как "используется для долгосрочного контроля", а значение `LongTermFrameIdx` назначается равным `long_term_frame_idx`.

8.2.5.4.4 Процесс декодирования `MaxLongTermFrameIdx`

Этот процесс активируют, если `memory_management_control_operation` равно 4.

Все изображения, для которых `LongTermFrameIdx` больше, чем `max_long_term_frame_idx_plus1 - 1`, и которые помечены как "используется для долгосрочного контроля", помечаются как "не используется для контроля".

Переменную `MaxLongTermFrameIdx` находят следующим образом:

- если `max_long_term_frame_idx_plus1` равно 0, `MaxLongTermFrameIdx` устанавливается равным "нет индексов долгосрочных кадров";
- иначе (`max_long_term_frame_idx_plus1` больше, чем 0) `MaxLongTermFrameIdx` устанавливается равным `max_long_term_frame_idx_plus1 - 1`.

ПРИМЕЧАНИЕ. – Команда `memory_management_control_operation`, равная 4, может быть использована для разметки долгосрочных контрольных изображений как "не используется для контроля". Частота передачи значения `max_long_term_frame_idx_plus1` не определена настоящей Рекомендацией | Международным стандартом. Однако кодеру следует отправлять команду `memory_management_control_operation`, равную 4, до получения сообщения ошибки, как, например, сообщения о запросе на внутреннее обновление.

8.2.5.4.5 Процесс разметки всех контрольных изображений как "не используется для контроля" и установка `MaxLongTermFrameIdx` на значение "нет индексов долгосрочных кадров"

Этот процесс активируют, если `memory_management_control_operation` равно 5.

Все контрольные изображения помечены как "не используется для контроля", а переменную `MaxLongTermFrameIdx` устанавливают равной "нет индексов долгосрочных кадров".

8.2.5.4.6 Процесс присвоения текущему изображению индекса долгосрочного кадра

Этот процесс активируют, если `memory_management_control_operation` равно 6.

Если переменная `LongTermFrameIdx`, равная `long_term_frame_idx`, уже присвоена долгосрочному контрольному кадру или долгосрочной паре дополнительных контрольных полей, то этот кадр или пара дополнительных полей и оба этих поля помечают как "не используется для контроля". Если `LongTermFrameIdx` уже присвоено непарному контрольному полю, а это поле не является дополнительным для текущего изображения, то это поле помечают как "не используется для контроля".

Текущее изображение помечено как "используется для долгосрочного контроля", и ему присваивают индекс `LongTermFrameIdx`, равный `long_term_frame_idx`.

Если `field_pic_flag` равно 0, оба поля помечены как "используется для долгосрочного контроля" и им присваивают индекс `LongTermFrameIdx`, равный `long_term_frame_idx`.

Если `field_pic_flag` равно 1, а текущее изображение – это второе (в порядке декодирования) поле пары дополнительных контрольных полей, и первое поле этой пары дополнительных контрольных полей также помечено как "используется для долгосрочного контроля", эта пара дополнительных контрольных полей также помечается как "используется для долгосрочного контроля" и ей присваивают индекс `LongTermFrameIdx`, равный `long_term_frame_idx`.

После разметки текущего декодированного контрольного изображения общее число кадров, по крайней мере, с одним полем, помеченным как "используется для контроля", плюс число пар дополнительных полей по крайней мере с одним полем, помеченным как "используется для контроля", плюс число непарных полей, помеченных как "используется для контроля", не должно превышать $\text{Max}(\text{num_ref_frames}, 1)$.

ПРИМЕЧАНИЕ. – При некоторых обстоятельствах вышеуказанное утверждение может накладывать ограничение на порядок, в котором элемент синтаксиса `memory_management_control_operation`, равный 6, может появиться в декодированном контрольном изображении, помечая синтаксис относительно элемента синтаксиса со значением `memory_management_control_operation`, равным 1, 2 или 4.

8.3 Процесс внутреннего предсказания (Intra предсказания)

Этот процесс активируют для типов макроблоков I и SI.

Входами в этот процесс являются созданные образцы до процесса фильтра устранения блочности и, для режимов предсказания `Intra_NxN` (где `NxN` равно 4x4 или 8x8), значения `IntraNxNPredMode` от смежных макроблоков.

Выходы этого процесса определены следующим образом.

- Если режимом предсказания макроблока является `Intra_4x4` или `Intra_8x8`, выходами являются созданные образцы яркости до процесса фильтра устранения блочности и (если `chroma_format_idc` не равно 0) образцы предсказания цветности `pred_c` макроблока, если `C` равно `Cb` и `Cr`.
- Иначе, если `mb_type` не равно `I_PCM`, выходами являются образцы предсказания яркости `pred_l` макроблока и (если `chroma_format_idc` не равно 0) образцы предсказания цветности `pred_c` макроблока, если `C` равно `Cb` и `Cr`.
- Иначе (`mb_type` равно `I_PCM`) выходами являются созданные образцы яркости и (если `chroma_format_idc` не равно 0) цветности до процесса фильтра устранения блочности.

Переменную `MvCnt` устанавливают равной 0.

В зависимости от значения `mb_type` используют следующие условия.

- Если `mb_type` равно `I_PCM`, активируют процесс, определенный в п. 8.3.5.
- Иначе (`mb_type` не равно `I_PCM`) используют следующие условия.
 - Процесс декодирования в режимах `Intra` предсказания описывают для компонентов яркости следующим образом:
 - если режим предсказания макроблока равен `Intra_4x4`, применяют спецификацию п. 8.3.1;
 - иначе (режим предсказания макроблока равен `Intra_8x8`) применяют спецификацию п. 8.3.2;
 - иначе (режим предсказания макроблока равен `Intra_16x16`) применяют спецификацию п. 8.3.3.
 - Процесс декодирования в режимах `Intra` предсказания для компонентов цветности описан в п. 8.3.4. Этот процесс активируется, только когда `chroma_format_idc` не равно 0 (монохромное изображение).

Образцы, использованные в процессе `Intra` предсказания, являются образцами до их изменения любой операцией фильтра устранения блочности.

8.3.1 Процесс предсказания образцов яркости Intra_4x4

Этот процесс активируют, если режим предсказания макроблока равен Intra_4x4.

Входами в этот процесс являются значения Intra4x4PredMode (если имеются) или Intra8x8PredMode (если имеются) от смежных макроблоков или пар макроблоков.

Компонент яркости макроблока состоит из 16 блоков 4x4 образцов яркости. Эти блоки – инверсно сканированные, с использованием процесса инверсного сканирования блока яркости 4x4, который описан в п. 6.4.3.

Для всех блоков яркости 4x4 компонента яркости макроблока с luma4x4BlkIdx = 0..15 процесс вывода для Intra4x4PredMode, описанный в п. 8.3.1.1, активируется с luma4x4BlkIdx, а также Intra4x4PredMode и Intra8x8PredMode, которые ранее (в порядке декодирования) были найдены для смежных макроблоков в качестве входа, и переменной Intra4x4PredMode[luma4x4BlkIdx] в качестве выхода.

Для всех образцов блока яркости 4x4, проиндексированных с использованием luma4x4BlkIdx = 0..15.

1. Активируется процесс предсказания образца Intra_4x4 согласно п. 8.3.1.2 с luma4x4BlkIdx и созданными образцами до (в порядке декодирования) процесса фильтра устранения блочности от смежных блоков яркости в качестве входа, а выходом являются образцы предсказания яркости Intra_4x4 pred4x4_L[x, y] с x, y = 0..3.
2. Положение верхнего левого образца блока яркости 4x4 с индексом luma4x4BlkIdx внутри текущего макроблока находят, активируя процесс инверсного сканирования блока яркости 4x4 согласно 6.4.3 с luma4x4BlkIdx в качестве входа и выходом, которому присваивается (xO, yO) с x, y = 0..3.

$$\text{pred}_L[xO + x, yO + y] = \text{pred4x4}_L[x, y]. \quad (8-41)$$

3. Процесс декодирования коэффициентов преобразования и процесс создания изображения согласно п. 8.5 активируют до процесса фильтра устранения блочности с pred_L и luma4x4BlkIdx в качестве входа и созданными образцами текущего блока яркости 4x4 luma block S'_L в качестве выхода.

8.3.1.1 Процесс вывода Intra4x4PredMode

Входы в этот процесс – это индекс блока яркости 4x4 luma4x4BlkIdx и массивы переменных Intra4x4PredMode (если доступен) и Intra8x8PredMode (если доступен), которые предварительно (в порядке декодирования) находят для смежных макроблоков.

Выход этого процесса – переменная Intra4x4PredMode[luma4x4BlkIdx].

В таблице 8-2 показаны значения Intra4x4PredMode[luma4x4BlkIdx] и связанные с ними имена.

Таблица 8-2 – Спецификация Intra4x4PredMode[luma4x4BlkIdx] и связанных имен

Intra4x4PredMode[luma4x4BlkIdx]	Имя Intra4x4PredMode[luma4x4BlkIdx]
0	Intra_4x4_Vertical (режим предсказания)
1	Intra_4x4_Horizontal (режим предсказания)
2	Intra_4x4_DC (режим предсказания)
3	Intra_4x4_Diagonal_Down_Left (режим предсказания)
4	Intra_4x4_Diagonal_Down_Right (режим предсказания)
5	Intra_4x4_Vertical_Right (режим предсказания)
6	Intra_4x4_Horizontal_Down (режим предсказания)
7	Intra_4x4_Vertical_Left (режим предсказания)
8	Intra_4x4_Horizontal_Up (режим предсказания)

Intra4x4PredMode[luma4x4BlkIdx], обозначенные как 0, 1, 3, 4, 5, 6, 7 и 8, представляют направления предсказаний, как показано на рисунке 8-1.

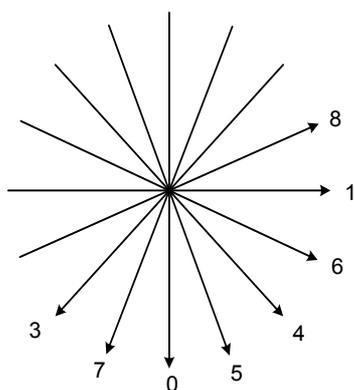


Рисунок 8-1 – Направления режимов предсказания Intra_4x4 (для информации)

Intra4x4PredMode[luma4x4BlkIdx] находят следующим образом.

- Процесс, определенный в п. 6.4.8.3, активируют с luma4x4BlkIdx, заданным как вход, а выход присваивается значениям mbAddrA, luma4x4BlkIdxA, mbAddrB и luma4x4BlkIdxB.
- Переменную dcPredModePredictedFlag находят следующим образом.
 - Если любые из следующих условий – истина, то dcPredModePredictedFlag устанавливают равным 1:
 - макроблок с адресом mbAddrA не доступен;
 - макроблок с адресом mbAddrB не доступен;
 - макроблок с адресом mbAddrA доступен и кодирован в режиме Inter предсказания с constrained_intra_pred_flag, равным 1;
 - макроблок с адресом mbAddrB доступен и кодирован в режиме Inter предсказания с constrained_intra_pred_flag, равным 1.
 - Иначе dcPredModePredictedFlag установлено равным 0.
- Для значения N, которое заменяют либо значением A, либо B, переменные ntraMxMPredModeN находят следующим образом.
 - Если dcPredModePredictedFlag равно 1 или макроблок с адресом mbAddrN кодирован не в режиме предсказания макроблока Intra_4x4 или Intra_8x8, то intraMxMPredModeN устанавливают равным 2 (режим предсказания Intra_4x4_DC).
 - Иначе (dcPredModePredictedFlag равно 0 и (макроблок с адресом mbAddrN кодирован в режиме предсказания макроблока Intra_4x4 или макроблок с адресом mbAddrN кодирован в режиме предсказания макроблока Intra_8x8)) применяется следующее:
 - если макроблок с адресом mbAddrN кодирован в режиме макроблока Intra_4x4, intraMxMPredModeN устанавливают равным Intra4x4PredMode[luma4x4BlkIdxN], где Intra4x4PredMode – массив переменных, присвоенных макроблоку mbAddrN;
 - иначе (макроблок с адресом mbAddrN кодирован в режиме макроблока Intra_8x8) intraMxMPredModeN устанавливают равным Intra8x8PredMode[luma4x4BlkIdxN >> 2], где Intra8x8PredMode – массив переменных, присвоенных макроблоку mbAddrN.
- Intra4x4PredMode[luma4x4BlkIdx] находят, применяя следующую процедуру:

```

predIntra4x4PredMode = Min( intraMxMPredModeA, intraMxMPredModeB )
if( prev_intra4x4_pred_mode_flag[ luma4x4BlkIdx ] )
  Intra4x4PredMode[ luma4x4BlkIdx ] = predIntra4x4PredMode
else
  if( rem_intra4x4_pred_mode[ luma4x4BlkIdx ] < predIntra4x4PredMode )
    Intra4x4PredMode[ luma4x4BlkIdx ] = rem_intra4x4_pred_mode[ luma4x4BlkIdx ]
  else
    Intra4x4PredMode[ luma4x4BlkIdx ] = rem_intra4x4_pred_mode[ luma4x4BlkIdx ] + 1.
  
```

(8-42)

8.3.1.2 Предсказание образца Intra_4x4

Этот процесс активируют для каждого блока яркости 4x4 макроблока с режимом предсказания, равным Intra_4x4, за которым следует преобразование процесса декодирования и процесс создания изображения перед разделением на блоки яркости 4x4.

Входом в этот процесс является индекс блока яркости 4x4 luma4x4BlkIdx.

Выход этого процесса – предсказанные образцы $\text{pred4x4}_L[x, y]$, с $x, y = 0..3$ для блока яркости 4x4 с индексом luma4x4BlkIdx.

Положение верхнего левого образца блока яркости 4x4 с индексом luma4x4BlkIdx внутри текущего макроблока находят, активируя процесс инверсного сканирования блока яркости 4x4 согласно п. 6.4.3 с luma4x4BlkIdx в качестве входа и выходом, присвоенным (xO, yO).

13 смежных образцов $p[x, y]$, которые являются созданными образцами яркости до процесса фильтра устранения блочности с $x = -1, y = -1..3$ и $x = 0..7, y = -1$, находят следующим образом.

– Положение яркости (xN, yN) определяется как:

$$xN = xO + x; \quad (8-43)$$

$$yN = yO + y. \quad (8-44)$$

– Процесс вывода смежных местоположений согласно п.6.4.9 активируют для местоположений яркости с (xN, yN) в качестве входа и mbAddrN и (xW, yW) в качестве выхода.

– Каждый образец $p[x, y]$ с $x = -1, y = -1..3$ и $x = 0..7, y = -1$ находят следующим образом.

– Если любое из следующих условий – истина, то образец $p[x, y]$ помечен как "не доступен для предсказания Intra_4x4":

– mbAddrN не доступно;

– макроблок mbAddrN кодирован в режиме Inter предсказания, а constrained_intra_pred_flag равно 1;

– тип mb_type макроблока mbAddrN равен SI с constrained_intra_pred_flag, равным 1, а текущий макроблок не имеет mb_type, равного SI;

– x больше 3, а luma4x4BlkIdx равно 3 или 11.

– Иначе образец $p[x, y]$ помечен "доступен для предсказания Intra_4x4", а образец яркости в местоположении яркости (xW, yW) внутри макроблока mbAddrN присваивается $p[x, y]$.

Если образцы $p[x, -1]$ с $x = 4..7$ помечены как "не доступен для предсказания Intra_4x4", а образец $p[3, -1]$ помечен как "доступен для предсказания Intra_4x4," то значение образца $p[3, -1]$ заменяют на значения образца $p[x, -1]$ с $x = 4..7$, а образцы $p[x, -1]$ с $x = 4..7$ помечают как "доступен для предсказания Intra_4x4".

ПРИМЕЧАНИЕ. – Считают, что каждый блок создан в массиве изображений до декодирования следующего блока.

В зависимости от Intra4x4PredMode[luma4x4BlkIdx] активируется один из режимов предсказания Intra_4x4, определенных в пп. 8.3.1.2.1–8.3.1.2.9.

8.3.1.2.1 Описание режима предсказания Intra_4x4_Vertical

Режим предсказания Intra_4x4 активируется, если Intra4x4PredMode[luma4x4BlkIdx] равно 0.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..3$ помечены как "доступен для предсказания Intra_4x4".

Значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $y = 0..3$ находят следующим образом:

$$\text{pred4x4}_L[x, y] = p[x, -1] \text{ с } x, y = 0..3. \quad (8-45)$$

8.3.1.2.2 Описание режима предсказания Intra_4x4_Horizontal

Режим предсказания Intra_4x4 активируется, если Intra4x4PredMode[luma4x4BlkIdx] равно 1.

Этот режим должен использоваться, только если образцы $p[-1, y]$ с $y = 0..3$ помечены как "доступен для предсказания Intra_4x4".

Значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом:

$$\text{pred4x4}_L[x, y] = p[-1, y] \text{ с } x, y = 0..3. \quad (8-46)$$

8.3.1.2.3 Описание режима предсказания Intra_4x4_DC

Режим предсказания Intra_4x4 активируется, если Intra4x4PredMode[luma4x4BlkIdx] равно 2.

Значения предсказания образцов $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом.

- Если все образцы $p[x, -1]$ с $x = 0..3$ и $p[-1, y]$ с $y = 0..3$ помечены как "доступен для предсказания Intra_4x4", то значения предсказания образцов $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом:

$$\text{pred4x4}_L[x, y] = (p[0, -1] + p[1, -1] + p[2, -1] + p[3, -1] + p[-1, 0] + p[-1, 1] + p[-1, 2] + p[-1, 3] + 4) \ggg 3. \quad (8-47)$$

- Иначе, если имеются образцы $p[x, -1]$ с $x = 0..3$, помеченные как "не доступен для предсказания Intra_4x4", а все образцы $p[-1, y]$ с $y = 0..3$ помечены как "доступен для предсказания Intra_4x4", значения предсказания образцов $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом:

$$\text{pred4x4}_L[x, y] = (p[-1, 0] + p[-1, 1] + p[-1, 2] + p[-1, 3] + 2) \ggg 2. \quad (8-48)$$

- Иначе, если имеются образцы $p[-1, y]$ с $y = 0..3$, помеченные как "не доступен для предсказания Intra_4x4", а все образцы $p[x, -1]$ с $x = 0..3$ помечены как "доступен для предсказания Intra_4x4", значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом:

$$\text{pred4x4}_L[x, y] = (p[0, -1] + p[1, -1] + p[2, -1] + p[3, -1] + 2) \ggg 2. \quad (8-49)$$

- Иначе (некоторые образцы $p[x, -1]$ с $x = 0..3$ и некоторые образцы $p[-1, y]$ с $y = 0..3$ помечены как "не доступен для предсказания Intra_4x4") значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом:

$$\text{pred4x4}_L[x, y] = (1 \ll (\text{BitDepth}_Y - 1)). \quad (8-50)$$

ПРИМЕЧАНИЕ. – Используя этот режим всегда возможно предсказать блок яркости 4x4.

8.3.1.2.4 Описание режима предсказания Intra_4x4_Diagonal_Down_Left

Режим предсказания Intra_4x4 активируется, если Intra4x4PredMode[luma4x4BlkIdx] равно 3.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..7$ помечены как "доступен для предсказания Intra_4x4".

Значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом.

- Если x равно 3 и y равно 3:

$$\text{pred4x4}_L[x, y] = (p[6, -1] + 3 * p[7, -1] + 2) \ggg 2. \quad (8-51)$$

- Иначе (x не равно 3 или y не равно 3):

$$\text{pred4x4}_L[x, y] = (p[x + y, -1] + 2 * p[x + y + 1, -1] + p[x + y + 2, -1] + 2) \ggg 2. \quad (8-52)$$

8.3.1.2.5 Описание режима предсказания Intra_4x4_Diagonal_Down_Right

Режим предсказания Intra_4x4 активируется, если Intra4x4PredMode[luma4x4BlkIdx] равно 4.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..3$ и $p[-1, y]$ с $y = -1..3$ помечены как "доступен для предсказания Intra_4x4".

Значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом.

- Если x больше y :

$$\text{pred4x4}_L[x, y] = (p[x - y - 2, -1] + 2 * p[x - y - 1, -1] + p[x - y, -1] + 2) \gg 2. \quad (8-53)$$

- Иначе, если x меньше y :

$$\text{pred4x4}_L[x, y] = (p[-1, y - x - 2] + 2 * p[-1, y - x - 1] + p[-1, y - x] + 2) \gg 2. \quad (8-54)$$

- Иначе (x равно y):

$$\text{pred4x4}_L[x, y] = (p[0, -1] + 2 * p[-1, -1] + p[-1, 0] + 2) \gg 2. \quad (8-55)$$

8.3.1.2.6 Описание режима предсказания Intra_4x4_Vertical_Right

Режим предсказания Intra_4x4 активируется, если Intra4x4PredMode[luma4x4BlkIdx] равно 5.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..3$ и $p[-1, y]$ с $y = -1..3$ помечены как "доступен для предсказания Intra_4x4".

Допустим, что переменная zVR установлена равной $2 * x - y$.

Значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом.

- Если zVR равно 0, 2, 4 или 6:

$$\text{pred4x4}_L[x, y] = (p[x - (y \gg 1) - 1, -1] + p[x - (y \gg 1), -1] + 1) \gg 1. \quad (8-56)$$

- Иначе, если zVR равно 1, 3 или 5:

$$\text{pred4x4}_L[x, y] = (p[x - (y \gg 1) - 2, -1] + 2 * p[x - (y \gg 1) - 1, -1] + p[x - (y \gg 1), -1] + 2) \gg 2. \quad (8-57)$$

- Иначе, если zVR равно -1:

$$\text{pred4x4}_L[x, y] = (p[-1, 0] + 2 * p[-1, -1] + p[0, -1] + 2) \gg 2. \quad (8-58)$$

- Иначе (zVR равно -2 или -3):

$$\text{pred4x4}_L[x, y] = (p[-1, y - 1] + 2 * p[-1, y - 2] + p[-1, y - 3] + 2) \gg 2. \quad (8-59)$$

8.3.1.2.7 Описание режима предсказания Intra_4x4_Horizontal_Down

Режим предсказания Intra_4x4 активируется, если Intra4x4PredMode[luma4x4BlkIdx] равно 6.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..3$ и $p[-1, y]$ с $y = -1..3$ помечены как "доступен для предсказания Intra_4x4".

Допустим, что переменная zHD установлена равной $2 * y - x$.

Значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом.

- Если zHD равно 0, 2, 4 или 6:

$$\text{pred4x4}_L[x, y] = (p[-1, y - (x \gg 1) - 1] + p[-1, y - (x \gg 1)] + 1) \gg 1. \quad (8-60)$$

- Иначе, если zHD равно 1, 3 или 5:

$$\text{pred4x4}_L[x, y] = (p[-1, y - (x \gg 1) - 2] + 2 * p[-1, y - (x \gg 1) - 1] + p[-1, y - (x \gg 1)] + 2) \gg 2. \quad (8-61)$$

- Иначе, если zHD равно -1:

$$\text{pred4x4}_L[x, y] = (p[-1, 0] + 2 * p[-1, -1] + p[0, -1] + 2) \gg 2. \quad (8-62)$$

- Иначе (zHD равно –2 или –3):

$$\text{pred4x4}_L[x, y] = (\text{p}[x - 1, -1] + 2 * \text{p}[x - 2, -1] + \text{p}[x - 3, -1] + 2) \gg 2. \quad (8-63)$$

8.3.1.2.8 Описание режима предсказания Intra_4x4_Vertical_Left

Режим предсказания Intra_4x4 активируется, если Intra4x4PredMode[luma4x4BlkIdx] равно 7.

Этот режим должен использоваться, только если образцы $\text{p}[x, -1]$ с $x = 0..7$ помечены как "доступен для предсказания Intra_4x4".

Значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом.

- Если y равно 0 или 2:

$$\text{pred4x4}_L[x, y] = (\text{p}[x + (y \gg 1), -1] + \text{p}[x + (y \gg 1) + 1, -1] + 1) \gg 1. \quad (8-64)$$

- Иначе (y равно 1 или 3):

$$\text{pred4x4}_L[x, y] = (\text{p}[x + (y \gg 1), -1] + 2 * \text{p}[x + (y \gg 1) + 1, -1] + \text{p}[x + (y \gg 1) + 2, -1] + 2) \gg 2. \quad (8-65)$$

8.3.1.2.9 Описание режима предсказания Intra_4x4_Horizontal_Up

Режим предсказания Intra_4x4 активируется, если Intra4x4PredMode[luma4x4BlkIdx] равно 8.

Этот режим должен использоваться, только если образцы $\text{p}[-1, y]$ с $y = 0..3$ помечены как "доступен для предсказания Intra_4x4".

Допустим, что переменная zHU установлена равной $x + 2 * y$.

Значения образцов предсказания $\text{pred4x4}_L[x, y]$ с $x, y = 0..3$ находят следующим образом.

- Если zHU равно 0, 2 или 4:

$$\text{pred4x4}_L[x, y] = (\text{p}[-1, y + (x \gg 1)] + \text{p}[-1, y + (x \gg 1) + 1] + 1) \gg 1. \quad (8-66)$$

- Иначе, если zHU равно 1 или 3:

$$\text{pred4x4}_L[x, y] = (\text{p}[-1, y + (x \gg 1)] + 2 * \text{p}[-1, y + (x \gg 1) + 1] + \text{p}[-1, y + (x \gg 1) + 2] + 2) \gg 2. \quad (8-67)$$

- Иначе, если zHU равно 5:

$$\text{pred4x4}_L[x, y] = (\text{p}[-1, 2] + 3 * \text{p}[-1, 3] + 2) \gg 2. \quad (8-68)$$

- Иначе (zHU больше 5):

$$\text{pred4x4}_L[x, y] = \text{p}[-1, 3]. \quad (8-69)$$

8.3.2 Процесс предсказания Intra_8x8 для образцов яркости

Этот процесс активируется, если режим предсказания макроблока равен Intra_8x8.

Входами в этот процесс являются значения Intra4x4PredMode (если имеются) или Intra8x8PredMode (если имеются) от смежных макроблоков или пар макроблоков.

Выходы этого процесса – массивы образцов яркости 8x8 как часть массивов яркости 16x16 образцов предсказания макроблока pred_L .

Компонент яркости макроблока состоит из 4 блоков образцов яркости 8x8. Эти блоки – инверсно сканированные, с использованием процесса инверсного сканирования блока яркости 8x8, который описан в п. 6.4.4.

Для всех блоков яркости 8x8 компонента яркости макроблока с $\text{luma8x8BlkIdx} = 0..3$, процесс вывода для Intra8x8PredMode, описанный в 8.3.2.1, активируется с luma8x8BlkIdx , а также Intra4x4PredMode и Intra8x8PredMode, которые ранее (в порядке декодирования) были найдены для смежных макроблоков в качестве входа, и переменной Intra8x8PredMode[luma8x8BlkIdx] в качестве выхода.

Для всех блоков яркости образцов 8x8, проиндексированных с использованием $\text{luma8x8BlkIdx} = 0..3$, применяется следующее.

- Активируется процесс предсказания образца Intra_8x8 согласно п. 8.3.2.2 с luma8x8BlkIdx и созданными образцами до (в порядке декодирования) процесса фильтра устранения блочности от смежных блоков яркости в качестве входа, а выходом являются образцы предсказания яркости $\text{Intra_8x8 pred8x8}_L[x, y]$ с $x, y = 0..7$.
- Положение верхнего левого образца блока яркости 8x8 с индексом luma8x8BlkIdx внутри текущего макроблока находят, активируя процесс инверсного сканирования блока яркости 8x8 согласно 6.4.4 с luma8x8BlkIdx в качестве входа и выходом, которому присваивается (xO, yO) с $x, y = 0..7$.

$$\text{pred}_L[xO + x, yO + y] = \text{pred8x8}_L[x, y]. \quad (8-70)$$

- Процесс декодирования коэффициентов преобразования и процесс создания изображения до процесса фильтра устранения блочности согласно п. 8.5 активируют с pred_L и luma8x8BlkIdx в качестве входа и созданными образцами текущего блока яркости 8x8 $\text{luma block } S'_L$ в качестве выхода.

8.3.2.1 Процесс вывода Intra8x8PredMode

Входами в этот процесс являются индекс блока яркости 8x8 luma8x8BlkIdx и массивы переменных Intra4x4PredMode (если доступен) и Intra8x8PredMode (если доступен), которые предварительно (в порядке декодирования) находят для смежных макроблоков.

Выход этого процесса – переменная $\text{Intra8x8PredMode}[\text{luma8x8BlkIdx}]$.

В таблице 8-3 показаны значения $\text{Intra8x8PredMode}[\text{luma8x8BlkIdx}]$ и связанные с ними мнемонические имена.

Таблица 8-3 – Спецификация $\text{Intra8x8PredMode}[\text{luma8x8BlkIdx}]$ и связанных имен

$\text{Intra8x8PredMode}[\text{luma8x8BlkIdx}]$	Имя $\text{Intra8x8PredMode}[\text{luma8x8BlkIdx}]$
0	$\text{Intra_8x8_Vertical}$ (режим предсказания)
1	$\text{Intra_8x8_Horizontal}$ (режим предсказания)
2	Intra_8x8_DC (режим предсказания)
3	$\text{Intra_8x8_Diagonal_Down_Left}$ (режим предсказания)
4	$\text{Intra_8x8_Diagonal_Down_Right}$ (режим предсказания)
5	$\text{Intra_8x8_Vertical_Right}$ (режим предсказания)
6	$\text{Intra_8x8_Horizontal_Down}$ (режим предсказания)
7	$\text{Intra_8x8_Vertical_Left}$ (режим предсказания)
8	$\text{Intra_8x8_Horizontal_Up}$ (режим предсказания)

$\text{Intra8x8PredMode}[\text{luma8x8BlkIdx}]$ находят следующим образом.

- Процесс, определенный в п. 6.4.8.2, активируется с luma8x8BlkIdx , заданным как вход, а выход присваивается значениям mbAddrA , luma8x8BlkIdxA , mbAddrB и luma8x8BlkIdxB .
- Переменную $\text{dcPredModePredictedFlag}$ находят следующим образом.
 - Если любое из следующих условий – истина, то $\text{dcPredModePredictedFlag}$ устанавливают равным 1:
 - макроблок с адресом mbAddrA не доступен;
 - макроблок с адресом mbAddrB не доступен;
 - макроблок с адресом mbAddrA доступен и кодирован в режиме Inter предсказания, и $\text{constrained_intra_pred_flag}$ равно 1;
 - макроблок с адресом mbAddrB доступен и кодирован в режиме Inter предсказания, и $\text{constrained_intra_pred_flag}$ равно 1.
 - Иначе $\text{dcPredModePredictedFlag}$ устанавливается равным 0.

- Для значения N, которое заменяют либо значением A, либо B, переменные ntraMxMPredModeN находят следующим образом.
 - Если dcPredModePredictedFlag равно 1 или (макроблок с адресом mbAddrN кодирован не в режиме предсказания макроблока Intra_4x4 и макроблок с адресом mbAddrN кодирован не в режиме предсказания макроблока Intra_8x8), то intraMxMPredModeN устанавливаются равным 2 (режим предсказания Intra_8x8_DC).
 - Иначе (dcPredModePredictedFlag равно 0 и (макроблок с адресом mbAddrN кодирован в режиме предсказания макроблока Intra_4x4 или макроблок с адресом mbAddrN кодирован в режиме предсказания макроблока Intra_8x8)) применяется следующее:
 - если макроблок с адресом mbAddrN кодирован в режиме макроблока Intra_8x8, intraMxMPredModeN устанавливается равным Intra8x8PredMode[luma8x8BlkIdxN], где Intra8x8PredMode – массив переменных, присвоенных макроблоку mbAddrN;
 - иначе (макроблок с адресом mbAddrN кодирован в режиме макроблока Intra_4x4) intraMxMPredModeN определяется с помощью следующей процедуры, где Intra4x4PredMode – массив переменных, присвоенных макроблоку mbAddrN.

$$\text{intraMxMPredModeN} = \text{Intra4x4PredMode}[\text{luma8x8BlkIdxN} * 4 + n], \quad (8-71)$$

где переменная n определяется следующим образом:

- если N равно A, в зависимости от переменной MbaffFrameFlag, переменной luma8x8BlkIdx, текущего макроблока и макроблока mbAddrN применяется следующее:
 - если MbaffFrameFlag равно 1, текущий макроблок – это макроблок, кодированный кадрами, макроблок mbAddrN – это макроблок, кодированный полями, и luma8x8BlkIdx равно 2, n устанавливается равным 3;
 - иначе (MbaffFrameFlag равно 0 или текущий макроблок – это макроблок, кодированный полями, или макроблок mbAddrN – это макроблок, кодированный кадрами, или luma8x8BlkIdx не равно 2) n устанавливается равным 1;
 - иначе (N равно B) n устанавливается равным 2.
- Наконец, при заданных intraMxMPredModeA и intraMxMPredModeB переменную Intra8x8PredMode[luma8x8BlkIdx] находят, применяя следующую процедуру:

```

predIntra8x8PredMode = Min( intraMxMPredModeA, intraMxMPredModeB )
if( prev_intra8x8_pred_mode_flag[ luma8x8BlkIdx ] )
  Intra8x8PredMode[ luma8x8BlkIdx ] = predIntra8x8PredMode
else
  if( rem_intra8x8_pred_mode[ luma8x8BlkIdx ] < predIntra8x8PredMode )
    Intra8x8PredMode[ luma8x8BlkIdx ] = rem_intra8x8_pred_mode[ luma8x8BlkIdx ]
  else
    Intra8x8PredMode[ luma8x8BlkIdx ] = rem_intra8x8_pred_mode[ luma8x8BlkIdx ] + 1.
  
```

(8-72)

8.3.2.2 Предсказание образца Intra_8x8

Этот процесс активируется для каждого блока яркости 8x8 макроблока с режимом предсказания, равным Intra_8x8, за которым следует процесс преобразования декодирования и процесс создания изображения до устранения блочности для каждого блока яркости 8x8.

Входом в этот процесс является индекс блока яркости 8x8 luma8x8BlkIdx.

Выход этого процесса – образцы предсказания pred8x8L[x, y] с x, y = 0..7 для блока яркости 8x8 с индексом luma8x8BlkIdx.

Положение верхнего левого образца блока яркости 8x8 с индексом luma8x8BlkIdx внутри текущего макроблока находят, активируя процесс инверсного сканирования блока яркости 8x8 согласно п. 6.4.4 с luma8x8BlkIdx в качестве входа и выходом, которому присваивается (xO, yO).

25 соседних образцов p[x, y] которые являются созданными образцами яркости до процесса фильтра устранения блочности с x = -1, y = -1..7 и x = 0..15, y = -1, находят следующим образом.

- Местоположение яркости (xN, yN) определяется как:

$$xN = xO + x; \quad (8-73)$$

$$yN = yO + y. \quad (8-74)$$

- Процесс вывода соседних местоположений согласно п. 6.4.9 активируют для местоположений яркости с (xN, yN) в качестве входа и $mbAddrN$ и (xW, yW) в качестве выхода.
- Каждый образец $p[x, y]$ с $x = -1, y = -1..7$ и $x = 0..15, y = -1$ находят следующим образом.
 - Если любое из следующих условий – истина, то образец $p[x, y]$ помечен как "не доступен для предсказания Intra_8x8":
 - $mbAddrN$ не доступно;
 - макроблок $mbAddrN$ кодирован в режиме Inter предсказания, и `constrained_intra_pred_flag` равно 1.
 - Иначе образец $p[x, y]$ помечен как "доступен для предсказания Intra_8x8", а образцу яркости в местоположении яркости (xW, yW) внутри макроблока $mbAddrN$ присваивается $p[x, y]$.

Если образцы $p[x, -1]$ с $x = 8..15$ помечены как "не доступен для предсказания Intra_8x8", а образец $p[7, -1]$ помечен как "доступен для предсказания Intra_8x8", то значение образца $p[7, -1]$ заменяет значения образца $p[x, -1]$ с $x = 8..15$, а образцы $p[x, -1]$ с $x = 8..15$ помечают как "доступен для предсказания Intra_8x8".

ПРИМЕЧАНИЕ. – Считают, что каждый блок создан в массиве изображений до декодирования следующего блока.

Процесс фильтрации контрольных образцов для предсказания образцов Intra_8x8 согласно п. 8.3.2.2.1 активируется с образцами $p[x, y]$ с $x = -1, y = -1..7$ и $x = 0..15, y = -1$ (если имеются) в качестве входа и $p'[x, y]$ с $x = -1, y = -1..7$ и $x = 0..15, y = -1$ в качестве выхода.

В зависимости от `Intra8x8PredMode[luma8x8BklIdx]` активируется один из режимов предсказания Intra_8x8, определенных в пп.8.3.2.2.2–8.3.2.2.10.

8.3.2.2.1 Процесс фильтрации контрольных образцов для предсказания образцов Intra_8x8

Входами в этот процесс являются контрольные образцы $p[x, y]$ с $x = -1, y = -1..7$ и $x = 0..15, y = -1$ (если имеются) для предсказания образцов Intra_8x8.

Выходами этого процесса являются фильтрованные контрольные образцы $p'[x, y]$ с $x = -1, y = -1..7$ и $x = 0..15, y = -1$ для предсказания образцов Intra_8x8.

Если все образцы $p[x, -1]$ с $x = 0..7$ помечены как "доступен для предсказания Intra_8x8", применяется следующее.

- Значение $p'[0, -1]$ определяется следующим образом.
 - Если $p[-1, -1]$ помечен как "доступен для предсказания Intra_8x8", $p'[0, -1]$ определяется как:

$$p'[0, -1] = (p[-1, -1] + 2 * p[0, -1] + p[1, -1] + 2) >> 2. \quad (8-75)$$

- Иначе ($p[-1, -1]$ помечен как "не доступен для предсказания Intra_8x8") $p'[0, -1]$ определяется как:

$$p'[0, -1] = (3 * p[0, -1] + p[1, -1] + 2) >> 2. \quad (8-76)$$

- Значения $p'[x, -1]$ с $x = 1..7$ определяют как:

$$p'[x, -1] = (p[x-1, -1] + 2 * p[x, -1] + p[x+1, -1] + 2) >> 2. \quad (8-77)$$

Если все образцы $p[x, -1]$ с $x = 7..15$ помечены как "доступен для предсказания Intra_8x8", применяется следующее.

- Значения $p'[x, -1]$ с $x = 8..14$ определяются как:

$$p'[x, -1] = (p[x-1, -1] + 2 * p[x, -1] + p[x+1, -1] + 2) >> 2. \quad (8-78)$$

- Значение $p'[15, -1]$ определяется как:

$$p'[15, -1] = (p[14, -1] + 3 * p[15, -1] + 2) >> 2. \quad (8-79)$$

Если все образцы $p[-1, -1]$ помечены как "доступен для предсказания Intra_8x8", значение $p'[-1, -1]$ определяется следующим образом.

- Если образец $p[0, -1]$ помечен как "не доступен для предсказания Intra_8x8" или образец $p[-1, 0]$ помечен как "не доступен для предсказания Intra_8x8", применяется следующее.

- Если образец $p[0, -1]$ помечен как "доступен для предсказания Intra_8x8", $p'[-1, -1]$ определяется как:

$$p'[-1, -1] = (3 * p[-1, -1] + p[0, -1] + 2) \gg 2. \quad (8-80)$$

- Иначе (образец $p[0, -1]$ помечен как "не доступен для предсказания Intra_8x8", а образец $p[-1, 0]$ помечен как "доступен для предсказания Intra_8x8") $p'[-1, -1]$ определяется как:

$$p'[-1, -1] = (3 * p[-1, -1] + p[-1, 0] + 2) \gg 2. \quad (8-81)$$

- Иначе (образец $p[0, -1]$ помечен как "доступен для предсказания Intra_8x8" и образец $p[-1, 0]$ помечен как "доступен для предсказания Intra_8x8") $p'[-1, -1]$ определяется как:

$$p'[-1, -1] = (p[0, -1] + 2 * p[-1, -1] + p[-1, 0] + 2) \gg 2. \quad (8-82)$$

Если все образцы $p[-1, y]$ с $y = 0..7$ помечены как "доступен для предсказания Intra_8x8", применяется следующее.

- Значение $p'[-1, 0]$ определяется следующим образом.

- Если $p[-1, -1]$ помечен как "доступен для предсказания Intra_8x8", $p'[-1, 0]$ определяется как:

$$p'[-1, 0] = (p[-1, -1] + 2 * p[-1, 0] + p[-1, 1] + 2) \gg 2. \quad (8-83)$$

- Иначе (if $p[-1, -1]$ помечен как "не доступен для предсказания Intra_8x8") $p'[-1, 0]$ определяется как:

$$p'[-1, 0] = (3 * p[-1, 0] + p[-1, 1] + 2) \gg 2. \quad (8-84)$$

- Значения $p'[-1, y]$ с $y = 1..6$ определяются как:

$$p'[-1, y] = (p[-1, y-1] + 2 * p[-1, y] + p[-1, y+1] + 2) \gg 2. \quad (8-85)$$

- Значение $p'[-1, 7]$ определяется как:

$$p'[-1, 7] = (p[-1, 6] + 3 * p[-1, 7] + 2) \gg 2. \quad (8-86)$$

8.3.2.2.2 Описание режима предсказания Intra_8x8_Vertical

Режим предсказания Intra_8x8 активируется, если Intra8x8PredMode[luma8x8BlkIdx] равно 0.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..7$ помечены как "доступен для предсказания Intra_8x8".

Значения образцов предсказания $pred8x8_L[x, y]$ с $x, y = 0..7$ находят следующим образом:

$$pred8x8_L[x, y] = p'[x, -1], \text{ с } x, y = 0..7. \quad (8-87)$$

8.3.2.2.3 Описание режима предсказания Intra_8x8_Horizontal

Режим предсказания Intra_8x8 активируется, если Intra8x8PredMode[luma8x8BlkIdx] равно 1.

Этот режим должен использоваться, только если образцы $p[-1, y]$ с $y = 0..7$ помечены как "доступен для предсказания Intra_8x8".

Значения образцов предсказания $pred8x8_L[x, y]$ с $x, y = 0..7$ находят следующим образом:

$$pred8x8_L[x, y] = p'[-1, y] \text{ с } x, y = 0..7. \quad (8-88)$$

8.3.2.2.4 Описание режима предсказания Intra_8x8_DC

Режим предсказания Intra_8x8 активируется, если Intra8x8PredMode[luma8x8BlkIdx] равно 2.

Значения образцов предсказания pred8x8L[x, y] с x, y = 0..7 находят следующим образом.

- Если все образцы p[x, -1] с x = 0..7 и p[-1, y] с y = 0..7 помечены как "доступен для предсказания Intra_8x8", то значения образцов предсказания pred8x8L[x, y] с x, y = 0..7 находят следующим образом:

$$\text{pred8x8L}[x, y] = \left(\sum_{x'=0}^7 p'[x', -1] + \sum_{y'=0}^7 p'[-1, y'] + 8 \right) \gg 4. \quad (8-89)$$

- Иначе, если какие-либо образцы p[x, -1] с x = 0..7 помечены как "не доступен для предсказания Intra_8x8", а все образцы p[-1, y] с y = 0..7 помечены как "доступен для предсказания Intra_8x8", значения образцов предсказания pred8x8L[x, y] с x, y = 0..7 находят следующим образом:

$$\text{pred8x8L}[x, y] = \left(\sum_{y'=0}^7 p'[-1, y'] + 4 \right) \gg 3. \quad (8-90)$$

- Иначе, если какие-либо образцы p[-1, y] с y = 0..7 помечены как "не доступен для предсказания Intra_8x8", а все образцы p[x, -1] с x = 0..7 помечены как "доступен для предсказания Intra_4x4", значения образцов предсказания pred8x8L[x, y] с x, y = 0..7 находят следующим образом:

$$\text{pred8x8L}[x, y] = \left(\sum_{x'=0}^7 p'[x', -1] + 4 \right) \gg 3. \quad (8-91)$$

- Иначе (некоторые образцы p[x, -1], с x = 0..7 и некоторые образцы p[-1, y] с y = 0..7 помечены как "не доступен для предсказания Intra_8x8") значения образцов предсказания pred8x8L[x, y] с x, y = 0..7 находят следующим образом:

$$\text{pred8x8L}[x, y] = (1 \ll (\text{BitDepth}_Y - 1)). \quad (8-92)$$

ПРИМЕЧАНИЕ. – Используя этот режим всегда возможно предсказать блок яркости 8x8.

8.3.2.2.5 Описание режима предсказания Intra_8x8_Diagonal_Down_Left

Режим предсказания Intra_8x8 активируется, если Intra8x8PredMode[luma8x8BlkIdx] равно 3.

Этот режим должен использоваться, только если образцы p[x, -1] с x = 0..15 помечены как "доступен для предсказания Intra_8x8".

Значения образцов предсказания pred8x8L[x, y], с x, y = 0..7 находят следующим образом.

- Если x равно 7 и y равно 7:

$$\text{pred8x8L}[x, y] = (p'[14, -1] + 3 * p'[15, -1] + 2) \gg 2. \quad (8-93)$$

- Иначе (x не равно 7 или y не равно 7):

$$\text{pred8x8L}[x, y] = (p'[x + y, -1] + 2 * p'[x + y + 1, -1] + p'[x + y + 2, -1] + 2) \gg 2. \quad (8-94)$$

8.3.2.2.6 Описание режима предсказания Intra_8x8_Diagonal_Down_Right

Режим предсказания Intra_8x8 активируется, если Intra8x8PredMode[luma8x8BlkIdx] равно 4.

Этот режим должен использоваться, только если образцы p[x, -1] с x = 0..7 и p[-1, y] с y = -1..7 помечены как "доступен для предсказания Intra_8x8".

Значения образцов предсказания pred8x8L[x, y] с x, y = 0..7 находят следующим образом.

- Если x больше y:

$$\text{pred8x8L}[x, y] = (p'[x - y - 2, -1] + 2 * p'[x - y - 1, -1] + p'[x - y, -1] + 2) \gg 2. \quad (8-95)$$

- Иначе, если x меньше y :

$$\text{pred8x8}_L[x, y] = (p'[-1, y - x - 2] + 2 * p'[-1, y - x - 1] + p'[-1, y - x] + 2) \gg 2. \quad (8-96)$$

- Иначе (x равно y):

$$\text{pred8x8}_L[x, y] = (p'[0, -1] + 2 * p'[-1, -1] + p'[-1, 0] + 2) \gg 2. \quad (8-97)$$

8.3.2.2.7 Описание режима предсказания Intra_8x8_Vertical_Right

Режим предсказания Intra_8x8 активируется, если Intra8x8PredMode[luma8x8BlkIdx] равно 5.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..7$ и $p[-1, y]$ с $y = -1..7$ помечены как "доступен для предсказания Intra_8x8".

Допустим, что переменная zVR установлена равной $2 * x - y$.

Значения образцов предсказания $\text{pred8x8}_L[x, y]$ с $x, y = 0..7$ находят следующим образом.

- Если zVR равно 0, 2, 4, 6, 8, 10, 12 или 14:

$$\text{pred8x8}_L[x, y] = (p'[x - (y \gg 1) - 1, -1] + p'[x - (y \gg 1), -1] + 1) \gg 1. \quad (8-98)$$

- Иначе, если zVR равно 1, 3, 5, 7, 9, 11 или 13:

$$\text{pred8x8}_L[x, y] = (p'[x - (y \gg 1) - 2, -1] + 2 * p'[x - (y \gg 1) - 1, -1] + p'[x - (y \gg 1), -1] + 2) \gg 2. \quad (8-99)$$

- Иначе, если zVR равно -1:

$$\text{pred8x8}_L[x, y] = (p'[-1, 0] + 2 * p'[-1, -1] + p'[0, -1] + 2) \gg 2. \quad (8-100)$$

- Иначе (zVR равно -2, -3, -4, -5, -6 или -7):

$$\text{pred8x8}_L[x, y] = (p'[-1, y - 2*x - 1] + 2 * p'[-1, y - 2*x - 2] + p'[-1, y - 2*x - 3] + 2) \gg 2. \quad (8-101)$$

8.3.2.2.8 Описание режима предсказания Intra_8x8_Horizontal_Down

Режим предсказания Intra_8x8 активируется, если Intra8x8PredMode[luma8x8BlkIdx] равно 6.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..7$ и $p[-1, y]$ с $y = -1..7$ помечены как "доступен для предсказания Intra_8x8".

Допустим, что переменная zHD установлена равной $2 * y - x$.

Значения образцов предсказания $\text{pred8x8}_L[x, y]$ с $x, y = 0..7$ находят следующим образом.

- Если zHD равно 0, 2, 4, 6, 8, 10, 12 или 14:

$$\text{pred8x8}_L[x, y] = (p'[-1, y - (x \gg 1) - 1] + p'[-1, y - (x \gg 1)] + 1) \gg 1. \quad (8-102)$$

- Иначе, если zHD равно 1, 3, 5, 7, 9, 11 или 13:

$$\text{pred8x8}_L[x, y] = (p'[-1, y - (x \gg 1) - 2] + 2 * p'[-1, y - (x \gg 1) - 1] + p'[-1, y - (x \gg 1)] + 2) \gg 2. \quad (8-103)$$

- Иначе, если zHD равно -1:

$$\text{pred8x8}_L[x, y] = (p'[-1, 0] + 2 * p'[-1, -1] + p'[0, -1] + 2) \gg 2. \quad (8-104)$$

- Иначе (zHD равно -2, -3, -4, -5, -6, -7):

$$\text{pred8x8}_L[x, y] = (p'[x - 2*y - 1, -1] + 2 * p'[x - 2*y - 2, -1] + p'[x - 2*y - 3, -1] + 2) \gg 2. \quad (8-105)$$

8.3.2.2.9 Описание режима предсказания Intra_8x8_Vertical_Left

Режим предсказания Intra_8x8 активируется, если Intra8x8PredMode[luma8x8BlkIdx] равно 7.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..15$ помечены как "доступен для предсказания Intra_8x8".

Значения образцов предсказания $pred8x8_L[x, y]$ с $x, y = 0..7$ находят следующим образом.

- Если y равно 0, 2, 4 или 6:

$$pred8x8_L[x, y] = (p'[x + (y >> 1), -1] + p'[x + (y >> 1) + 1, -1] + 1) >> 1. \quad (8-106)$$

- Иначе (y равно 1, 3, 5, 7):

$$pred8x8_L[x, y] = (p'[x + (y >> 1), -1] + 2 * p'[x + (y >> 1) + 1, -1] + p'[x + (y >> 1) + 2, -1] + 2) >> 2. \quad (8-107)$$

8.3.2.2.10 Описание режима предсказания Intra_8x8_Horizontal_Up

Режим предсказания Intra_8x8 активируется, если Intra8x8PredMode[luma8x8BlkIdx] равно 8.

Этот режим должен использоваться, только если образцы $p[-1, y]$ с $y = 0..7$ помечены как "доступен для предсказания Intra_8x8".

Допустим, что переменная zHU установлена равной $x + 2 * y$.

Значения образцов предсказания $pred8x8_L[x, y]$ с $x, y = 0..7$ находят следующим образом.

- Если zHU равно 0, 2, 4, 6, 8, 10 или 12:

$$pred8x8_L[x, y] = (p'[-1, y + (x >> 1)] + p'[-1, y + (x >> 1) + 1] + 1) >> 1. \quad (8-108)$$

- Иначе, если zHU равно 1, 3, 5, 7, 9 или 11:

$$pred8x8_L[x, y] = (p'[-1, y + (x >> 1)] + 2 * p'[-1, y + (x >> 1) + 1] + p'[-1, y + (x >> 1) + 2] + 2) >> 2. \quad (8-109)$$

- Иначе, если zHU равно 13:

$$pred8x8_L[x, y] = (p'[-1, 6] + 3 * p'[-1, 7] + 2) >> 2. \quad (8-110)$$

- Иначе (zHU больше 13):

$$pred8x8_L[x, y] = p'[-1, 7]. \quad (8-111)$$

8.3.3 Процесс предсказания для образцов яркости Intra_16x16

Процесс активируют, если режим предсказания макроблока равен Intra_16x16. Этот режим определяет, как находят образцы яркости текущего макроблока при Intra предсказании.

Выходы этого процесса – Intra предсказания образцов яркости для текущего макроблока $pred_L[x, y]$.

33 смежных образца $p[x, y]$, которые создают образцы яркости до процесса фильтра устранения блочности с $x = -1, y = -1..15$ и с $x = 0..15, y = -1$, находят следующим образом.

- Активируют процесс вывода смежных местоположений согласно п. 6.4.9 для местоположений яркости с (x, y) , присвоенным значению (xN, yN) как входу, а $mbAddrN$ и (xW, yW) – в качестве выхода.
- Каждый образец $p[x, y]$ с $x = -1, y = -1..15$ с $x = 0..15, y = -1$ находят следующим образом.
 - Если любое из следующих условий – истина, образец $p[x, y]$ помечен как "не доступен для предсказания Intra_16x16":
 - $mbAddrN$ не доступен;
 - макроблок $mbAddrN$ кодирован в режиме Inter предсказания и ограничен $constrained_intra_pred_flag$, равным 1;
 - макроблок $mbAddrN$ имеет mb_type , равное SI, и ограничен $constrained_intra_pred_flag$, равным 1.

- Иначе образец $p[x, y]$ помечен как "доступен для предсказания Intra_16x16", а образцу яркости в местоположении яркости (xW, yW) внутри макроблока mbAddrN присвоено значение $p[x, y]$.

Предположим, что $\text{predL}[x, y]$ с $x, y = 0..15$ указывает на образцы предсказания для блока образцов яркости 16x16.

Режим предсказания Intra_16x16 определен в таблице 8-3.

Таблица 8-4 – Спецификация Intra16x16PredMode и связанные с этим имена

Intra16x16PredMode	Имя Intra16x16PredMode
0	Intra_16x16_Vertical (режим предсказания)
1	Intra_16x16_Horizontal (режим предсказания)
2	Intra_16x16_DC (режим предсказания)
3	Intra_16x16_Plane (режим предсказания)

В зависимости от Intra16x16PredMode активируется один из режимов предсказания Intra_16x16, определенный в пп. 8.3.3.1–8.3.3.4.

8.3.3.1 Описание режима предсказания Intra_16x16_Vertical

Этот режим предсказания Intra_16x16 должен использоваться, если образцы $p[x, -1]$ с $x = 0..15$ помечены как "доступен для предсказания Intra_16x16".

$$\text{predL}[x, y] = p[x, -1] \text{ с } x, y = 0..15. \quad (8-112)$$

8.3.3.2 Описание режима предсказания Intra_16x16_Horizontal

Этот режим предсказания Intra_16x16 включается, только если образцы $p[-1, y]$ с $y = 0..15$ помечены как "доступен для предсказания Intra_16x16".

$$\text{predL}[x, y] = p[-1, y] \text{ с } x, y = 0..15. \quad (8-113)$$

8.3.3.3 Описание режима предсказания Intra_16x16_DC

Этот режим предсказания Intra_16x16 включается в зависимости от того, помечены ли смежные образцы как "доступен для предсказания Intra_16x16", следующим образом.

- Если все соседние образцы $p[x, -1]$ с $x = 0..15$ и $p[-1, y]$ с $y = 0..15$ помечены как "доступен для предсказания Intra_16x16", предсказания для всех образцов яркости в макроблоке заданы уравнением:

$$\text{predL}[x, y] = \left(\sum_{x'=0}^{15} p[x', -1] + \sum_{y'=0}^{15} p[-1, y'] + 16 \right) \gg 5 \text{ с } x, y = 0..15. \quad (8-114)$$

- Иначе, если какие-либо соседние образцы $p[x, -1]$ с $x = 0..15$ помечены как "не доступен для предсказания Intra_16x16", а все смежные образцы $p[-1, y]$ с $y = 0..15$ помечены как "доступен для предсказания Intra_16x16", то предсказания для всех образцов яркости в макроблоке заданы уравнением:

$$\text{predL}[x, y] = \left(\sum_{y'=0}^{15} p[-1, y'] + 8 \right) \gg 4 \text{ с } x, y = 0..15. \quad (8-115)$$

- Иначе, если какие-либо соседние образцы $p[-1, y]$ с $y = 0..15$ помечены как "не доступен для предсказания Intra_16x16", а все соседние образцы $p[x, -1]$ с $x = 0..15$ помечены как "доступен для предсказания Intra_16x16", предсказания для всех образцов яркости в макроблоке заданы уравнением:

$$\text{predL}[x, y] = \left(\sum_{x'=0}^{15} p[x', -1] + 8 \right) \gg 4 \text{ с } x, y = 0..15. \quad (8-116)$$

- Иначе (некоторые из соседних образцов $p[x, -1]$ с $x = 0..15$ и некоторые из соседних образцов $p[-1, y]$ с $y = 0..15$ помечены как "не доступен для предсказания Intra_16x16") предсказания для всех образцов яркости в макроблоке заданы уравнением:

$$\text{predL}[x, y] = (1 \ll (\text{BitDepthY} - 1)) \text{ с } x, y = 0..15. \quad (8-117)$$

8.3.3.4 Описание режима предсказания Intra_16x16_Plane

Этот режим предсказания Intra_16x16 должен активироваться только в том случае, если образцы $p[x, -1]$ с $x = -1..15$ и $p[-1, y]$ с $y = 0..15$ помечены как "доступен для предсказания Intra_16x16".

$$\text{predL}[x, y] = \text{Clip1}((a + b * (x - 7) + c * (y - 7) + 16) \gg 5) \text{ с } x, y = 0..15, \quad (8-118)$$

где:

$$a = 16 * (p[-1, 15] + p[15, -1]); \quad (8-119)$$

$$b = (5 * H + 32) \gg 6; \quad (8-120)$$

$$c = (5 * V + 32) \gg 6, \quad (8-121)$$

а H и V определены в уравнениях 8-122 и 8-123:

$$H = \sum_{x'=0}^7 (x'+1) * (p[8+x', -1] - p[6-x', -1]); \quad (8-122)$$

$$V = \sum_{y'=0}^7 (y'+1) * (p[-1, 4+y'] - p[-1, 2-y']). \quad (8-123)$$

8.3.4 Процесс Intra предсказания для образцов цветности

Этот процесс активируют для типов макроблоков I и SI. Процесс определяет, как находят Intra предсказания для образцов цветности текущего макроблока.

Выходы этого процесса – образцы цветности Intra предсказания для текущего макроблока $\text{pred}_{Cb}[x, y]$ и $\text{pred}_{Cr}[x, y]$.

Оба блока цветности (Cb и Cr) макроблока используют тот же режим предсказания. Режим предсказания применяют к каждому из блоков цветности по отдельности. Процесс, определенный в этом пункте, активируют для каждого блока цветности. В оставшейся части этого пункта за блок цветности принимают один из двух блоков цветности, а индекс C используют как замену индексов Cb или Cr.

Соседние образцы $p[x, y]$, которые создают образцы цветности до процесса фильтра устранения блочности с $x = -1$, $y = -1..MbHeightC - 1$ и с $x = 0..MbWidthC - 1$, $y = -1$, находят следующим образом.

- Процесс вывода для смежных местоположений согласно п. 6.4.9 активируют для местоположений цветности со значением (x, y) , присвоенным (xN, yN) как входу, а mbAddrN и (xW, yW) – в качестве выхода.
- Каждый образец $p[x, y]$ находят следующим образом.
 - Если любое из следующих условий – истина, то образец $p[x, y]$ помечен как "не доступен для Intra предсказания цветности":
 - mbAddrN не доступно;
 - макроблок mbAddrN кодирован в режиме Inter предсказания с флагом `constrained_intra_pred_flag`, равным 1;
 - макроблок mbAddrN имеет `mb_type`, равное SI, с флагом `constrained_intra_pred_flag`, равным 1, а текущий макроблок не имеет `mb_type`, равное SI.
 - Иначе образец $p[x, y]$ помечен как "доступен для Intra предсказания цветности", а образцу цветности компонента C в местоположении цветности (xW, yW) внутри макроблока mbAddrN присвоено значение $p[x, y]$.

Положим, что $pred_c[x, y]$ с $x = 0..MbWidthC - 1$, $y = 0..MbHeightC - 1$ обозначает образцы предсказания для блока образцов цветности.

Режимы Intra предсказания цветности приведены в таблице 8-5.

Таблица 8-5 – Спецификация режима Intra предсказания цветности и связанные с этим имена

intra_chroma_pred_mode	Имя intra_chroma_pred_mode
0	Intra_Chroma_DC (режим предсказания)
1	Intra_Chroma_Horizontal (режим предсказания)
2	Intra_Chroma_Vertical (режим предсказания)
3	Intra_Chroma_Plane (режим предсказания)

В зависимости от `intra_chroma_pred_mode` активируется один из режимов Intra предсказания цветности, определенный в пп. 8.3.4.1–8.3.4.4.

8.3.4.1 Описание режима предсказания Intra_Chroma_DC

Режим Intra предсказания цветности активируется, если `intra_chroma_pred_mode` равно 0.

Для всех блоков цветности образцов 4x4, индексированных с `chroma4x4BlkIdx = 0..(1 << (chroma_format_idc + 1)) - 1`, применяется следующее.

- В зависимости от `chroma_format_idc` местоположение верхнего левого образца блока цветности 4x4 с индексом `chroma4x4BlkIdx` определяется следующим образом.

- Если `chroma_format_idc` равно 1 или 2, применяется следующее:

$$xO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 0); \quad (8-124)$$

$$yO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 1). \quad (8-125)$$

- Иначе (`chroma_format_idc` равно 3) применяется следующее:

$$xO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 0) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 0); \quad (8-126)$$

$$yO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 1) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 1). \quad (8-127)$$

- Если (`xO, yO`) равно (0, 0) или `xO` и `yO` больше 0, значения образцов предсказания $pred_c[x + xO, y + yO]$ с $x, y = 0..3$ определяются следующим образом.

- Если все образцы $p[x + xO, -1]$ с $x = 0..3$ и $p[-1, y + yO]$ с $y = 0..3$ помечены как "доступен для Intra предсказания цветности", значения образцов предсказания $pred_c[x + xO, y + yO]$ с $x, y = 0..3$ находят следующим образом:

$$pred_c[x + xO, y + yO] = \left(\sum_{x'=0}^3 p[x'+xO, -1] + \sum_{y'=0}^3 p[-1, y'+yO] + 4 \right) >> 3 \text{ с } x, y = 0..3. \quad (8-128)$$

- Иначе, если какие-либо образцы $p[x + xO, -1]$ с $x = 0..3$ помечены как "не доступен для Intra предсказания цветности", а все образцы $p[-1, y + yO]$ с $y = 0..3$ помечены как "доступен для Intra предсказания цветности", значения образцов предсказания $pred_c[x + xO, y + yO]$ с $x, y = 0..3$ находят следующим образом:

$$pred_c[x + xO, y + yO] = \left(\sum_{y'=0}^3 p[-1, y'+yO] + 2 \right) >> 2 \text{ с } x, y = 0..3. \quad (8-129)$$

- Иначе, если какие-либо образцы $p[-1, y+yO]$ с $y=0..3$ помечены как "не доступен для Intra предсказания цветности", а все образцы $p[x+xO, -1]$ с $x=0..3$ помечены как "доступен для Intra предсказания цветности", значения образцов предсказания $pred_C[x+xO, y+yO]$ с $x, y=0..3$ находят следующим образом:

$$pred_C[x+xO, y+yO] = \left(\sum_{x'=0}^3 p[x'+xO, -1] + 2 \right) \gg 2 \text{ с } x, y = 0..3. \quad (8-130)$$

- Иначе (некоторые образцы $p[x+xO, -1]$ с $x=0..3$ и некоторые образцы $p[-1, y+yO]$ с $y=0..3$ помечены как "не доступен для Intra предсказания цветности") значения образцов предсказания $pred_C[x+xO, y+yO]$ с $x, y=0..3$ находят следующим образом:

$$pred_C[x+xO, y+yO] = (1 \ll (\text{BitDepth}_C - 1)) \text{ с } x, y = 0..3. \quad (8-131)$$

- Иначе, если xO больше 0, а yO равно 0, значения образцов предсказания $pred_C[x+xO, y+yO]$ с $x, y=0..3$ находят следующим образом.

- Если все образцы $p[x+xO, -1]$ с $x=0..3$ помечены как "доступен для Intra предсказания цветности" значения образцов предсказания $pred_C[x+xO, y+yO]$ с $x, y=0..3$ находят следующим образом:

$$pred_C[x+xO, y+yO] = \left(\sum_{x'=0}^3 p[x'+xO, -1] + 2 \right) \gg 2 \text{ с } x, y = 0..3. \quad (8-132)$$

- Иначе, если все образцы $p[-1, y+yO]$ с $y=0..3$ помечены как "доступен для Intra предсказания цветности" значения образцов предсказания $pred_C[x+xO, y+yO]$ с $x, y=0..3$ находят следующим образом:

$$pred_C[x+xO, y+yO] = \left(\sum_{y'=0}^3 p[-1, y'+yO] + 2 \right) \gg 2 \text{ с } x, y = 0..3. \quad (8-133)$$

- Иначе (некоторые образцы $p[x+xO, -1]$ с $x=0..3$ и некоторые образцы $p[-1, y+yO]$ с $y=0..3$ помечены как "не доступен для Intra предсказания цветности") значения образцов предсказания $pred_C[x+xO, y+yO]$ с $x, y=0..3$ находят следующим образом:

$$pred_C[x+xO, y+yO] = (1 \ll (\text{BitDepth}_C - 1)) \text{ с } x, y = 0..3. \quad (8-134)$$

- Иначе (xO равно 0, а yO больше 0) значения образцов предсказания $pred_C[x+xO, y+yO]$ с $x, y=0..3$ находят следующим образом

- Если все образцы $p[-1, y+yO]$ с $y=0..3$ помечены как "доступен для Intra предсказания цветности" значения образцов предсказания $pred_C[x+xO, y+yO]$ с $x, y=0..3$ находят следующим образом:

$$pred_C[x+xO, y+yO] = \left(\sum_{y'=0}^3 p[-1, y'+yO] + 2 \right) \gg 2 \text{ с } x, y = 0..3. \quad (8-135)$$

- Иначе, если все образцы $p[x+xO, -1]$ с $x=0..3$ помечены как "доступен для Intra предсказания цветности", значения образцов предсказания $pred_C[x+xO, y+yO]$ с $x, y=0..3$ находят следующим образом:

$$pred_C[x+xO, y+yO] = \left(\sum_{x'=0}^3 p[x'+xO, -1] + 2 \right) \gg 2 \text{ с } x, y = 0..3. \quad (8-136)$$

- Иначе (некоторые образцы $p[x+xO, -1]$ с $x=0..3$ и некоторые образцы $p[-1, y+yO]$ с $y=0..3$ помечены как "не доступен для Intra предсказания цветности") значения образцов предсказания $pred_C[x+xO, y+yO]$, с $x, y=0..3$ находят следующим образом:

$$pred_C[x+xO, y+yO] = (1 \ll (\text{BitDepth}_C - 1)) \text{ с } x, y = 0..3. \quad (8-137)$$

8.3.4.2 Описание режима предсказания Intra_Chroma_Horizontal

Этот режим Intra предсказания цветности активируется, если `intra_chroma_pred_mode` равно 1.

Этот режим должен использоваться, только если образцы $p[-1, y]$ с $y = 0..MbHeightC - 1$ помечены как "доступен для Intra предсказания цветности".

Значения образцов предсказания $pred_c[x, y]$ находят следующим образом:

$$pred_c[x, y] = p[-1, y] \text{ с } x = 0..MbWidthC - 1 \text{ и } y = 0..MbHeightC - 1. \quad (8-138)$$

8.3.4.3 Описание режима предсказания Intra_Chroma_Vertical

Этот режим Intra предсказания цветности активируется, если `intra_chroma_pred_mode` равно 2.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..MbWidthC - 1$ помечены как "доступен для Intra предсказания цветности".

Значения образцов предсказания $pred_c[x, y]$ находят следующим образом:

$$pred_c[x, y] = p[x, -1] \text{ с } x = 0..MbWidthC - 1 \text{ и } y = 0..MbHeightC - 1. \quad (8-139)$$

8.3.4.4 Описание режима предсказания Intra_Chroma_Plane

Этот режим Intra предсказания цветности активируется, если `intra_chroma_pred_mode` равно 3.

Этот режим должен использоваться, только если образцы $p[x, -1]$ с $x = 0..MbWidthC - 1$ и $p[-1, y]$, с $y = -1..MbHeightC - 1$ помечены как "доступен для Intra предсказания цветности".

Допустим переменная `xCF` установлена равной $4 * (\text{chroma_format_idc} == 3)$, а переменная `yCF` установлена равной $4 * (\text{chroma_format_idc} != 1)$.

$$pred_c[x, y] = Clip1_c((a + b * (x - 3 - xCF) + c * (y - 3 - yCF) + 16) >> 5) \text{ с } x = 0..MbWidthC - 1 \text{ и } y = 0..MbHeightC - 1, \quad (8-140)$$

где:

$$a = 16 * (p[-1, MbHeightC - 1] + p[MbWidthC - 1, -1]); \quad (8-141)$$

$$b = ((34 - 29 * (\text{chroma_format_idc} == 3)) * H + 32) >> 6; \quad (8-142)$$

$$c = ((34 - 29 * (\text{chroma_format_idc} != 1)) * V + 32) >> 6, \quad (8-143)$$

а H и V определены следующим образом:

$$H = \sum_{x'=0}^{3+xCF} (x'+1) * (p[4+xCF+x', -1] - p[2+xCF-x', -1]); \quad (8-144)$$

$$V = \sum_{y'=0}^{3+yCF} (y'+1) * (p[-1, 4+yCF+y'] - p[-1, 2+yCF-y']). \quad (8-145)$$

8.3.5 Процесс создания образцов для макроблоков I_PCM

Этот процесс активируют, если `mb_type` равно `I_PCM`.

Переменную `du` находят следующим образом:

- если `MbaffFrameFlag` равно 1, а текущий макроблок – это макроблок поля, `du` устанавливают равным 2;
- иначе (`MbaffFrameFlag` равно 0 или текущий макроблок – это макроблок кадра) `du` устанавливают равным 1.

Положение верхнего левого образца яркости текущего макроблока находят, активируя процесс инверсного сканирования макроблока согласно п. 6.4.1 с CurrMbAddr как вход, и выходом, которому присваивается (xP, yP).

Созданные образцы яркости до процесса устранения блочности образуют как определено ниже:

```
for ( i = 0; i < 256; i++ )
    SL[ xP + ( i % 16 ), yP + dy * ( i / 16 ) ] = pcm_sample_luma[ i ].
```

(8-146)

Если chroma_format_idc не равно 0 (монохромное изображение), созданные образцы цветности до процесса устранения блочности образуют как описано ниже:

```
for ( i = 0; i < MbWidthC * MbHeightC; i++ ) {
    SCb[ ( xP / SubWidthC ) + ( i % MbWidthC ),
        ( ( yP + SubHeightC - 1 ) / SubHeightC ) + dy * ( i / MbWidthC ) ] =
        pcm_sample_chroma[ i ]
    SCr[ ( xP / SubWidthC ) + ( i % MbWidthC ),
        ( ( yP + SubHeightC - 1 ) / SubHeightC ) + dy * ( i / MbWidthC ) ] =
        pcm_sample_chroma[ i + MbWidthC * MbHeightC ]
}
```

(8-147)

8.4 Процесс Inter предсказания

Этот процесс активируют при декодировании макроблоков типов P и B.

Выходы этого процесса – образцы Inter предсказания для текущего макроблока, которые представляют массив 16x16 pred_L образцов яркости и, если chroma_format_idc не равно 0 (монохромное изображение), два массива 8x8 pred_{Cr} и pred_{Cb} образцов цветности, с одним из каждого компонента цветности Cb и Cr.

Разделение макроблока определено значением mb_type. Каждое разделение макроблока обозначают mbPartIdx. Если разделение макроблока состоит из разделенных частей, которые равны субмакроблокам, каждый субмакроблок может быть далее разделен на части субмакроблока, как определено значением sub_mb_type. Каждое разделение субмакроблока обозначают значением subMbPartIdx. Если разделение макроблока не состоит из субмакроблоков, subMbPartIdx устанавливают равным 0.

Для каждой части разделенного макроблока или субмакроблока определены следующие шаги.

Функции MbPartWidth(), MbPartHeight(), SubMbPartWidth() и SubMbPartHeight(), описывающие ширину и высоту частей макроблока и субмакроблока, определены в таблицах 7-13, 7-14, 7-17 и 7-18.

Диапазон индекса разделения макроблока mbPartIdx находят следующим образом:

- если mb_type равно B_Skip или B_Direct_16x16, mbPartIdx принимает значения 0..3;
- иначе (mb_type не равно B_Skip или B_Direct_16x16) mbPartIdx принимает значения 0..NumMbPart(mb_type) – 1.

Для всех значений mbPartIdx переменные partWidth и partHeight для каждого разделения макроблока или разделения субмакроблока в макроблоке находят следующим образом.

- Если mb_type не равно P_8x8, P_8x8ref0, B_Skip, B_Direct_16x16 или B_8x8, subMbPartIdx устанавливается равным 0, а partWidth и partHeight находят следующим образом:

$$\text{partWidth} = \text{MbPartWidth}(\text{mb_type});$$
(8-148)

$$\text{partHeight} = \text{MbPartHeight}(\text{mb_type});$$
(8-149)

- Иначе, если mb_type равно P_8x8 или P_8x8ref0, или mb_type равно B_8x8 и sub_mb_type[mbPartIdx] не равно B_Direct_8x8, subMbPartIdx принимает значения 0..NumSubMbPart(sub_mb_type) – 1, а partWidth и partHeight находят следующим образом:

$$\text{partWidth} = \text{SubMbPartWidth}(\text{sub_mb_type}[\text{mbPartIdx}]);$$
(8-150)

$$\text{partHeight} = \text{SubMbPartHeight}(\text{sub_mb_type}[\text{mbPartIdx}]);$$
(8-151)

- Иначе (mb_type равно B_Skip или B_Direct_16x16 или же mb_type равно B_8x8 , а $sub_mb_type[mbPartIdx]$ равно B_Direct_8x8) $subMbPartIdx$ принимает значения 0..3, а $partWidth$ и $partHeight$ находят следующим образом:

$$partWidth = 4; \quad (8-152)$$

$$partHeight = 4. \quad (8-153)$$

Если $chroma_format_idc$ не равно 0 (монохромное изображение), переменные $partWidthC$ и $partHeightC$ находят следующим образом:

$$partWidthC = partWidth / SubWidthC; \quad (8-154)$$

$$partHeightC = partHeight / SubHeightC. \quad (8-155)$$

Допустим, что переменная $MvCnt$ первоначально была установлена на 0 до всякой активации этого макроблока согласно п. 8.4.1.

Процесс $Inter$ предсказания для части макроблока $mbPartIdx$ и субмакроблока $subMbPartIdx$ состоит из следующих шагов.

1. Процесс вывода компонентов вектора движения и индексов контроля, как определено в п. 8.4.1.

Входы этого процесса:

- часть макроблока $mbPartIdx$;
- часть субмакроблока $subMbPartIdx$.

Выходы этого процесса:

- векторы движения яркости $mvL0$ и $mvL1$ и, если $chroma_format_idc$ не равно 0 (монохромное изображение), векторы движения цветности $mvCL0$ и $mvCL1$;
- индексы контроля $refIdxL0$ и $refIdxL1$;
- флаги использования списка предсказания $predFlagL0$ и $predFlagL1$;
- вычисление вектора движения части субмакроблока $subMvCnt$.

2. Приращение $subMvCnt$ переменной $MvCnt$.

3. Процесс декодирования образцов $Inter$ предсказания, как определено в п. 8.4.2.

Входы этого процесса:

- часть макроблока $mbPartIdx$;
- часть субмакроблока $subMbPartIdx$;
- переменные, определяющие ширину и высоту части: для яркости и цветности (если имеются) $partWidth$, $partHeight$, $partWidthC$ (если имеются) и $partHeightC$ (если имеются);
- векторы движения яркости $mvL0$ и $mvL1$, и, если $chroma_format_idc$ не равно 0 (монохромное изображение), цветности $mvCL0$ и $mvCL1$;
- индексы контроля $refIdxL0$ и $refIdxL1$;
- флаги использования списка предсказания $predFlagL0$ и $predFlagL1$.

Выходы этого процесса:

- образцы $inter$ предсказания ($pred$), которые представляют массив $(partWidth) \times (partHeight)$ $predPart_L$ образцов яркости предсказания и, если $chroma_format_idc$ не равно 0 (монохромное изображение), два $(partWidthC) \times (partHeightC)$ массива $predPart_{Cr}$ и $predPart_{Cb}$ образцов цветности предсказания, по одному для каждого из компонентов цветности Cb и Cr .

Для использования в процессе вывода переменных, активированных позже в процессе декодирования, сделаны следующие присвоения:

$$MvL0[mbPartIdx][subMbPartIdx] = mvL0; \quad (8-156)$$

$$MvL1[mbPartIdx][subMbPartIdx] = mvL1; \quad (8-157)$$

$$\text{RefIdxL0}[\text{mbPartIdx}] = \text{refIdxL0}; \quad (8-158)$$

$$\text{RefIdxL1}[\text{mbPartIdx}] = \text{refIdxL1}; \quad (8-159)$$

$$\text{PredFlagL0}[\text{mbPartIdx}] = \text{predFlagL0}; \quad (8-160)$$

$$\text{PredFlagL1}[\text{mbPartIdx}] = \text{predFlagL1}. \quad (8-161)$$

Положение верхнего левого образца части от разделения относительно верхнего левого образца макроблока находят, активируя процесс инверсного сканирования части макроблока, как описано в п. 6.4.2.1, с mbPartIdx в качестве входа и (xP, yP) в качестве выхода.

Положение верхнего левого образца вторично разделенной части макроблока относительно верхнего левого образца части макроблока находят, активируя процесс инверсного сканирования части субмакроблока, как описано в п. 6.4.2.2, с subMbPartIdx в качестве входа и (xS, yS) в качестве выхода.

Макроблок предсказания формируют, помещая образцы предсказания частей макроблока или субмакроблока на их корректные позиции в макроблоке следующим образом.

Переменную $\text{pred}_L[xP + xS + x, yP + yS + y]$ с $x = 0 \dots \text{partWidth} - 1, y = 0 \dots \text{partHeight} - 1$ находят как:

$$\text{pred}_L[xP + xS + x, yP + yS + y] = \text{predPart}_L[x, y]. \quad (8-162)$$

Если chroma_format_idc не равно 0 (монохромное изображение) переменную pred_C с $x = 0 \dots \text{partWidthC} - 1, y = 0 \dots \text{partHeightC} - 1$ и C в pred_C и predPart_C с заменой на C_b или C_r , находят следующим образом:

$$\text{pred}_C[xP / \text{SubWidthC} + xS / \text{SubWidthC} + x, yP / \text{SubHeightC} + yS / \text{SubHeightC} + y] = \text{predPart}_C[x, y]. \quad (8-163)$$

8.4.1 Процесс вывода компонентов вектора движения и индексов контроля

Входы в этот процесс:

- разделенная часть макроблока mbPartIdx ;
- разделенная часть субмакроблока subMbPartIdx .

Выходы этого процесса:

- векторы движения яркости mvL0 и mvL1 , а также векторы движения цветности mvCL0 и mvCL1 ;
- индексы контроля refIdxL0 и refIdxL1 ;
- флаги использования списка предсказания predFlagL0 и predFlagL1 ;
- переменная вычисления вектора движения части субмакроблока subMvCnt .

Для вывода переменных mvL0 и mvL1 , а также refIdxL0 и refIdxL1 используют следующие условия.

- Если mb_type равно P_Skip , активируют процесс вывода векторов движения яркости для пропущенных макроблоков в секциях P и SP согласно п. 8.4.1.1, с выходом векторов движения яркости mvL0 и индексов контроля refIdxL0 , а predFlagL0 устанавливают равным 1. Значения mvL1 и refIdxL1 помечают как не доступные, а predFlagL1 устанавливают равным 0. Переменную вычисленного вектора движения части субмакроблока subMvCnt устанавливают равной 1.
- Иначе, если mb_type равно B_Skip или B_Direct_16x16 , или $\text{sub_mb_type}[\text{mbPartIdx}]$ равно B_Direct_8x8 , активируют процесс вывода векторов движения яркости B_Skip , B_Direct_16x16 и B_Direct_8x8 в секциях B п. 8.4.1.2 с mbPartIdx и subMbPartIdx в качестве входа, а в качестве выхода – векторы движения яркости mvL0 , mvL1 , индексы контроля refIdxL0 , refIdxL1 , вычисления вектора движения части субмакроблока subMvCnt и флаги использования списка предсказания predFlagL0 и predFlagL1 .
- Иначе с заменой X на 0 или 1 в переменных predFlagLX , mvLX , refIdxLX , а также Pred_LX и в элементах синтаксиса ref_idx_IX и mvd_IX , используют следующие условия.

1. Переменные `refIdxLX` и `predFlagLX` находят следующим образом.
 - Если `MbPartPredMode(mb_type, mbPartIdx)` или `SubMbPredMode(sub_mb_type[mbPartIdx])` равно `Pred_LX` или `BiPred`:

$$\text{refIdxLX} = \text{ref_idx_lX}[\text{mbPartIdx}]; \quad (8-164)$$

$$\text{predFlagLX} = 1. \quad (8-165)$$
 - Иначе переменные `refIdxLX` и `predFlagLX` определяют как:

$$\text{refIdxLX} = -1; \quad (8-166)$$

$$\text{predFlagLX} = 0. \quad (8-167)$$
2. Переменную `subMvCnt` при вычислении вектора движения части субмакроблока устанавливают равной `predFlagL0 + predFlagL1`.
3. Переменную `currSubMbType` находят следующим образом:
 - если тип макроблока равен `B_8x8`, `currSubMbType` устанавливается равным `sub_mb_type[mbPartIdx]`;
 - иначе (тип макроблока не равен `B_8x8`) `currSubMbType` устанавливается равным "na".
4. Если `predFlagLX` равно 1, активируют процесс вывода предсказания вектора движения согласно п. 8.4.1.3 с `mbPartIdx`, `subMbPartIdx`, `refIdxLX` и суффиксом списка `LX` в качестве входа, а в качестве выхода – `mvpLX`. Векторы движения яркости находят как:

$$\text{mvLX}[0] = \text{mvpLX}[0] + \text{mvd_lX}[\text{mbPartIdx}][\text{subMbPartIdx}][0]; \quad (8-168)$$

$$\text{mvLX}[1] = \text{mvpLX}[1] + \text{mvd_lX}[\text{mbPartIdx}][\text{subMbPartIdx}][1]. \quad (8-169)$$

Для вывода переменных векторов движения цветности используют следующие условия. Если `predFlagLX` (с `X` либо 0, либо 1) равно 1, активируют процесс вывода векторов движения цветности согласно п. 8.4.1.4 с `mvLX` и `refIdxLX` в качестве входа, а в качестве выхода – `mvCLX`.

8.4.1.1 Процесс вывода векторов движения яркости пропущенных макроблоков в секциях P и SP

Процесс активируют, если `mb_type` равно `P_Skip`.

Выходы этого процесса – вектор движения `mvL0` и индекс контроля `refIdxL0`.

Индекс контроля `refIdxL0` для пропущенного макроблока находят следующим образом:

$$\text{refIdxL0} = 0. \quad (8-170)$$

Для вывода вектора движения `mvL0` макроблока типа `P_Skip` используют следующие условия.

- Активируют процесс, определенный в п. 8.4.1.3.2, с `mbPartIdx`, установленным равным 0, `subMbPartIdx`, установленным равным 0, `currSubMbType`, установленным равным "na", и `listSuffixFlag`, установленным равным 0, в качестве входа, а выход присваивается `mbAddrA`, `mbAddrB`, `mvL0A`, `mvL0B`, `refIdxL0A` и `refIdxL0B`.
- Переменная `mvL0` определена следующим образом.
 - Если любое из следующих условий – истина, то оба компонента вектора движения `mvL0` устанавливают на 0:
 - `mbAddrA` не доступно;
 - `mbAddrB` не доступно;
 - `refIdxL0A` равно 0, и оба компонента `mvL0A` равны 0;
 - `refIdxL0B` равно 0, и оба компонента `mvL0B` равны 0.

- Иначе активируют процесс для вывода предсказания вектора движения яркости, как определено в п. 8.4.1.3, с `mbPartIdx = 0`, `subMbPartIdx = 0`, `refIdxL0` и `currSubMbType = "na"` в качестве входов, а в качестве выхода – присвоенное значение `mvL0`.

ПРИМЕЧАНИЕ. – Выходу прямо присвоено значение `mvL0`, поскольку предсказанный параметр равен действительному вектору движения.

8.4.1.2 Процесс вывода векторов движения яркости **V_Skip**, **V_Direct_16x16** и **V_Direct_8x8**

Процесс активируют, если `mb_type` равно `V_Skip` или `V_Direct_16x16`, или `sub_mb_type[mbPartIdx]` равно `V_Direct_8x8`.

Входы этого процесса – `mbPartIdx` и `subMbPartIdx`.

Выходы этого процесса – это индексы контроля `refIdxL0`, `refIdxL1`, векторы движения `mvL0` и `mvL1`, вычисленная часть вектора движения `subMvCnt` и флаги использованных списков предсказаний `predFlagL0` и `predFlagL1`.

Процесс вывода зависит от значения `direct_spatial_mv_pred_flag`, которое представлено в потоке битов в синтаксисе заголовка секции, как определено в п. 7.3.3, и описано следующим образом.

- Если `direct_spatial_mv_pred_flag` равно 1, то режим, в котором этот процесс находится на выходе, считают пространственным прямым режимом предсказания.
- Иначе (`direct_spatial_mv_pred_flag` равно 0) режим, в котором этот процесс находится на выходе, считают временным прямым режимом предсказания.

Оба эти прямые режимы предсказания, пространственный и временный, используют близко расположенные векторы движения и индексы контроля, как это определено в п. 8.4.1.2.1.

Векторы движения и индексы контроля находят следующим образом.

- Если используют пространственный прямой режим предсказания, то применяют прямой вектор движения и режим предсказания индекса контроля, определенный в п. 8.4.1.2.2, с `subMvCnt` в качестве выхода.
- Иначе (используют временный прямой режим предсказания) используют прямой вектор движения и режим предсказания индекса контроля, определенный в п. 8.4.1.2.3, а переменную `subMvCnt` находят следующим образом:
 - если `subMbPartIdx` равно 0, `subMvCnt` устанавливают равным 2;
 - иначе (`subMbPartIdx` не равно 0) `subMvCnt` устанавливают равным 0.

8.4.1.2.1 Процесс вывода совпадающих разделенных частей субмакроблоков 4x4

Входы в этот процесс – `mbPartIdx` и `subMbPartIdx`.

Выходы этого процесса – изображение `colPic`, совпадающий (по месту) макроблок `mbAddrCol`, вектор движения `mvCol`, индекс контроля `refIdxCol` и переменная `vertMvScale` (которыми могут быть `One_To_One`, `Frm_To_Fld` или `Fld_To_Frm`).

Если `RefPicList1[0]` – это кадр или пара дополнительных полей, то допустим, что `firstRefPicL1Top` и `firstRefPicL1Bottom` – это верхнее и нижнее поля `RefPicList1[0]`, соответственно, и положим, что следующие переменные определены как:

$$\text{topAbsDiffPOC} = \text{Abs}(\text{DiffPicOrderCnt}(\text{firstRefPicL1Top}, \text{CurrPic})); \quad (8-171)$$

$$\text{bottomAbsDiffPOC} = \text{Abs}(\text{DiffPicOrderCnt}(\text{firstRefPicL1Bottom}, \text{CurrPic})). \quad (8-172)$$

Переменная colPic определяет изображение, которое содержит совпадающий по месту макроблок, как определено в таблице 8-6.

Таблица 8-6 – Спецификация переменной colPic

field_pic_flag	RefPicList1 [0] – это ...	mb_field_decoding_flag	Дополнительное условие	colPic
1	поле декодированного кадра			кадр, содержащий RefPicList1[0]
	декодированное поле			RefPicList1[0]
0	декодированный кадр			RefPicList1[0]
	пара дополнительных полей	0	topAbsDiffPOC < bottomAbsDecRefPOC	firstRefPicL1Top
			topAbsDiffPOC >= bottomAbsDiffPOC	firstRefPicL1Bottom
		1	(CurrMbAddr & 1) == 0	firstRefPicL1Top
(CurrMbAddr & 1) != 0			firstRefPicL1Bottom	

Если direct_8x8_inference_flag равно 1, subMbPartIdx устанавливают следующим образом:

$$\text{subMbPartIdx} = \text{mbPartIdx}. \quad (8-173)$$

Положим, что PicCodingStruct(X) – функция с аргументом X либо CurrPic, либо colPic. Это показано в таблице 8-7.

Таблица 8-7 – Спецификация PicCodingStruct(X)

X кодирован с field_pic_flag, равным ...	mb_adaptive_frame_field_flag	PicCodingStruct(X)
1		FLD
0	0	FRM
0	1	AFRM

При luma4x4BlkIdx = mbPartIdx * 4 + subMbPartIdx, активируют процесс инверсного сканирования блока яркости 4x4, как определено в п. 6.4.3, с luma4x4BlkIdx в качестве входа и (x, y), присвоенного значению (xCol, yCol), в качестве выхода.

Таблица 8-8 определяет совпадающий адрес макроблока mbAddrCol, yM, и переменную vertMvScale в два этапа.

1. Спецификация адреса макроблока mbAddrX в зависимости от PicCodingStruct(CurrPic) и PicCodingStruct(colPic).

ПРИМЕЧАНИЕ. – Для CurrPic и colPic невозможны типы кодирующих изображений ни (FRM, AFRM), ни (AFRM, FRM), т. к. эти типы кодирующих изображений должны быть разделены IDR изображением.

2. Спецификацию mbAddrCol, yM и vertMvScale, зависящую от mb_field_decoding_flag и переменной fieldDecodingFlagX, находят следующим образом:

- если макроблок mbAddrX в изображении colPic – это макроблок поля, то fieldDecodingFlagX устанавливают равным 1;
- иначе (макроблок mbAddrX в изображении colPic – это макроблок кадра) fieldDecodingFlagX устанавливают равным 0.

Неопределенные значения в таблице 8-8 указывают, что значение соответствующей переменной не отвечает текущему ряду таблицы.

Значение mbAddrCol устанавливают равным CurrMbAddr или одному из следующих значений:

$$\text{mbAddrCol1} = 2 * \text{PicWidthInMbs} * (\text{CurrMbAddr} / \text{PicWidthInMbs}) + (\text{CurrMbAddr} \% \text{PicWidthInMbs}) + \text{PicWidthInMbs} * (\text{yCol} / 8); \quad (8-174)$$

$$\text{mbAddrCol2} = 2 * \text{CurrMbAddr} + (\text{yCol} / 8); \quad (8-175)$$

$$\text{mbAddrCol3} = 2 * \text{CurrMbAddr} + \text{bottom_field_flag}; \quad (8-176)$$

$$\text{mbAddrCol4} = \text{PicWidthInMbs} * (\text{CurrMbAddr} / (2 * \text{PicWidthInMbs})) + (\text{CurrMbAddr} \% \text{PicWidthInMbs}); \quad (8-177)$$

$$\text{mbAddrCol5} = \text{CurrMbAddr} / 2; \quad (8-178)$$

$$\text{mbAddrCol6} = 2 * (\text{CurrMbAddr} / 2) + ((\text{topAbsDiffPOC} < \text{bottomAbsDiffPOC}) ? 0 : 1); \quad (8-179)$$

$$\text{mbAddrCol7} = 2 * (\text{CurrMbAddr} / 2) + (\text{yCol} / 8); \quad (8-180)$$

Таблица 8-8 – Спецификация mbAddrCol, yM и vertMvScale

PicCodingStruct(CurrPic)	PicCodingStruct(colPic)	mbAddrX	mb_field_decoded_flag	fieldDecodingFlagX	mbAddrCol	yM	vertMvScale
FLD	FLD				CurrMbAddr	yCol	One_To_One
	FRM				mbAddrCol1	(2 * yCol) % 16	Frm_To_Fld
	AFRM	2*CurrMbAddr	0		mbAddrCol2	(2 * yCol) % 16	Frm_To_Fld
1				mbAddrCol3	yCol	One_To_One	
FRM	FLD				mbAddrCol4	8 * ((CurrMbAddr / PicWidthInMbs) % 2) + 4 * (yCol / 8)	Fld_To_Frm
	FRM				CurrMbAddr	yCol	One_To_One
AFRM	FLD		0		mbAddrCol5	8 * (CurrMbAddr % 2) + 4 * (yCol / 8)	Fld_To_Frm
			1		mbAddrCol5	yCol	One_To_One
	AFRM	CurrMbAddr	0	0	CurrMbAddr	yCol	One_To_One
				1	mbAddrCol6	8 * (CurrMbAddr % 2) + 4 * (yCol / 8)	Fld_To_Frm
			1	0	mbAddrCol7	(2 * yCol) % 16	Frm_To_Fld
1	CurrMbAddr	yCol		One_To_One			

Положим, что mbPartIdxCol – индекс разделенной части макроблока, расположенной вместе с макроблоком, а subMbPartIdxCol – субиндекс разделенной части макроблока, расположенной вместе с субмакроблоком. Часть в макроблоке mbAddrCol внутри изображения colPic, покрывающего образец (xCol, yM), присваивается mbPartIdxCol, а часть субмакроблока внутри разделенной части mbPartIdxCol, покрывающей образец (xCol, yM) в макроблоке mbAddrCol внутри изображения colPic, присваивается subMbPartIdxCol.

Флаги использованных предсказаний `predFlagL0Col` и `predFlagL1Col` устанавливаются равными `PredFlagL0[mbPartIdxCol]` и `PredFlagL1[mbPartIdxCol]` соответственно. Это флаги использованных предсказаний, которые присвоены частям макроблока `mbAddrCol\mbPartIdxCol` внутри изображения `colPic`.

Вектор движения `mvCol` и индекс контроля `refIdxCol` находят следующим образом.

- Если макроблок `mbAddrCol` кодирован в режиме Intra предсказания макроблока или если оба флага использованных предсказаний, `predFlagL0Col` и `predFlagL1Col`, равны 0, то оба компонента `mvCol` устанавливаются равными 0, а `refIdxCol` устанавливается равным –1.
- Иначе используют следующие условия:
 - если `predFlagL0Col` равно 1, вектор движения `mvCol` и индекс контроля `refIdxCol` устанавливаются равными `MvL0[mbPartIdxCol][subMbPartIdxCol]` и `RefIdxL0[mbPartIdxCol]` соответственно. Эти значения – вектор движения `mvL0` и индекс контроля `refIdxL0`, которые присвоены частям (суб)макроблока `mbAddrCol\mbPartIdxCol\subMbPartIdxCol` внутри изображения `colPic`;
 - иначе (`predFlagL0Col` равно 0 и `predFlagL1Col` равно 1) вектор движения `mvCol` и индекс контроля `refIdxCol` устанавливаются равными `MvL1[mbPartIdxCol][subMbPartIdxCol]` и `RefIdxL1[mbPartIdxCol]` соответственно. Эти значения – вектор движения `mvL1` и индекс контроля `refIdxL1`, которые присвоены частям (суб)макроблока `mbAddrCol\mbPartIdxCol\subMbPartIdxCol` внутри изображения `colPic`.

8.4.1.2.2 Процесс вывода режима предсказания пространственного прямого вектора движения яркости и индекса контроля

Процесс активируют, если `direct_spatial_mv_pred_flag` равно 1 и любое из следующих условий является истиной:

- `mb_type` равно `B_Skip`;
- `mb_type` равно `B_Direct_16x16`;
- `sub_mb_type[mbPartIdx]` равно `B_Direct_8x8`.

Входы в этот процесс – `mbPartIdx`, `subMbPartIdx`.

Выходы из этого процесса – индексы контроля `refIdxL0`, `refIdxL1`, векторы движения `mvL0` и `mvL1`, вычисленный вектор движения субчасти `subMvCnt` и флаги использованных списков предсказаний, `predFlagL0` и `predFlagL1`.

Индексы контроля `refIdxL0` и `refIdxL1`, а также переменную `directZeroPredictionFlag` находят, применяя следующие шаги по порядку.

1. Положим, что переменная `currSubMbType` установлена равной `sub_mb_type[mbPartIdx]`.
2. Активируют процесс, определенный в п. 8.4.1.3.2, с `mbPartIdx = 0`, `subMbPartIdx = 0`, `currSubMbType` и `listSuffixFlag = 0` в качестве входов, а в качестве выхода – присвоенные значения векторам движения `mvL0N` и индексам контроля `refIdxL0N` с заменой N на A, B или C.
3. Активируют процесс, определенный в п. 8.4.1.3.2, с `mbPartIdx = 0`, `subMbPartIdx = 0`, `currSubMbType` и `listSuffixFlag = 1` в качестве входов, а в качестве выхода – присвоенные значения векторам движения `mvL1N` и индексам контроля `refIdxL1N` с заменой N на A, B или C.

ПРИМЕЧАНИЕ 1. – Векторы движения `mvL0N`, `mvL1N` и индексы контроля `refIdxL0N`, `refIdxL1N` идентичны для всех разделенных частей субмакроблока 4x4 любого макроблока.

4. Индексы контроля `refIdxL0`, `refIdxL1` и `directZeroPredictionFlag` находят как:

$$\text{refIdxL0} = \text{MinPositive}(\text{refIdxL0A}, \text{MinPositive}(\text{refIdxL0B}, \text{refIdxL0C})); \quad (8-181)$$

$$\text{refIdxL1} = \text{MinPositive}(\text{refIdxL1A}, \text{MinPositive}(\text{refIdxL1B}, \text{refIdxL1C})); \quad (8-182)$$

$$\text{directZeroPredictionFlag} = 0, \quad (8-183)$$

где:

$$\text{MinPositive}(x, y) = \begin{cases} \text{Min}(x, y) & \text{если } x \geq 0 \text{ и } y \geq 0, \\ \text{Max}(x, y) & \text{в иных случаях.} \end{cases} \quad (8-184)$$

5. Если оба индекса контроля refIdxL0 и refIdxL1 меньше 0:

$$\text{refIdxL0} = 0; \quad (8-185)$$

$$\text{refIdxL1} = 0; \quad (8-186)$$

$$\text{directZeroPredictionFlag} = 1. \quad (8-187)$$

Активируют процесс, определенный в п. 8.4.1.2.1, с mbPartIdx, subMbPartIdx, заданными в качестве входа, а в качестве выхода – присвоенные значения refIdxCol и mvCol.

Переменную colZeroFlag находят следующим образом.

- Если все из следующих условий истинны, то colZeroFlag устанавливают равным 1.
 - RefPicList1[0] помечено как "используется для краткосрочного контроля".
 - refIdxCol равно 0.
 - Оба компонента вектора движения mvCol[0] и mvCol[1] находятся в диапазоне от –1 до 1 в единицах, определенных следующим образом:
 - если имеющий то же местоположение макроблок – это макроблок кадра, единицы mvCol[0] и mvCol[1] являются единицами четверти яркости образцов кадра;
 - иначе (имеющий то же местоположение макроблок – это макроблок поля) единицы mvCol[0] и mvCol[1] являются единицами четверти яркости образцов поля.

ПРИМЕЧАНИЕ 2. – Для определенности установленного выше условия значение mvCol[1] не масштабируют, чтобы использовать единицы вектора движения для текущего макроблока в случаях, когда текущий макроблок является макроблоком кадра, а имеющий то же местоположение макроблок является макроблоком поля или если текущий макроблок является макроблоком поля, а имеющий то же местоположение макроблок является макроблоком кадра. Этот аспект отличается от использования mvCol[1] во временном прямом режиме, как определено в п. 8.4.1.2.3, в котором применяют вектор движения имеющего то же местоположение макроблока, чтобы использовать те же самые единицы, что и для вектора движения текущего макроблока, используя в этих случаях уравнение 8-190 или уравнение 8-191.

- Иначе colZeroFlag устанавливают равным 0.

Векторы движения mvLX (с X равным 0 или 1) находят следующим образом.

- Если любое из следующих условий – истина, оба компонента вектора движения mvLX устанавливают равным 0:
 - directZeroPredictionFlag равно 1;
 - refIdxLX меньше 0;
 - refIdxLX равно 0 и colZeroFlag равно 1.
- Иначе активируют процесс, определенный в п. 8.4.1.3, с mbPartIdx = 0, subMbPartIdx = 0, refIdxLX и currSubMbType в качестве входов, а в качестве выхода – значение, присвоенное mvLX.

ПРИМЕЧАНИЕ 3. – Вектор движения mvLX, полученный согласно п. 8.4.1.3, идентичен для всех частей 4x4 субмакроблока макроблока, для которого активирован указанный процесс..

Флаги использованных предсказаний predFlagL0 и predFlagL1 находят согласно таблице 8-9.

Таблица 8-9 – Назначения флагов использованных предсказаний

refIdxL0	refIdxL1	predFlagL0	predFlagL1
>= 0	>= 0	1	1
>= 0	< 0	1	0
< 0	>= 0	0	1

Переменную subMvCnt находят следующим образом:

- если subMbPartIdx не равно 0 или direct_8x8_inference_flag равно 0, то subMvCnt устанавливают равным 0;
- иначе (subMbPartIdx равно 0 и direct_8x8_inference_flag равно 1) subMvCnt устанавливают равным predFlagL0 + predFlagL1.

8.4.1.2.3 Процесс вывода режимов предсказания временного прямого вектора движения яркости и индекса контроля

Этот процесс активируют, если `direct_spatial_mv_pred_flag` равно 0, а любое из следующих условий – истина:

- `mb_type` равно `B_Skip`;
- `mb_type` равно `B_Direct_16x16`;
- `sub_mb_type[mbPartIdx]` равно `B_Direct_8x8`.

Входы в этот процесс – `mbPartIdx` и `subMbPartIdx`.

Выходы из этого процесса – векторы движения `mvL0` и `mvL1`, индексы контроля `refIdxL0` и `refIdxL1` и флаги использованных списков предсказаний, `predFlagL0` и `predFlagL1`.

Процесс, определенный в п. 8.4.1.2.1, активируют с `mbPartIdx`, `subMbPartIdx`, заданными в качестве входа, а в качестве выхода – присвоенные значения `colPic`, `mbAddrCol`, `mvCol`, `refIdxCol` и `vertMvScale`.

Индексы контроля `refIdxL0` и `refIdxL1` находят следующим образом:

$$\text{refIdxL0} = ((\text{refIdxCol} < 0) ? 0 : \text{MapColToList0}(\text{refIdxCol})); \quad (8-188)$$

$$\text{refIdxL1} = 0. \quad (8-189)$$

ПРИМЕЧАНИЕ 1. – Если текущий макроблок – это макроблок поля, то `refIdxL0` и `refIdxL1` – индексы списка поля; иначе (текущий макроблок – макроблок кадра) `refIdxL0` и `refIdxL1` – индексы списка кадра или пары дополнительных контрольных полей.

Положим, что `refPicCol` – это кадр, поле или пара дополнительных полей, которые направляет индекс контроля `refIdxCol` при декодировании расположенного в том же месте макроблока `mbAddrCol` внутри изображения `colPic`. Функция `MapColToList0(refIdxCol)` определена следующим образом.

- Если `vertMvScale` равно `One_To_One`, используют следующие условия.
 - Если `field_pic_flag` равно 0, а текущий макроблок – макроблок поля, используют следующие условия.
 - Положим, что `refIdxL0Frm` – это наименьший обозначенный индекс контроля в текущем списке контрольных изображений `RefPicList0`, которое сравнивает кадр или пару дополнительных полей, содержащих поле `refPicCol`. `RefPicList0` должен содержать кадр или пару дополнительных полей, которая содержит поле `refPicCol`. Возвратное значение `MapColToList0()` определено следующим образом:
 - если поле, на которое ссылается `refIdxCol`, имеет ту же четность, что и текущий макроблок, то `MapColToList0(refIdxCol)` возвращает индекс контроля (`refIdxL0Frm << 1`);
 - иначе (поле, на которое ссылается `refIdxCol`, имеет противоположную четность текущего макроблока) `MapColToList0(refIdxCol)` возвращает индекс контроля ((`refIdxL0Frm << 1`) + 1).
 - Иначе (`field_pic_flag` равно 1 или текущий макроблок – это макроблок кадра) `MapColToList0(refIdxCol)` возвращает наименьший обозначенный индекс контроля `refIdxL0` в текущем списке контрольных изображений `RefPicList0`, который контролирует `refPicCol`. Значение `RefPicList0` должно содержать `refPicCol`.
 - Иначе, если `vertMvScale` равно `Frm_To_Fld`, используют следующие условия.
 - Если `field_pic_flag` равно 0, положим, что `refIdxL0Frm` – наименьшее значение индекса контроля в текущем списке контрольных изображений `RefPicList0`, который контролирует `refPicCol`. Значение `MapColToList0(refIdxCol)` возвращает индекс контроля (`refIdxL0Frm << 1`). `RefPicList0` должно содержать `refPicCol`.
 - Иначе (`field_pic_flag` равно 1) `MapColToList0(refIdxCol)` возвращает наименьшее значение индекса контроля `refIdxL0` в текущем списке контрольных изображений `RefPicList0`, которое сравнивает поле `refPicCol` с той же четностью, что и текущего изображения `CurrPic`. `RefPicList0` должно содержать `refPicCol` с той же четностью, что и текущего изображения `CurrPic`.
 - Иначе (`vertMvScale` равно `Fld_To_Frm`) `MapColToList0(refIdxCol)` возвращает наименьшее значение индекса контроля `refIdxL0` в текущем списке контрольных изображений `RefPicList0`, которое контролирует кадр или пару дополнительных полей, содержащих `refPicCol`. Значение `RefPicList0` должно содержать кадр или пару дополнительных полей, которая содержит поле `refPicCol`.

ПРИМЕЧАНИЕ 2. – Декодированное контрольное изображение, которое было помечено как "используется для краткосрочного контроля" в то время, когда его использовали как эталонное (контрольное) в процессе декодирования изображения, содержащего в том же месте макроблок, можно было бы модифицировать и пометить как "используется для долгосрочного контроля" прежде, чем использовать его в качестве эталонного для `inter` предсказания с режимом прямого предсказания текущего макроблока.

В зависимости от значения `vertMvScale` вертикальный компонент `mvCol` модифицируют следующим образом.

- Если `vertMvScale` равно `Frm_To_Fld`:

$$mvCol[1] = mvCol[1] / 2. \quad (8-190)$$

- Иначе, если `vertMvScale` равно `Fld_To_Frm`:

$$mvCol[1] = mvCol[1] * 2. \quad (8-191)$$

- Иначе (`vertMvScale` равно `One_To_One`) `mvCol[1]` остается неизменным.

Переменные `currPicOrField`, `pic0` и `pic1` находят следующим образом.

- Если `field_pic_flag` равно 0, а текущий макроблок – это макроблок поля, используют следующие условия.
 - `currPicOrField` – это поле текущего изображения `CurrPic`, которое имеет ту же четность, что и текущий макроблок.
 - `pic1` – это поле `RefPicList1[0]`, которое имеет ту же четность, что и текущий макроблок.
 - Переменную `pic0` находят следующим образом:
 - если `refIdxL0 % 2` равно 0, `pic0` – это поле `RefPicList0[refIdxL0 / 2]`, которое имеет ту же четность, что и текущий макроблок;
 - иначе (`refIdxL0 % 2` не равно 0) `pic0` – это поле `RefPicList0[refIdxL0 / 2]`, которое имеет противоположную текущему макроблоку четность.
- Иначе (`field_pic_flag` равно 1 или текущий макроблок – это макроблок кадра) `currPicOrField` – текущее изображение `CurrPic`, `pic1` – декодированное контрольное изображение `RefPicList1[0]`, а `pic0` – декодированное контрольное изображение `RefPicList0[refIdxL0]`.

Два вектора движения `mvL0` и `mvL1` каждой части 4x4 субмакроблока текущего макроблока находят следующим образом.

ПРИМЕЧАНИЕ 3. – Часто встречается ситуации, когда многие части субмакроблока 4x4 имеют общие векторы движения и контрольные изображения. В этих случаях временный режим прямой компенсации движения может вычислить значение образца `Inter` предсказания в единицах блоков образцов яркости больших размеров, чем 4x4. Например, если `direct_8x8_inference_flag` равно 1, по крайней мере, каждый квадрат образца яркости 8x8 макроблока совместно использует те же векторы движения и контрольные изображения.

- Если индекс контроля `refIdxL0` указывает на долгосрочное контрольное изображение или `DiffPicOrderCnt(pic1, pic0)` равно 0, векторы движения `mvL0`, `mvL1` режима прямого разделения на части находят как:

$$mvL0 = mvCol; \quad (8-192)$$

$$mvL1 = 0. \quad (8-193)$$

- Иначе векторы движения `mvL0`, `mvL1` находят как масштабированные версии вектора движения `mvCol`, расположенного в том же месте части субмакроблока, как определено ниже (см. рисунок 8-2):

$$tx = (16384 + Abs(td / 2)) / td; \quad (8-194)$$

$$DistScaleFactor = Clip3(-1024, 1023, (tb * tx + 32) >> 6); \quad (8-195)$$

$$mvL0 = (DistScaleFactor * mvCol + 128) >> 8; \quad (8-196)$$

$$mvL1 = mvL0 - mvCol, \quad (8-197)$$

где `tb` и `td` находят следующим образом:

$$tb = Clip3(-128, 127, DiffPicOrderCnt(currPicOrField, pic0)); \quad (8-198)$$

$$td = Clip3(-128, 127, DiffPicOrderCnt(pic1, pic0)). \quad (8-199)$$

ПРИМЕЧАНИЕ 4. – Значения $mvL0$ и $mvL1$ не могут выходить за диапазоны, определенные в Приложении А.

Оба флага использованных предсказаний $predFlagL0$ и $predFlagL1$ устанавливаются на 1.

Рисунок 8-2 иллюстрирует временной прямой режим предполагаемого вектора движения, если текущее изображение временно находится между контрольным изображением из списка контрольных изображений 0 и контрольным изображением из списка контрольных изображений 1.

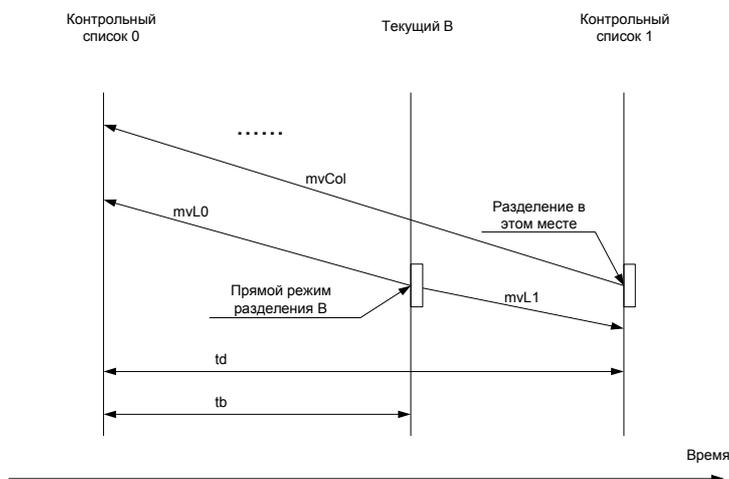


Рисунок 8-2 – Пример временного прямого режима предполагаемого вектора движения (для информации)

8.4.1.3 Процесс вывода предсказания вектора движения яркости

Входы в этот процесс:

- индекс разделенной части макроблока $mbPartIdx$;
- индекс разделенной части субмакроблока $subMbPartIdx$;
- индекс контроля текущей части $refIdxLX$ (с X , равным 0 или 1);
- переменная $currSubMbType$.

Выход этого процесса – предсказания $mvpLX$ вектора движения $mvLX$ (с X , равным 0 или 1).

Процесс вывода смежных блоков для данных движения согласно п. 8.4.1.3.2 активируют с $mbPartIdx$, $subMbPartIdx$, $currSubMbType$ и $listSuffixFlag = X$ (с X , равным 0 или 1 для $refIdxLX$, равным $refIdxL0$ или $refIdxL1$, соответственно) в качестве входа и с $mbAddrN\mbPartIdxN\subMbPartIdxN$, индексами контроля $refIdxLXN$ и векторами движения $mvLXN$ с заменой N на A , B или C – в качестве выхода.

Процесс вывода для предсказания медианы вектора движения яркости согласно п. 8.4.1.3.1 активируют с $mbAddrN\mbPartIdxN\subMbPartIdxN$, $mvLXN$, $refIdxLXN$ с заменой N на A , B или C , и $refIdxLX$ в качестве входа и $mvpLX$ – в качестве выхода, пока одно из следующих условий не будет истиной.

- $MbPartWidth(mb_type)$ равно 16, $MbPartHeight(mb_type)$ равно 8, $mbPartIdx$ равно 0 и $refIdxLXB$ равно $refIdxLX$:

$$mvpLX = mvLXB. \quad (8-200)$$

- $MbPartWidth(mb_type)$ равно 16, $MbPartHeight(mb_type)$ равно 8, $mbPartIdx$ равно 1 и $refIdxLXA$ равно $refIdxLX$:

$$mvpLX = mvLXA. \quad (8-201)$$

- $MbPartWidth(mb_type)$ равно 8, $MbPartHeight(mb_type)$ равно 16, $mbPartIdx$ равно 0 и $refIdxLXA$ равно $refIdxLX$:

$$mvpLX = mvLXA. \quad (8-202)$$

- MbPartWidth(mb_type) равно 8, MbPartHeight(mb_type) равно 16, mbPartIdx равно 1 и refIdxLXC равно refIdxLX:

$$mvpLX = mvLXC. \quad (8-203)$$

Рисунок 8-3 иллюстрирует не медианные предсказания, как описано выше.

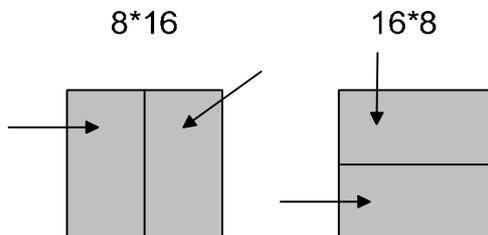


Рисунок 8-3 – Предсказания ориентированной сегментации (для информации)

8.4.1.3.1 Процесс вывода предсказания медианы вектора движения яркости

Входы в этот процесс:

- смежные части mbAddrN\mbPartIdxN\subMbPartIdxN (с заменой N на A, B или C);
- векторы движения mvLXN (с заменой N на A, B или C) смежных частей;
- индексы контроля refIdxLXN (с заменой N на A, B или C) смежных частей; и
- индекс контроля refIdxLX текущей части.

Выход этого процесса – предсказание вектора движения mvpLX.

Переменную mvpLX находят следующим образом.

- Если обе части mbAddrB\mbPartIdxB\subMbPartIdxB и mbAddrC\mbPartIdxC\subMbPartIdxC не доступны, а mbAddrA\mbPartIdxA\subMbPartIdxA доступно:

$$mvLXB = mvLXA; \quad (8-204)$$

$$mvLXC = mvLXA; \quad (8-205)$$

$$refIdxLXB = refIdxLXA; \quad (8-206)$$

$$refIdxLXC = refIdxLXA. \quad (8-207)$$

- В зависимости от индексов контроля refIdxLXA, refIdxLXB или refIdxLXC используют следующие условия.
 - Если один и только один из индексов контроля refIdxLXA, refIdxLXB или refIdxLXC равен индексу контроля refIdxLX текущей части, используют следующие условия. Положим, что refIdxLXN – индекс контроля, который равен refIdxLX, а вектор движения mvLXN присвоен предсказанию вектора движения mvpLX:

$$mvpLX = mvLXN. \quad (8-208)$$

- Иначе каждый компонент предсказания вектора движения mvpLX задан медианой соответствующих векторных компонентов вектора движения mvLXA, mvLXB и mvLXC:

$$mvpLX[0] = \text{Median}(mvLXA[0], mvLXB[0], mvLXC[0]); \quad (8-209)$$

$$mvLX[1] = \text{Median}(mvLXA[1], mvLXB[1], mvLXC[1]). \quad (8-210)$$

8.4.1.3.2 Процесс вывода данных движения смежных частей

Входы в этот процесс:

- индекс разделенной части макроблока mbPartIdx;
- индекс разделенной части субмакроблока subMbPartIdx;
- тип текущего субмакроблока currSubMbType;
- суффикс списка flag listSuffixFlag.

Выходы из этого процесса (с заменой N на A, B или C):

- mbAddrN\mbPartIdxN\subMbPartIdxN, определяющие смежные части;
- векторы движения mvLXN смежных частей; и
- индексы контроля refIdxLXN смежных частей.

Имена переменных, которые включают строку "LX", интерпретируются как имеющие X в значении flag listSuffixFlag.

Части mbAddrN\mbPartIdxN\subMbPartIdxN с заменой N на A, B или C находят в следующем порядке.

1. Положим, что mbAddrD\mbPartIdxD\subMbPartIdxD – переменные, определяющие дополнительную смежную часть.
2. Процесс согласно п. 6.4.8.5 активируют с mbPartIdx, currSubMbType и subMbPartIdx в качестве входа, а в качестве выхода – присвоенные значения mbAddrN\mbPartIdxN\subMbPartIdxN с заменой N на A, B, C или D.
3. Если часть mbAddrC\mbPartIdxC\subMbPartIdxC не доступна, используют следующие условия:

$$mbAddrC = mbAddrD; \quad (8-211)$$

$$mbPartIdxC = mbPartIdxD; \quad (8-212)$$

$$subMbPartIdxC = subMbPartIdxD. \quad (8-213)$$

Векторы движения mvLXN и индексы контроля refIdxLXN (с заменой N на A, B или C) находят следующим образом.

- Если часть макроблока или часть субмакроблока mbAddrN\mbPartIdxN\subMbPartIdxN не доступны или mbAddrN кодировано в режиме Intra предсказания, или если predFlagLX значения mbAddrN\mbPartIdxN\subMbPartIdxN равно 0, оба компонента mvLXN устанавливают равными 0, а refIdxLXN устанавливают равным -1.
- Иначе используют следующие условия.
 - Вектор движения mvLXN и индекс контроля refIdxLXN устанавливают равными $MvLX[mbPartIdxN][subMbPartIdxN]$ и $RefIdxLX[mbPartIdxN]$, которые, соответственно, являются вектором движения mvLX и индексом контроля refIdxLX, присвоенными части (суб)макроблока mbAddrN\mbPartIdxN\subMbPartIdxN.
 - Далее переменные mvLXN[1] и refIdxLXN обрабатывают следующим образом.

- Если текущий макроблок – макроблок поля, а макроблок mbAddrN – макроблок кадра:

$$mvLXN[1] = mvLXN[1] / 2; \quad (8-214)$$

$$refIdxLXN = refIdxLXN * 2. \quad (8-215)$$

- Иначе, если текущий макроблок – макроблок кадра, а макроблок mbAddrN – макроблок поля:

$$mvLXN[1] = mvLXN[1] * 2; \quad (8-216)$$

$$\text{refIdxLXN} = \text{refIdxLXN} / 2. \quad (8-217)$$

- Иначе вертикальный компонент вектора движения $\text{mvLXN}[1]$ и индекс контроля refIdxLXN остаются неизменными.

8.4.1.4 Процесс вывода векторов движения цветности

Данный процесс активируется, только если chroma_format_idc не равно 0 (монохромное изображение).

Входы в этот процесс – вектор движения яркости mvLX и индекс контроля refIdxLX .

Выход из этого процесса – вектор движения цветности mvCLX .

Вектор движения цветности находят из соответствующего вектора движения яркости.

Точность компонентов вектора движения цветности определяется как $1 \div (4 * \text{SubWidthC})$ по горизонтали и как $1 \div (4 * \text{SubHeightC})$ по вертикали.

ПРИМЕЧАНИЕ. – Например, при использовании формата цветности 4:2:0, поскольку единицы векторов движения яркости составляют одну четвертую единиц образца яркости, а цветность имеет половинное разрешение по горизонтали и вертикали по сравнению с яркостью, единицы векторов движения цветности составляет одну восьмую единиц образцов цветности, т. е. значение 1 для вектора движения цветности означает смещение на одну восьмую образца цветности. Например, если вектор яркости применяют к образцам яркости 8x16, то соответствующий вектор цветности в формате цветности 4:2:0 применяют к образцам цветности 4x8, а если вектор яркости применяют к образцам яркости 4x4, то соответствующий вектор цветности в формате цветности 4:2:0 применяют к образцам цветности 2x2.

При отыскании вектора движения mvCLX используют следующие условия.

- Если chroma_format_idc не равно 1 или текущий макроблок – макроблок кадра, горизонтальный и вертикальный компоненты векторов движения цветности mvCLX находят следующим образом:

$$\text{mvCLX}[0] = \text{mvLX}[0]; \quad (8-218)$$

$$\text{mvCLX}[1] = \text{mvLX}[1]. \quad (8-219)$$

- Иначе (chroma_format_idc равно 1 и текущий макроблок – макроблок поля) находят только горизонтальный компонент вектора движения цветности $\text{mvCLX}[0]$, используя уравнение 8-218. Вертикальный компонент вектора движения цветности $\text{mvCLX}[1]$ зависит от четности текущего поля или текущего макроблока и контрольного изображения, которые учитывают индексом контроля refIdxLX . Значение $\text{mvCLX}[1]$ находят из $\text{mvLX}[1]$ согласно таблице 8-10.

Таблица 8-10 – Вывод вертикального компонента вектора цветности в режиме кодирования поля

Условия четности		$\text{mvCLX}[1]$
Контрольное изображение (refIdxLX)	Текущее поле (изображение/макроблок)	
Верхнее поле	Нижнее поле	$\text{mvLX}[1] + 2$
Нижнее поле	Верхнее поле	$\text{mvLX}[1] - 2$
Иначе		$\text{mvLX}[1]$

8.4.2 Процесс декодирования образцов Inter предсказания

Входы в этот процесс:

- часть макроблока mbPartIdx ;
- часть субмакроблока subMbPartIdx ;
- переменные, определяющие ширину и высоту частей для яркости и цветности (если имеются), partWidth , partHeight , partWidthC (если имеется) и partHeightC (если имеется);
- векторы движения яркости mvL0 и mvL1 и, если chroma_format_idc не равно 0 (монохромное изображение) векторы движения цветности mvCL0 и mvCL1 ;

- индексы контроля $refIdxL0$ и $refIdxL1$;
- флаги использованных списков предсказаний $predFlagL0$ и $predFlagL1$.

Выходы из этого процесса:

- образцы Inter предсказания $predPart$, которые являются массивом $predPart_L - (partWidth) \times (partHeight)$ – образцов яркости предсказания, и, если $chroma_format_idc$ не равно 0 (монохромное изображение), два массива $(partWidthC) \times (partHeightC) - predPart_{Cb}$, $predPart_{Cr}$ – образцов цветности предсказания, по одному для каждой компоненты цветности C_b и C_r .

Положим, что $predPartL0_L$ и $predPartL1_L$ – массивы $(partWidth) \times (partHeight)$ предсказанных значений образца яркости, и, если $chroma_format_idc$ не равно 0 (монохромное изображение), $predPartL0_{Cb}$, $predPartL1_{Cb}$, $predPartL0_{Cr}$ и $predPartL1_{Cr}$ – массивы $(partWidthC) \times (partHeightC)$ предсказанных значений образца цветности.

При замене LX на L0 или L1 в переменных $predFlagLX$, $RefPicListX$, $refIdxLX$, $refPicLX$, $predPartLX$ определяют следующее.

Если $predFlagLX$ равно 1, то используют следующие условия:

- контрольное изображение, состоящее из упорядоченного двумерного массива $refPicLX_L$ образцов яркости и, если $chroma_format_idc$ не равно 0 (монохромное изображение), двух упорядоченных двумерных массивов, $refPicLX_{Cb}$ и $refPicLX_{Cr}$, образцов цветности, находят, активируя процесс, определенный в п. 8.4.2.1, с $refIdxLX$ и $RefPicListX$, заданными как вход;
- массив $predPartLX_L$, и, если $chroma_format_idc$ не равно 0 (монохромное изображение), $predPartLX_{Cb}$ и $predPartLX_{Cr}$ находят, активируя процесс, определенный в п. 8.4.2.2, с текущей частью, определенной значениями $mbPartIdx$ \ $subMbPartIdx$, векторами движения $mvLX$, $mvCLX$ (если имеются) и контрольными массивами с $refPicLX_L$, $refPicLX_{Cb}$ и $refPicLX_{Cr}$ (если имеются), заданными как вход.

При замене C на L, C_b (если имеется) или C_r (если имеется) массив $predPart_C$ для образцов предсказания компоненты C находят, активируя процесс, определенный в п. 8.4.2.3, с текущей частью, определенной $mbPartIdx$ и $subMbPartIdx$ и массивами $predPartL0_C$ и $predPartL1_C$, а также $predFlagL0$ и $predFlagL1$, заданными как вход.

8.4.2.1 Процесс выбора контрольного изображения

Вход в этот процесс – индекс контроля $refIdxLX$.

Выход этого процесса – контрольное изображение, состоящее из двумерного массива образцов яркости $refPicLX_L$ и двух двумерных массивов образцов цветности $refPicLX_{Cb}$ и $refPicLX_{Cr}$.

В зависимости от $field_pic_flag$ список контрольных изображений $RefPicListX$ (который был получен, как это описано в п. 8.2.4) состоит из следующего:

- если $field_pic_flag$ равно 1, каждый вход $RefPicListX$ – это контрольное поле или поле контрольного кадра;
- иначе ($field_pic_flag$ равно 0) каждый вход $RefPicListX$ – это контрольное поле или пара дополнительных контрольных полей.

При отыскании контрольного изображения используют следующие условия.

- Если $field_pic_flag$ равно 1, контрольное поле или поле контрольного кадра, $PicNum = RefPicListX[refIdxLX]$ является выходом. Выходной контрольный кадр или поле контрольного кадра состоит из массива $(PicWidthInSamples_L) \times (PicHeightInSamples_L)$ образцов яркости $refPicLX_L$ и, если $chroma_format_idc$ не равно 0 (монохромное изображение), двух массивов $(PicWidthInSamples_C) \times (PicHeightInSamples_C)$ образцов цветности $refPicLX_{Cb}$ и $refPicLX_{Cr}$.
- Иначе ($field_pic_flag$ равно 0) используют следующие условия.
 - Если текущий макроблок – макроблок кадра, то контрольный кадр или пара дополнительных контрольных полей, $PicNum = RefPicListX[refIdxLX]$ является выходом. Выходной контрольный кадр или пара дополнительных контрольных полей состоит из массива $(PicWidthInSamples_L) \times (PicHeightInSamples_L)$ образцов яркости $refPicLX_L$ и, если $chroma_format_idc$ не равно 0 (монохромное изображение), двух массивов $(PicWidthInSamples_C) \times (PicHeightInSamples_C)$ образцов цветности $refPicLX_{Cb}$ и $refPicLX_{Cr}$.
 - Иначе (текущий макроблок – макроблок поля) используют следующие условия.
 - Положим, что $refFrame$ – контрольный кадр или пара дополнительных контрольных полей $RefPicListX[refIdxLX / 2]$.
 - Поле $refFrame$ выбирают следующим образом:
 - если $refIdxLX \% 2$ равно 0, поле $refFrame$, которое имеет ту же четность, что и у текущего макроблока, является выходом;
 - иначе ($refIdxLX \% 2$ равно 1) поле $refFrame$, которое имеет четность, противоположную четности текущего макроблока, является выходом.

- Выходное контрольное поля или поле контрольного кадра состоит из массива $(PicWidthInSamples_L) \times (PicHeightInSamples_L/2)$ образцов яркости $refPicLX_L$ и, если $chroma_format_idc$ не равно 0 (монохромное изображение), двух массивов $(PicWidthInSamples_C) \times (PicHeightInSamples_C/2)$ образцов цветности $refPicLX_{Cb}$ и $refPicLX_{Cr}$.

Массивы образцов контрольных изображений $refPicLX_L$, $refPicLX_{Cb}$ (если имеются) и $refPicLX_{Cr}$ (если имеются) соответствуют массивам декодированных образцов S_L , S_{Cb} (если имеются), S_{Cr} (если имеются), полученных в п. 8.7 для ранее декодированного контрольного поля или контрольного кадра либо пары дополнительных контрольных полей или поля контрольного кадра.

8.4.4.2 Процесс интерполяции фрагментарного образца

Входы в этот процесс:

- текущее разделение на части, заданное индексом разделения $mbPartIdx$ и субиндексом разделенной части макроблока $subMbPartIdx$;
- ширина и высота – $partWidth$, $partHeight$ – этой части в единицах от образца яркости;
- вектор движения яркости $mvLX$, заданный в одной четвертой единицы от образца яркости;
- вектор движения цветности $mvCLX$, заданный в одной восьмой единицы от образца цветности; и
- выбранные массивы образцов контрольных изображений $refPicLX_L$, $refPicLX_{Cb}$ и $refPicLX_{Cr}$.

Выходы из этого процесса:

- массив $(partWidth) \times (partHeight)$, $predPartLX_L$, значения предсказания образца яркости; и
- если $chroma_format_idc$ не равно 0 (монохромное изображение), два массива $(partWidthC) \times (partHeightC)$ – $predPartLX_{Cb}$ и $predPartLX_{Cr}$ – значений предсказания образца цветности.

Положим, что (x_{A_L}, y_{A_L}) – местоположение, заданное в полных единицах образца для верхнего левого образца яркости текущего разделения, которое определено значением $mbPartIdx \setminus subMbPartIdx$ относительно верхнего левого местоположения образца яркости заданного двумерного массива образца яркости.

Положим, что (x_{Int_L}, y_{Int_L}) – местоположение, заданное в полных единицах образца яркости, а (x_{Frac_L}, y_{Frac_L}) – сдвиг, заданный в единицах четверти образца. Эти переменные используют только в этом пункте для определения общего расположения фракций образца внутри массивов контрольного образца $refPicLX_L$, $refPicLX_{Cb}$ (если имеются) и $refPicLX_{Cr}$ (если имеются).

Для каждого расположения образца яркости ($0 \leq x_L < partWidth$, $0 \leq y_L < partHeight$) внутри массива предсказанного образца яркости $predPartLX_L$ значение соответствующего предсказания образца яркости $predPartLX_L[x_L, y_L]$ находят следующим образом.

- Переменные x_{Int_L} , y_{Int_L} , x_{Frac_L} и y_{Frac_L} находят следующим образом:

$$x_{Int_L} = x_{A_L} + (mvLX[0] \gg 2) + x_L; \quad (8-220)$$

$$y_{Int_L} = y_{A_L} + (mvLX[1] \gg 2) + y_L; \quad (8-221)$$

$$x_{Frac_L} = mvLX[0] \& 3; \quad (8-222)$$

$$y_{Frac_L} = mvLX[1] \& 3. \quad (8-223)$$

- Значение образца яркости предсказания $predPartLX_L[x_L, y_L]$ находят, активируя процесс, определенный в п. 8.4.2.2.1, с (x_{Int_L}, y_{Int_L}) , (x_{Frac_L}, y_{Frac_L}) и $refPicLX_L$, заданными как вход.

Если $chroma_format_idc$ не равно 0 (монохромное изображение), применяется следующее.

Положим, что (x_{Int_C}, y_{Int_C}) – местоположение, заданное в полных единицах образца цветности, а (x_{Frac_C}, y_{Frac_C}) – сдвиг, заданный в одной восьмой единицы образца. Эти переменные используют только в этом пункте для определения общего расположения фракций образца внутри массивов контрольного образца $refPicLX_{Cb}$ и $refPicLX_{Cr}$.

Для каждого местоположения образца цветности ($0 \leq x_C < partWidthC$, $0 \leq y_C < partHeightC$) внутри массивов образцов цветности предсказания $predPartLX_{Cb}$ и $predPartLX_{Cr}$ соответствующие значения предсказанных образцов цветности $predPartLX_{Cb}[x_C, y_C]$ и $predPartLX_{Cr}[x_C, y_C]$ находят следующим образом.

- В зависимости от $chroma_format_idc$, переменные x_{Int_C} , y_{Int_C} , x_{Frac_C} и y_{Frac_C} находят следующим образом.
 - Если $chroma_format_idc$ не равно 1:

$$xInt_C = (xA_L / SubWidthC) + (mvCLX[0] \gg 3) + x_C; \quad (8-224)$$

$$yInt_C = (yA_L / SubHeightC) + (mvCLX[1] \gg 3) + y_C; \quad (8-225)$$

$$xFrac_C = mvCLX[0] \& 7; \quad (8-226)$$

$$yFrac_C = mvCLX[1] \& 7. \quad (8-227)$$

- Иначе, если chroma_format_idc равно 2:

$$xInt_C = (xA_L / SubWidthC) + (mvCLX[0] \gg 3) + x_C; \quad (8-228)$$

$$yInt_C = (yA_L / SubHeightC) + (mvCLX[1] \gg 2) + y_C; \quad (8-229)$$

$$xFrac_C = mvCLX[0] \& 7; \quad (8-230)$$

$$yFrac_C = (mvCLX[1] \& 3) \ll 1. \quad (8-231)$$

- Иначе (chroma_format_idc равно 3):

$$xInt_C = (xA_L / SubWidthC) + (mvCLX[0] \gg 2) + x_C; \quad (8-232)$$

$$yInt_C = (yA_L / SubHeightC) + (mvCLX[1] \gg 2) + y_C; \quad (8-233)$$

$$xFrac_C = (mvCLX[0] \& 3) \ll 1; \quad (8-234)$$

$$yFrac_C = (mvCLX[1] \& 3) \ll 1. \quad (8-235)$$

- Значение образца предсказания $predPartLX_{Cb}[x_C, y_C]$ находят, активируя процесс, определенный в п. 8.4.2.2.2, с $(xInt_C, yInt_C)$, $(xFrac_C, yFrac_C)$ и $refPicLX_{Cb}$, заданными как вход.
- Значение образца предсказания $predPartLX_{Cr}[x_C, y_C]$ находят, активируя процесс, определенный в п. 8.4.2.2.2, с $(xInt_C, yInt_C)$, $(xFrac_C, yFrac_C)$ и $refPicLX_{Cr}$, заданными как вход.

8.4.2.2.1 Процесс интерполяции образца яркости

Входы в этот процесс:

- местоположение яркости в полных единицах образца $(xInt_L, yInt_L)$;
- местоположение сдвига яркости в единицах фракций образца $(xFrac_L, yFrac_L)$; и
- массив образца яркости выбранного контрольного изображения $refPicLX_L$.

Выход этого процесса – предсказанное значение образца яркости $predPartLX_L[x_L, y_L]$.

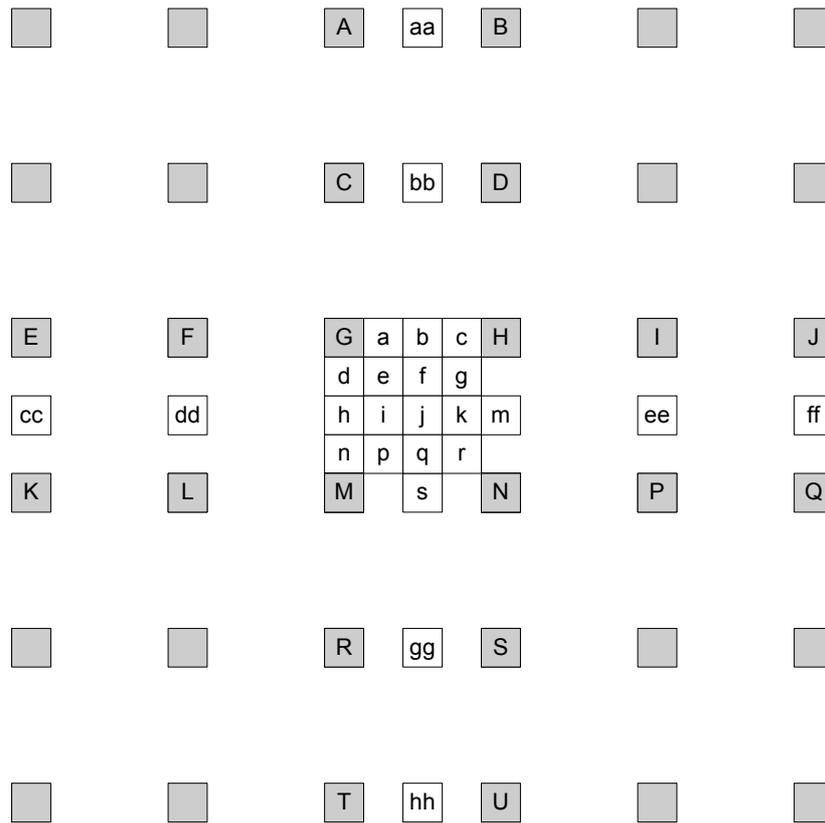


Рисунок 8-4 – Расположение целых образцов (затененные блоки с заглавными буквами) и фракций образцов (незатененные блоки со строчными буквами) для интерполяции четверти образца яркости

Переменную $refPicHeightEffective_L$, которая представляет высоту действующего массива контрольных изображений яркости, находят следующим образом:

- если $MbaffFrameFlag$ равно 0 или $mb_field_decoding_flag$ равно 0, $refPicHeightEffectiveL$ устанавливают равным $PicHeightInSamplesL$;
- иначе ($MbaffFrameFlag$ равно 1 и $mb_field_decoding_flag$ равно 1) $refPicHeightEffectiveL$ устанавливают равным $PicHeightInSamplesL / 2$.

На рисунке 8-4 местоположения, обозначенные заглавными буквами в затененных блоках, представляют местоположения образцов яркости в полном образце внутри заданного двумерного массива $refPicLX_L$ образцов яркости. Эти образцы могут быть использованы для создания предсказаний значения образца яркости $predPartLX_L[x_L, y_L]$. Местоположения (x_{ZL}, y_{ZL}) для каждого из соответствующих образцов яркости Z (где Z может быть A, B, C, D, E, F, G, H, I, J, K, L, M, N, P, Q, R, S, T или U внутри заданного массива $refPicLX_L$ образцов яркости) находят следующим образом:

$$\begin{aligned} x_{ZL} &= Clip3(0, PicWidthInSamplesL - 1, x_{IntL} + x_{DZL}); \\ y_{ZL} &= Clip3(0, refPicHeightEffectiveL - 1, y_{IntL} + y_{DZL}). \end{aligned} \quad (8-236)$$

Таблица 8-11 определяет значения (x_{DZL}, y_{DZL}) для различных расположений Z .

Таблица 8-11 – Различное расположение полного образца яркости

Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R	S	T	U
x_{DZL}	0	1	0	1	-2	-1	0	1	2	3	-2	-1	0	1	2	3	0	1	0	1
y_{DZL}	-2	-2	-1	-1	0	0	0	0	0	0	1	1	1	1	1	1	2	2	3	3

При заданных образцах яркости от "А" до "U" в местоположениях полных образцов от (x_{A_L}, y_{A_L}) до (x_{U_L}, y_{U_L}) образцы яркости от "a" до "s" в местоположениях фракций образцов местоположения находят по следующим правилам. Значения предсказания яркости для половины местоположений образцов получают с применением фильтра с 6-ю отводами со значениями отводов $(1, -5, 20, 20, -5, 1)$. Значения предсказания яркости для местоположения четверти образца находятся путем усреднения местоположений полных образцов и половин образцов. Процесс для каждого местоположения фракции описан ниже.

- Образцы в местоположениях половины образца, обозначенные как b , находят при первом вычислении промежуточного значения, обозначенного как b_1 , присоединением 6-отводного фильтра к ближайшему целочисленному местоположению образцов в горизонтальном направлении. Образцы в местоположениях половины образца, обозначенные как h , находят при первом вычислении промежуточного значения, обозначенного как h_1 , присоединением 6-отводного фильтра к ближайшему целочисленному местоположению образцов в вертикальном направлении:

$$b_1 = (E - 5 * F + 20 * G + 20 * H - 5 * I + J); \quad (8-237)$$

$$h_1 = (A - 5 * C + 20 * G + 20 * M - 5 * R + T). \quad (8-238)$$

Окончательное значение предсказания b и h получают используя уравнения:

$$b = \text{Clip1}((b_1 + 16) \gg 5); \quad (8-239)$$

$$h = \text{Clip1}((h_1 + 16) \gg 5). \quad (8-240)$$

- Образцы в местоположениях половины образца, обозначенные как j , находят при первом вычислении промежуточного значения, обозначенного как j_1 , присоединением 6-отводного фильтра к ближайшему целочисленному местоположению образцов либо в горизонтальном, либо в вертикальном направлении, поскольку это даст одинаковый результат.

$$j_1 = cc - 5 * dd + 20 * h_1 + 20 * m_1 - 5 * ee + ff; \text{ или} \quad (8-241)$$

$$j_1 = aa - 5 * bb + 20 * b_1 + 20 * s_1 - 5 * gg + hh, \quad (8-242)$$

где предварительные значения, обозначенные как aa , bb , gg , s_1 и hh , получают с применением фильтра с 6-ю отводами горизонтально тем же способом, что и нахождение b_1 . Предварительные значения, обозначенные как cc , dd , ee , m_1 и ff , находят с применением фильтра с 6-ю отводами вертикально тем же способом, что и нахождение h_1 . Окончательное значение предсказания j получают, используя уравнение:

$$j = \text{Clip1}_Y((j_1 + 512) \gg 10). \quad (8-243)$$

Окончательные значения предсказания s и m находят из s_1 и m_1 тем же способом, что и b и h , как:

$$s = \text{Clip1}((s_1 + 16) \gg 5); \quad (8-244)$$

$$m = \text{Clip1}((m_1 + 16) \gg 5). \quad (8-245)$$

- Образцы в местоположениях четверти образца, обозначенные как a , c , d , n , f , i , k и q , находят усреднением с округлением в большую сторону двух ближайших образцов в местоположениях целых и половин образцов, используя уравнения:

$$a = (G + b + 1) \gg 1; \quad (8-246)$$

$$c = (H + b + 1) \gg 1; \quad (8-247)$$

$$d = (G + h + 1) \gg 1; \quad (8-248)$$

$$n = (M + h + 1) \gg 1; \quad (8-249)$$

$$f = (b + j + 1) \gg 1; \quad (8-250)$$

$$i = (h + j + 1) \gg 1; \quad (8-251)$$

$$k = (j + m + 1) \gg 1; \quad (8-252)$$

$$q = (j + s + 1) \gg 1. \quad (8-253)$$

- Образцы в местоположениях четверти образца, обозначенные как e , g , r и t , получают усреднением с округлением в большую сторону двух ближайших образцов в местоположениях половин образцов в диагональном направлении, используя уравнения:

$$e = (b + h + 1) \gg 1; \quad (8-254)$$

$$g = (b + m + 1) \gg 1; \quad (8-255)$$

$$p = (h + s + 1) \gg 1; \quad (8-256)$$

$$r = (m + s + 1) \gg 1. \quad (8-257)$$

Сдвиг расположения яркости фракции в единицах образца ($xFrac_L, yFrac_L$) определяет, какой из созданных образцов яркости в местоположениях полного образца и фракции образца присвоен значению предсказанного образца яркости $predPartLX_L[x_L, y_L]$. Это присвоение выполняют согласно таблице 8-12. Значение $predPartLX_L[x_L, y_L]$ является выходом.

Таблица 8-12 – Присвоение предсказанного образца яркости $predPartLX_L[x_L, y_L]$

$xFrac_L$	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
$yFrac_L$	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
$predPartLX_L[x_L, y_L]$	G	D	h	n	a	e	i	p	b	f	j	q	c	g	k	r

8.4.2.2.2 Процесс интерполяции образца цветности

Этот процесс активируется, только если $chroma_format_idc$ не равно 0 (монохромное изображение).

Входы в этот процесс:

- расположение цветности в единицах полного образца ($xInt_C, yInt_C$);
- сдвиг расположения цветности в единицах фракции образца ($xFrac_C, yFrac_C$); и
- компоненты образцов цветности из выбранного контрольного изображения $refPicLX_C$.

Выход этого процесса – значение предсказанного образца цветности $predPartLX_C[x_C, y_C]$. На рисунке 8-5 местоположения, обозначенные как А, В, С и D, представляют образцы цветности в расположении полного образца внутри заданного двумерного массива $refPicLX_C$ образцов цветности.

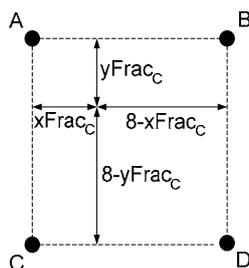


Рисунок 8-5 – Расположение фракций образца в зависимости от переменных в интерполяции цветности и окружающих целочисленных местоположений образцов А, В, С и D

Переменную $refPicHeightEffective_C$, которая выражает высоту действующего массива контрольных изображений цветности, находят следующим образом.

- Если $MbaffFrameFlag$ равно 0 или $mb_field_decoding_flag$ равно 0, то $refPicHeightEffective_C$ устанавливают равным $PicHeightInSamples_C$.
- Иначе ($MbaffFrameFlag$ равно 1 и $mb_field_dicoding_flag$ равно 1) $refPicHeightEffective_C$ устанавливают равным $PicHeightInSamples_C / 2$.

Координаты образца, определенные в уравнениях 8-258–8-265, используют для создания предсказанного значения образца цветности $predPartLX_C[x_C, y_C]$.

$$xA_C = Clip3(0, PicWidthInSamples_C - 1, xInt_C); \quad (8-258)$$

$$xB_C = Clip3(0, PicWidthInSamples_C - 1, xInt_C + 1); \quad (8-259)$$

$$xC_C = Clip3(0, PicWidthInSamples_C - 1, xInt_C); \quad (8-260)$$

$$xD_C = Clip3(0, PicWidthInSamples_C - 1, xInt_C + 1); \quad (8-261)$$

$$yA_C = Clip3(0, refPicHeightEffective_C - 1, yInt_C); \quad (8-262)$$

$$yB_C = Clip3(0, refPicHeightEffective_C - 1, yInt_C); \quad (8-263)$$

$$yC_C = Clip3(0, refPicHeightEffective_C - 1, yInt_C + 1); \quad (8-264)$$

$$yD_C = Clip3(0, refPicHeightEffective_C - 1, yInt_C + 1); \quad (8-265)$$

При заданных образцах цветности A, B, C и D в местоположениях полных образцов, определенных уравнениями от 8-258 до 8-265, значение предсказанного образца цветности $\text{predPartLXc}[x_c, y_c]$ находят следующим образом:

$$\text{predPartLXc}[x_c, y_c] = \left(\frac{(8 - x_{\text{Frac}_C}) * (8 - y_{\text{Frac}_C}) * A + x_{\text{Frac}_C} * (8 - y_{\text{Frac}_C}) * B + (8 - x_{\text{Frac}_C}) * y_{\text{Frac}_C} * C + x_{\text{Frac}_C} * y_{\text{Frac}_C} * D}{32} \right) \gg 6. \quad (8-266)$$

8.4.2.3 Процесс взвешенного предсказания образца

Входы в этот процесс:

- mbPartIdx : текущее разделение, заданное индексом разделения;
- subMbPartIdx : субиндекс разделенной части макроблока;
- predFlagL0 и predFlagL1 : флаги использованных списков предсказаний;
- predPartLXL : массив $(\text{partWidth}) \times (\text{partHeight})$ предсказанных образцов яркости (с заменой LX на L0 или L1 в зависимости от predFlagL0 и predFlagL1);
- если chroma_format_idc не равно 0 (монохромное изображение), predPartLXCb и predPartLXCc : массивы $(\text{partWidthC}) \times (\text{partHeightC})$ образцов цветности предсказания, по одному для каждого из компонентов цветности Cb и Cr (с заменой LX на L0 или L1 в зависимости от predFlagL0 и predFlagL1).

Выходы из этого процесса:

- predPartL : массив $(\text{partWidth}) \times (\text{partHeight})$ предсказанных образцов яркости; и
- если chroma_format_idc не равно 0 (монохромное изображение), predPartCb и predPartCr : массивы $(\text{partWidthC}) \times (\text{partHeightC})$ образцов цветности предсказания, по одному для каждого из компонентов цветности Cb и Cr.

Для макроблоков или разделенных частей с predFlagL0 , равным 1, в секциях P и SP используют следующие условия.

- Если $\text{weighted_pred_flag}$ равно 0, процесс взвешенного по умолчанию предсказания образца, как описано в п. 8.4.2.3.1, активируют с теми же входами и выходами, что и в описанном в этом пункте процессе.
- Иначе ($\text{weighted_pred_flag}$ равно 1) подробный взвешенный процесс предсказания, как описано в п. 8.4.2.3.2, активируют с теми же входами и выходами, что и в описанном в этом пункте процессе.

Для макроблоков или разделенных частей с predFlagL0 или predFlagL1 , равными 1, в секциях B используют следующие условия.

- Если $\text{weighted_bipred_idc}$ равно 0, процесс взвешенного по умолчанию предсказания образца, как описано в п. 8.4.2.3.1, активируют с теми же входами и выходами, что и в описанном в этом пункте процессе.
- Иначе, если $\text{weighted_bipred_idc}$ равно 1, подробный взвешенный процесс предсказания, как описано в п. 8.4.2.3.2, для макроблоков или разделенных частей с predFlagL0 или predFlagL1 , равными 1, активируют с теми же входами и выходами, что и в описанном в этом пункте процессе.
- Иначе ($\text{weighted_bipred_idc}$ равно 2) используют следующие условия:
 - если predFlagL0 равно 1 и predFlagL1 равно 1, неявный взвешенный образец предсказания, как описано в п. 8.4.2.3.2, активируют с теми же входами и выходами, что и в описанном в этом пункте процессе;
 - иначе (predFlagL0 или predFlagL1 равны 1, но не оба вместе) процесс взвешенного по умолчанию предсказания образца, как описано в п. 8.4.2.3.1, активируют с теми же входами и выходами, что и в описанном в этом пункте процессе.

8.4.2.3.1 Процесс взвешенного по умолчанию предсказания образца

Входы в этот процесс – те же, что и в п. 8.4.2.3.

Выходы этого процесса – те же, что и в п. 8.4.2.3.

В зависимости от доступного компонента, для которого находят блок предсказания, используют следующие условия.

- Если предсказанные значения образца яркости $\text{predPart}_l[x, y]$ найдены, используют следующие условия: значение C установлено равным L, x установлено равным $0 \dots \text{partWidth} - 1$, а y установлено равным $0 \dots \text{partHeight} - 1$.
- Иначе, если предсказанные значения компонента образца цветности Cb $\text{predPart}_{cb}[x, y]$ найдены, используют следующие условия: C установлено равным Cb, x установлено равным $0 \dots \text{partWidthC} - 1$ и y установлено равным $0 \dots \text{partHeightC} - 1$.

- Иначе (предсказанные значения компонента образца цветности Cr $\text{predPart}_{Cr}[x, y]$ найдены) используют следующие условия: C установлено равным Cr, x установлено равным $0 \dots \text{partWidthC} - 1$ и y установлено равным $0 \dots \text{partHeightC} - 1$.

Предсказанное значение образца находят следующим образом.

- Если predFlagL0 равно 1 и predFlagL1 равно 0, то для текущего разделения:

$$\text{predPart}_c[x, y] = \text{predPartL0}_c[x, y]. \quad (8-267)$$

- Иначе, если predFlagL0 равно 0 и predFlagL1 равно 1, то для текущего разделения:

$$\text{predPart}_c[x, y] = \text{predPartL1}_c[x, y]. \quad (8-268)$$

- Иначе (predFlagL0 и predFlagL1 равны 1 для текущего разделения):

$$\text{predPart}_c[x, y] = (\text{predPartL0}_c[x, y] + \text{predPartL1}_c[x, y] + 1) \gg 1. \quad (8-269)$$

8.4.2.3.2 Процесс взвешенного предсказания образца

Входы в этот процесс – те же, что и в п. 8.4.2.3.

Выходы этого процесса – те же, что и в п. 8.4.2.3.

В зависимости от доступного компонента, для которого находят блок предсказания, используют следующие условия.

- Если предсказанные значения образца яркости $\text{predPart}_L[x, y]$ найдены, используют следующие условия: C установлено равным L, x установлено равным $0 \dots \text{partWidth} - 1$ и y установлено равным $0 \dots \text{partHeight} - 1$.
- Иначе, если предсказанные значения компонента образца цветности Cb $\text{predPart}_{Cb}[x, y]$ найдены, используют следующие условия: C установлено равным Cb, x установлено равным $0 \dots \text{partWidthC} - 1$, а y установлено равным $0 \dots \text{partHeightC} - 1$.
- Иначе (предсказанные значения компонента образца цветности Cr $\text{predPart}_{Cr}[x, y]$ найдены) используют следующие условия: C установлено равным Cr, x установлено равным $0 \dots \text{partWidthC} - 1$, а y установлено равным $0 \dots \text{partHeightC} - 1$.

Предсказанные значения образца находят следующим образом.

- Если разделение $\text{mbPartIdx} \setminus \text{subMbPartIdx}$ имеет predFlagL0 , равное 1, а predFlagL1 равно 0, то окончательные предсказанные значения образца $\text{predPart}_c[x, y]$ находят как:

$$\begin{aligned} &\text{if}(\log\text{WD} \geq 1) \\ &\quad \text{predPart}_c[x, y] = \text{Clip1}_c(((\text{predPartL0}_c[x, y] * w_0 + 2^{\log\text{WD}-1}) \gg \log\text{WD}) + o_0) \\ &\text{else} \\ &\quad \text{predPart}_c[x, y] = \text{Clip1}_c(\text{predPartL0}_c[x, y] * w_0 + o_0). \end{aligned} \quad (8-270)$$

- Иначе, если разделение $\text{mbPartIdx} \setminus \text{subMbPartIdx}$ имеет predFlagL0 , равное 0, а predFlagL1 равно 1, то окончательные предсказанные значения образца $\text{predPart}_c[x, y]$ находят как:

$$\begin{aligned} &\text{if}(\log\text{WD} \geq 1) \\ &\quad \text{predPart}_c[x, y] = \text{Clip1}_c(((\text{predPartL1}_c[x, y] * w_1 + 2^{\log\text{WD}-1}) \gg \log\text{WD}) + o_1) \\ &\text{else} \\ &\quad \text{predPart}_c[x, y] = \text{Clip1}_c(\text{predPartL1}_c[x, y] * w_1 + o_1). \end{aligned} \quad (8-271)$$

- Иначе (разделение $\text{mbPartIdx} \setminus \text{subMbPartIdx}$ имеет оба значения, predFlagL0 и predFlagL1 , равными 1) окончательные предсказанные значения образца $\text{predPart}_c[x, y]$ находят как:

$$\text{predPart}_c[x, y] = \text{Clip1}_c(((\text{predPartL0}_c[x, y] * w_0 + \text{predPartL1}_c[x, y] * w_1 + 2^{\log\text{WD}}) \gg (\log\text{WD} + 1)) + ((o_0 + o_1 + 1) \gg 1)). \quad (8-272)$$

Переменные в вышеприведенных соотношениях для предсказанных образцов находят следующим образом.

- Если $\text{weighted_bipred_idc}$ равно 2, а slice_type равно B, то неявный режим взвешенного предсказания используют следующим образом:

$$\log WD = 5; \quad (8-273)$$

$$o_0 = 0; \quad (8-274)$$

$$o_1 = 0, \quad (8-275)$$

а w_0 и w_1 находят следующим образом:

- Переменные `currPicOrField`, `pic0` и `pic1` находят следующим образом.
 - Если `field_pic_flag` равно 0, а текущий макроблок – макроблок поля, используют следующие условия.
 - `currPicOrField` – это поле текущего изображения `CurrPic`, которое имеет ту же четность, что и у текущего макроблока.
 - Переменную `pic0` находят следующим образом:
 - если `refIdxL0 % 2` равно 0, `pic0` – поле `RefPicList0[refIdxL0 / 2]`, которое имеет ту же четность, что и у текущего макроблока;
 - иначе (`refIdxL0 % 2` не равно 0) `pic0` – поле `RefPicList1[refIdxL1 / 2]`, которое имеет четность, противоположную четности текущего макроблока.
 - Переменную `pic1` находят следующим образом.
 - если `refIdxL1 % 2` равно 0, `pic1` – поле `RefPicList1[refIdxL1 / 2]`, которое имеет ту же четность, что и у текущего макроблока;
 - иначе (`refIdxL1 % 2` не равно 0) `pic1` – поле `RefPicList1[refIdxL1 / 2]`, которое имеет четность, противоположную четности текущего макроблока.
 - Иначе (`field_pic_flag` равно 1 или текущий макроблок – макроблок кадра) `currPicOrField` – текущее изображение `CurrPic`, `pic1` – `RefPicList1[refIdxL1]`, а `pic0` – `RefPicList0[refIdxL0]`.
- Переменные `tb`, `td`, `tx` и `DistScaleFactor` находят из значений `currPicOrField`, `pic0`, `pic1`, используя уравнения 8-198, 8-199, 8-194 и 8-195, соответственно.
- Если `DiffPicOrderCnt(pic1, pic0)` равно 0 или если одно или оба значения `pic1` и `pic0` помечены как "используется для долгосрочного контроля" или $(DistScaleFactor \gg 2) < -64$ или $(DistScaleFactor \gg 2) > 128$, то w_0 и w_1 находят как:

$$w_0 = 32; \quad (8-276)$$

$$w_1 = 32. \quad (8-277)$$

– Иначе:

$$w_0 = 64 - (DistScaleFactor \gg 2); \quad (8-278)$$

$$w_1 = DistScaleFactor \gg 2. \quad (8-279)$$

- Иначе (`weighted_pred_flag` равно 1 в секциях P или SP или `weighted_bipred_idc` равно 1 в секциях B) явный режим взвешенного предсказания используют следующим образом.
 - Переменные `refIdxL0WP` и `refIdxL1WP` находят следующим образом.
 - Если `MbaffFrameFlag` равно 1, а текущий макроблок – макроблок поля:

$$refIdxL0WP = refIdxL0 \gg 1; \quad (8-280)$$

$$refIdxL1WP = refIdxL1 \gg 1. \quad (8-281)$$
 - Иначе (`MbaffFrameFlag` равно 0 или текущий макроблок – макроблок кадра):

$$refIdxL0WP = refIdxL0; \quad (8-282)$$

$$\text{refIdxL1WP} = \text{refIdxL1}. \quad (8-283)$$

- Переменные $\log\text{WD}$, w_0 , w_1 , o_0 и o_1 находят следующим образом.
- Если C в $\text{predPart}_C[x, y]$ заменяют на L для образцов яркости:

$$\log\text{WD} = \text{luma_log2_weight_denom}; \quad (8-284)$$

$$w_0 = \text{luma_weight_l0}[\text{refIdxL0WP}]; \quad (8-285)$$

$$w_1 = \text{luma_weight_l1}[\text{refIdxL1WP}]; \quad (8-286)$$

$$o_0 = \text{luma_offset_l0}[\text{refIdxL0WP}] * (1 \ll (\text{BitDepth}_Y - 8)); \quad (8-287)$$

$$o_1 = \text{luma_offset_l1}[\text{refIdxL1WP}] * (1 \ll (\text{BitDepth}_Y - 8)). \quad (8-288)$$

- Иначе (C в $\text{predPart}_C[x, y]$ заменяют на C_b или C_r для образцов цветности, с $iCbCr = 0$ для C_b , $iCbCr = 1$ для C_r):

$$\log\text{WD} = \text{chroma_log2_weight_denom}; \quad (8-289)$$

$$w_0 = \text{chroma_weight_l0}[\text{refIdxL0WP}][iCbCr]; \quad (8-290)$$

$$w_1 = \text{chroma_weight_l1}[\text{refIdxL1WP}][iCbCr]; \quad (8-291)$$

$$o_0 = \text{chroma_offset_l0}[\text{refIdxL0WP}][iCbCr] * (1 \ll (\text{BitDepth}_C - 8)); \quad (8-292)$$

$$o_1 = \text{chroma_offset_l1}[\text{refIdxL1WP}][iCbCr] * (1 \ll (\text{BitDepth}_C - 8)). \quad (8-293)$$

Если используют явный режим взвешенного предсказания, а разделение $\text{mbPartIdx} \setminus \text{subMbPartIdx}$ имеет оба значения, predFlagL0 и predFlagL1 , равные 1, то должно действовать следующее ограничение:

$$-128 \leq w_0 + w_1 \leq ((\log\text{WD} == 7) ? 127 : 128). \quad (8-294)$$

ПРИМЕЧАНИЕ. – Для явного режима взвешенного предсказания каждый вес, w_0 и w_1 , гарантировано находится в диапазоне $-64..128$, а ограничение, выраженное в уравнении 8-294, хотя это явно и не указано, всегда будет выполняться. Для явного режима взвешенного предсказания с $\log\text{WD}$, равным 7, если один из двух весов, w_0 или w_1 , считают равным 128 (поскольку последовательность $\text{luma_weight_l0_flag}$, $\text{luma_weight_l1_flag}$, $\text{chroma_weight_l0_flag}$, или $\text{chroma_weight_l1_flag}$, равна 0), другой вес (w_1 или w_0) должен иметь отрицательное значение в порядке ограничения, наложенного на уравнение 8-294 (и поэтому другое значение флага $\text{luma_weight_l0_flag}$, $\text{luma_weight_l1_flag}$, $\text{chroma_weight_l0_flag}$ или $\text{chroma_weight_l1_flag}$ должно быть равно 1).

8.5 Процесс декодирования коэффициента преобразования и процесс построения изображения до фильтра устранения блочности

Входы в этот процесс – Intra16x16DCLevel (если имеется), Intra16x16ACLevel (если имеется), LumaLevel (если имеется), LumaLevel8x8 (если имеется), ChromaDCLevel (если имеется), ChromaACLevel (если имеется) и доступные массивы образцов Inter или Intra предсказаний текущего макроблока с доступными компонентами pred_L , pred_{Cb} или pred_{Cr} .

ПРИМЕЧАНИЕ 1. – При декодировании макроблока в режиме предсказания Intra_4x4 (или Intra_8x8) массив предсказания компонентов яркости макроблока может быть неполным, поскольку для каждого блока яркости 4×4 (8×8) процесс предсказания Intra_4x4 (или Intra_8x8) для образцов яркости, как это определено в п. 8.3.1 (или п. 8.3.2), и процесс, определенный в этом пункте, являются итеративными.

Выходы из этого процесса – построенные массивы образцов до процесса фильтра устранения блочности соответствующих компонентов S'_L , S'_{Cb} или S'_{Cr} .

ПРИМЕЧАНИЕ 2. – При декодировании макроблока в режиме предсказания Intra_4x4 (или Intra_8x8) компонент яркости массивов образцов макроблоков, построенных до процесса фильтра устранения блочности, может быть неполным, поскольку для каждого блока яркости 4×4 (или 8×8) процесс предсказания Intra_4x4 (или Intra_8x8) для

образцов яркости, как это определено в п. 8.3.1 (или п. 8.3.2), и процесс, определенный в этом пункте, являются итеративными.

В этом пункте определяют декодирование коэффициента преобразования и построение изображения перед фильтровым разделением на блоки.

Если текущий макроблок кодирован как P_Skip или B_Skip, то для текущего макроблока все значения LumaLevel, LumaLevel8x8, ChromaDCLevel, ChromaACLevel устанавливают равными 0.

Если residual_colour_transform_flag установлено равным 1, активируется процесс остаточного преобразования цвета, описанный в п. 8.5.13.

8.5.1 Описание процесса декодирования преобразования остаточных блоков яркости 4x4

Данное описание применяется, если transform_size_8x8_flag равно 0.

Если режим предсказания текущего макроблока не равен Intra_16x16, переменная LumaLevel содержит уровни для коэффициентов преобразования яркости. Для блока яркости 4x4, индексированного luma4x4BlkIdx = 0..15, определены следующие упорядоченные шаги:

1. Активируют процесс инверсного сканирования коэффициентов преобразования, который описан в п. 8.5.5, с LumaLevel[luma4x4BlkIdx] в качестве входа и двумерным массивом c в качестве выхода.
2. Активируют процесс масштабирования и преобразования остаточных блоков 4x4, как это определено в п. 8.5.8, с параметрами c в качестве входа и g в качестве выхода.
3. Если residual_colour_transform_flag равно 1, переменная $R_{y,ij}$ устанавливается равной r_{ij} с $i, j = 0..3$, и процесс приостанавливается до завершения процесса остаточного преобразования цвета, описание которого содержится в п. 8.5.13, а по его завершении переменная r_{ij} устанавливается равной $R_{G,ij}$ с $i, j = 0..3$, и продолжается выполнение данного процесса.
4. Положение верхнего левого образца блока яркости 4x4 с индексом luma4x4BlkIdx внутри макроблока находят, активируя процесс инверсного сканирования блока яркости 4x4, который описан п. 6.4.3, с luma4x4BlkIdx в качестве входа и выходом, которому присваивается (xO, yO).
5. Массив u 4x4 с элементами u_{ij} для $i, j = 0..3$ находят как:

$$u_{ij} = \text{Clip1}(\text{pred}_L[xO + j, yO + i] + r_{ij}). \quad (8-295)$$

Если qpprime_y_zero_transform_bypass_flag равно 1, а QP'_Y равно 0, поток битов не должен содержать данных, в результате которых значение u_{ij} , вычисляемое по уравнению 8-295, окажется не равным $\text{pred}_L[xO + j, yO + i] + r_{ij}$.

6. Активируют процесс построения изображения до процесса фильтра устранения блочности, который описан в п. 8.5.12, с luma4x4BlkIdx и u в качестве входа.

8.5.2 Описание процесса декодирования преобразования для образцов яркости режима предсказания макроблока Intra_16x16

Если режим предсказания текущего макроблока равен Intra_16x16, переменные Intra16x16DCLevel и Intra16x16ACLevel содержат уровни коэффициентов преобразования яркости. Декодирование коэффициентов преобразования производят следующими ступенями.

1. Декодируют DC коэффициенты преобразования яркости 4x4 всех блоков яркости 4x4 макроблока.
 - a. Активируют процесс инверсного сканирования коэффициентов преобразования, как описано в п. 8.5.5, с Intra16x16DCLevel в качестве входа и двумерным массивом c в качестве выхода.
 - b. Активируют процесс масштабирования и преобразования DC коэффициентов преобразования яркости для типа макроблока Intra_16x16, как определено в п. 8.5.8, со значением c в качестве входа и 'dcY' в качестве выхода.
2. Для блока яркости 4x4, индексированного luma4x4BlkIdx = 0..15, определена такая последовательность действий.
 - a. Находят переменную lumaList, которая является списком из 16 входов. Первый вход lumaList представляет соответствующее значение из массива dcY. На рисунке 8-6 показано присвоение индексов массива dcY значению luma4x4BlkIdx. Две цифры в малых квадратах относятся к индексам i и j в dcY_{ij} , а цифры в больших квадратах относятся к luma4x4BlkIdx.

00	01	02	03
0	1	4	5
10	11	12	13
2	3	6	7
20	21	22	23
8	9	12	13
30	31	32	33
10	11	14	15

Рисунок 8-6 – Присвоение индексов массива dcY значению luma4x4BlkIdx

Элементы в lumaList с индексами $k = 1..15$ определены как

$$\text{lumaList}[k] = \text{Intra16x16ACLevel}[\text{luma4x4BlkIdx}][k - 1]. \quad (8-296)$$

- b. Процесс сканирования инверсного преобразования коэффициентов, как описано в п. 8.5.5, активируют с lumaList в качестве входа и двумерным массивом с в качестве выхода.
- c. Процесс масштабирования и преобразования для остаточных блоков 4x4, который описан в п. 8.5.10, активируют с параметрами с в качестве входа и г в качестве выхода.
- d. Если residual_colour_transform_flag равно 1, переменная $R_{Y,ij}$ устанавливается равной r_{ij} с $i, j = 0..7$, и процесс приостанавливается до завершения процесса остаточного преобразования цвета, описание которого содержится в п. 8.5.13, а по его завершении переменная r_{ij} устанавливается равной $R_{G,ij}$ с $i, j = 0..3$, и продолжается выполнение данного процесса.
- e. Положение верхнего левого образца блока яркости 4x4 с индексом luma4x4BlkIdx внутри макроблока находят, активируя процесс инверсного сканирования блока яркости 4x4, описание которого приведено в п. 6.4.3, с luma4x4BlkIdx в качестве входа и выходом, которому присваивается (xO, yO).
- f. Массив u 4x4 с элементами u_{ij} для $i, j = 0..3$ находят как:

$$u_{ij} = \text{Clip1}(\text{pred}_L[xO + j, yO + i] + r_{ij}). \quad (8-297)$$

Если qpprime_y_zero_transform_bypass_flag равно 1 и QP'_Y равно 0, поток битов не должен содержать данных, в результате которых значение u_{ij} , вычисляемое по уравнению 8-297, окажется не равным $\text{pred}_L[xO + j, yO + i] + r_{ij}$.

- g. Активируют процесс построения изображения до процесса фильтра устранения блочности, который описан в п. 8.5.12, с luma4x4BlkIdx и u в качестве входа и S' в качестве выхода.

8.5.3 Описание процесса декодирования преобразования остаточных блоков яркости 8x8

Данное описание применяется, если transform_size_8x8_flag равно 1.

Переменная LumaLevel8x8[luma8x8BlkIdx] с luma8x8BlkIdx = 0..3 содержит уровни для коэффициентов преобразования яркости блока яркости 8x8 с index luma8x8BlkIdx.

Для блока яркости 8x8, индексированного luma8x8BlkIdx = 0..3, определены следующие упорядоченные шаги.

1. Активируют процесс инверсного сканирования коэффициентов преобразования яркости 8x8, который описан в п. 8.5.6 с LumaLevel8x8[luma8x8BlkIdx] в качестве входа и двумерным массивом с в качестве выхода.
2. Активируют процесс масштабирования и преобразования остаточных блоков 8x8, который описан в п. 8.5.11, с параметрами с в качестве входа и г в качестве выхода.
3. Если residual_colour_transform_flag равно 1, переменная $R_{Y,ij}$ устанавливается равной r_{ij} с $i, j = 0..7$, и процесс приостанавливается до завершения процесса остаточного преобразования цвета, который описан в п. 8.5.13, а по его завершении переменная r_{ij} устанавливается равной $R_{G,ij}$ с $i, j = 0..7$, и продолжается выполнение данного процесса.

4. Положение верхнего левого образца блока яркости 8x8 с индексом luma8x8BlkIdx внутри макроблока находят, активируя процесс инверсного сканирования блока яркости 8x8, который описан в п. 6.4.4, с luma8x8BlkIdx в качестве входа и выходом, которому присваивается (xO, yO) .
5. Массив u 8x8 с элементами u_{ij} для $i, j = 0..7$ находят как:

$$u_{ij} = \text{Clip}_{1\gamma}(\text{pred}_L[xO + j, yO + i] + r_{ij}). \quad (8-298)$$

Если $\text{qrrprime_y_zero_transform_bypass_flag}$ равно 1 и QP'_Y равно 0, поток битов не должен содержать данных, в результате которых значение u_{ij} , вычисляемое по уравнению 8-298, окажется не равным $\text{pred}_L[xO + j, yO + i] + r_{ij}$.

6. Активируют процесс построения изображения до процесса фильтра устранения блочности, который описан в п. 8.5.12, с luma8x8BlkIdx и u в качестве входа.

8.5.4 Описание процесса декодирования преобразования образцов цветности

Этот процесс активируют для каждого компонента цветности Cb и Cr отдельно.

Для каждого компонента цветности переменные $\text{ChromaDCLevel}[iCbCr]$ и $\text{ChromaACLevel}[iCbCr]$ с $iCbCr$, установленным равным 0 для Cb, и $iCbCr$, установленным равным 1 для Cr, содержат уровни для обоих компонентов коэффициентов преобразования цветности.

Пусть переменная numChroma4x4Blks установлена равной $(\text{MbWidthC} / 4) * (\text{MbHeightC} / 4)$.

Для каждого компонента цветности декодирование преобразования производят отдельно следующими шагами.

1. Декодируют коэффициенты преобразования цветности DC numChroma4x4Blks блоков цветности 4x4 компонента, индексированного $iCbCr$ макроблока.

a. В зависимости от переменной chroma_format_idc применяется следующее.

- Если chroma_format_idc равно 1, массив с 2x2 находят, используя процесс инверсного растрового сканирования, применяемый к ChromaDCLevel следующим образом:

$$c = \begin{bmatrix} \text{ChromaDCLevel}[iCbCr][0] & \text{ChromaDCLevel}[iCbCr][1] \\ \text{ChromaDCLevel}[iCbCr][2] & \text{ChromaDCLevel}[iCbCr][3] \end{bmatrix}. \quad (8-299)$$

- Иначе, если chroma_format_idc равно 2, массив 2x4 с находят, используя процесс инверсного растрового сканирования, применяемый к ChromaDCLevel следующим образом:

$$c = \begin{bmatrix} \text{ChromaDCLevel}[iCbCr][0] & \text{ChromaDCLevel}[iCbCr][2] \\ \text{ChromaDCLevel}[iCbCr][1] & \text{ChromaDCLevel}[iCbCr][5] \\ \text{ChromaDCLevel}[iCbCr][3] & \text{ChromaDCLevel}[iCbCr][6] \\ \text{ChromaDCLevel}[iCbCr][4] & \text{ChromaDCLevel}[iCbCr][7] \end{bmatrix}. \quad (8-300)$$

- Иначе (chroma_format_idc равно 3) активируется процесс инверсного сканирования для коэффициентов преобразования, которые описан в п. 8.5.5, с $\text{ChromaDCLevel}[iCbCr]$ в качестве входа и двумерным массивом 4x4 с в качестве выхода.

b. Процесс масштабирования и преобразования для коэффициентов преобразования DC цветности, который определен в п. 8.5.9, активируют с c в качестве входа и dcC в качестве выхода.

2. Для каждого блока цветности 4x4, индексированного $\text{chroma4x4BlkIdx} = 0..\text{numChroma4x4Blks} - 1$ компонента индексированного $iCbCr$, определены следующие упорядоченные шаги.

a. Находят переменную chromaList , которая является списком из 16 входов. Первый вход chromaList – это соответствующее значение из массива dcC . На рисунке 8-7 показано присвоение индексов массива dcC параметру chroma4x4BlkIdx . Две цифры в малых квадратах относятся к индексам i и j в dcC_{ij} , а цифры в больших квадратах – к chroma4x4BlkIdx .

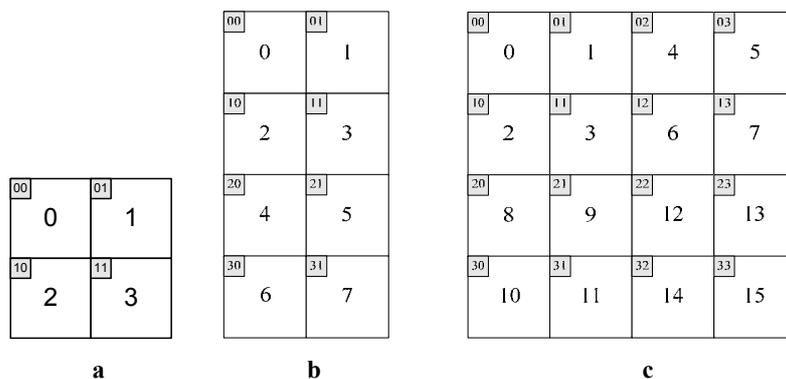


Рисунок 8-7 – Присвоение индексов массива dcC параметру chroma4x4BlkIdx:
a) chroma_format_idc равно 1, b) chroma_format_idc равно 2, c) chroma_format_idc равно 3

Элементы в chromaList с индексами $k = 1..15$ определены как:

$$\text{chromaList}[k] = \text{ChromaACLevel}[\text{chroma4x4BlkIdx}][k - 1]. \quad (8-301)$$

- b. Активируют процесс инверсного сканирования коэффициентов преобразования, который описан в п. 8.5.9, с chromaList в качестве входа и двумерным массивом c в качестве выхода.
- c. Активируют процессы масштабирования и преобразования для остаточных блоков 4x4, которые описаны в п. 8.5.10, с c в качестве входа и g в качестве выхода.
- d. В зависимости от переменной chroma_format_idc, местоположение верхнего левого образца блока цветности 4x4 с индексом chroma4x4BlkIdx внутри макроблока находят следующим образом.

– Если chroma_format_idc равно 1 или 2, применяется следующее:

$$xO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 0); \quad (8-302)$$

$$yO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 1). \quad (8-303)$$

– Иначе (chroma_format_idc равно 3) применяется следующее:

$$xO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 0) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 0); \quad (8-304)$$

$$yO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 1) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 1). \quad (8-305)$$

- e. Если residual_colour_transform_flag равно 1, переменная xO' устанавливается равной $xO \% (4 \ll \text{transform_size_8x8_flag})$, переменная yO' устанавливается равной $yO \% (4 \ll \text{transform_size_8x8_flag})$, и применяется следующее.

– Если этот процесс активируют для компонента цветности Cb, переменную $R_{Cb,mn}$ устанавливают равной r_{ij} с $i, j = 0..3$, $m = xO' + i$, $n = yO' + j$, и процесс приостанавливается до завершения процесса остаточного преобразования цвета, описание которого содержится в п. 8.5.13, а по его завершении переменная r_{ij} устанавливается равной $R_{B,mn}$ с $i, j = 0..3$, $m = xO' + i$, $n = yO' + j$, и продолжается выполнение данного процесса.

– Иначе (этот процесс активируют для компонента цветности Cr) переменную $R_{Cr,mn}$ устанавливают равной r_{ij} с $i, j = 0..3$, $m = xO' + i$, $n = yO' + j$, и процесс приостанавливается до завершения процесса остаточного преобразования цвета, описание которого содержится в п. 8.5.13, а по его завершении переменная r_{ij} устанавливается равной $R_{B,mn}$ с $R_{R,mn}$ с $i, j = 0..3$, $m = xO' + i$, $n = yO' + j$, и продолжается выполнение данного процесса.

- f. Массив u 4x4 с элементами u_{ij} для $i, j = 0..3$ находят как:

$$u_{ij} = \text{Clip1}(\text{pred}_c[xO + j, yO + i] + r_{ij}). \quad (8-306)$$

Если `qrprime_y_zero_transform_bypass_flag` равно 1 и QP'_Y равно 0, поток битов не должен содержать данных, в результате которых значение u_{ij} , вычисляемое по уравнению 8-306, окажется не равным $\text{pred}_c[xO + j, yO + i] + r_{ij}$.

- г. Процесс построения изображения до процесса фильтра устранения блочности, который описан в п. 8.5.12, активируют с `chroma4x4BlkIdx` и u в качестве входа.

8.5.5 Процесс инверсного сканирования для коэффициентов преобразования

Вход в этот процесс – список, содержащий 16 значений.

Выход этого процесса – переменная s , содержащая двумерный массив значений 4x4. В случае коэффициентов преобразования уровней эти значения 4x4 представляют уровни, присвоенные местоположениям в блоке преобразования. В случае применения процесса инверсного сканирования к списку масштабирования переменная s на выходе содержит двумерный массив, представляющий матрицу масштабирования 4x4.

Процесс инверсного сканирования для коэффициентов преобразования отображает последовательность уровней коэффициентов преобразования в местоположения уровней коэффициентов преобразования. Таблица 8-13 определяет два отображения: инверсное сканирование зигзагом и инверсное сканирование полем. Инверсное сканирование зигзагом используется для коэффициентов преобразования в макроблоках кадра, а инверсное сканирование полем используется для коэффициентов преобразования в макроблоках поля.

Процесс инверсного сканирования для списков масштабирования отображает последовательность входов списка масштабирования в местоположения в соответствующей матрице масштабирования. Для такого отображения используется инверсное сканирование зигзагом.

На рисунке 8-8 показан порядок сканирования.

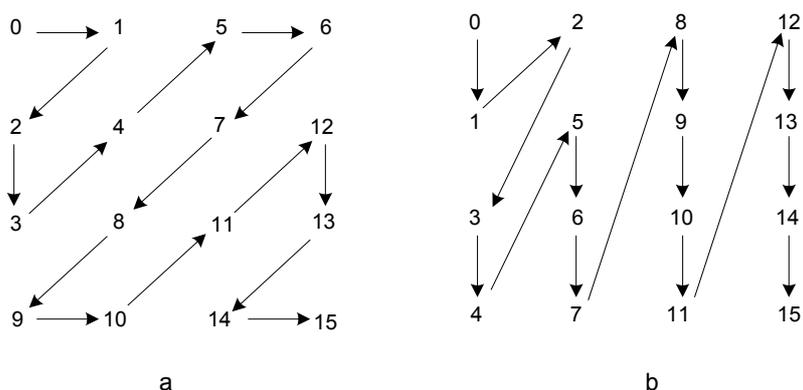


Рисунок 8-8 – Сканирование блоков 4x4: а) сканирование зигзагом; б) сканирование полем

В таблице 8-13 показано отображение индексов `idx` списка из 16 элементов на входе в индексы i и j двумерного массива s .

Таблица 8-13 – Спецификация отображения `idx` в c_{ij} для сканирования зигзагом и полем

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
зигзаг	c_{00}	c_{01}	c_{10}	c_{20}	c_{11}	c_{02}	c_{03}	c_{12}	c_{21}	c_{30}	c_{31}	c_{22}	c_{13}	c_{23}	c_{32}	c_{33}
поле	c_{00}	c_{10}	c_{01}	c_{20}	c_{30}	c_{11}	c_{21}	c_{31}	c_{02}	c_{12}	c_{22}	c_{32}	c_{03}	c_{13}	c_{23}	c_{33}

8.5.6 Процесс инверсного сканирования для коэффициентов преобразования яркости 8x8

Вход в этот процесс – список, содержащий 64 значения.

Выход этого процесса – переменная *s*, содержащая двумерный массив значений 8x8. В случае коэффициентов преобразования уровней эти значения 8x8 представляют уровни, присвоенные местоположениям в блоке преобразования. В случае применения процесса инверсного сканирования к списку масштабирования переменная *s* на выходе содержит двумерный массив, представляющий матрицу масштабирования 8x8.

Процесс инверсного сканирования для коэффициентов преобразования отображает последовательность уровней коэффициентов преобразования в местоположения уровней коэффициентов преобразования. Таблица 8-14 определяет два отображения: инверсное сканирование 8x8 зигзагом и инверсное сканирование 8x8 полем. Инверсное сканирование 8x8 зигзагом используется для коэффициентов преобразования в макроблоках кадра, а инверсное сканирование 8x8 полем используется для коэффициентов преобразования в макроблоках поля.

Процесс инверсного сканирования для списков масштабирования отображает последовательность входов списка масштабирования в местоположения в соответствующей матрице масштабирования. Для такого отображения используется инверсное сканирование зигзагом.

На рисунке 8-9 показан порядок сканирования.

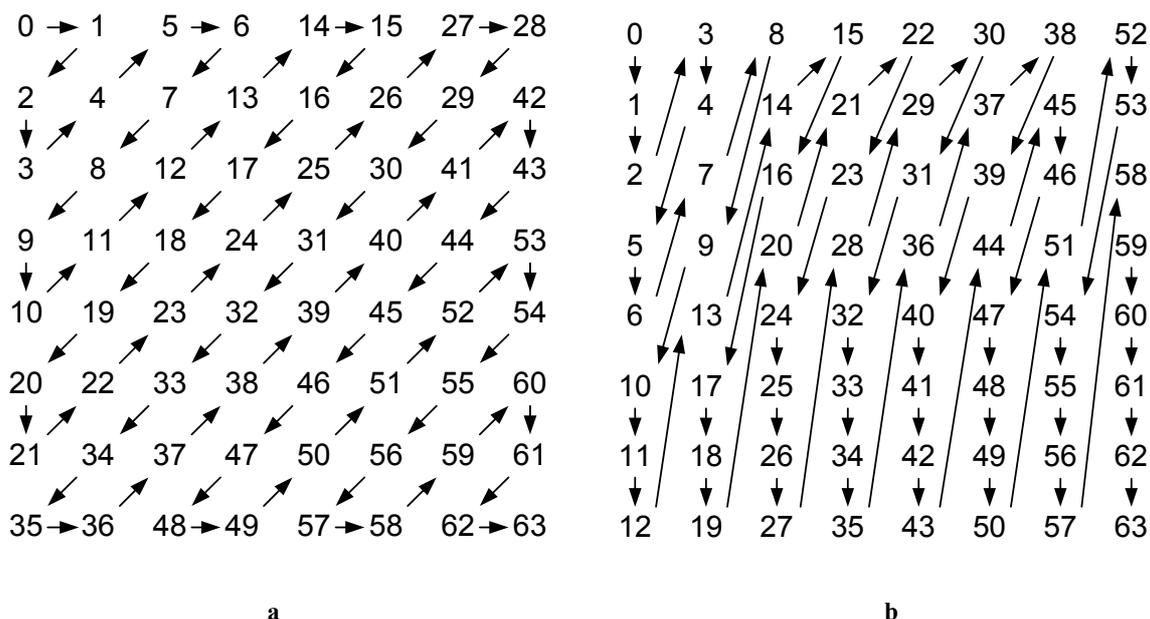


Рисунок 8-9 – Сканирование блоков 8x8:
а) сканирование зигзагом 8x8; б) сканирование полем 8x8 (для информации)

В таблице 8-14 показано отображение индексов *idx* списка из 16 элементов на входе в индексы *i* и *j* двумерного массива *s*.

Таблица 8-14 – Спецификация отображения idx в c_{ij} для сканирования зигзагом 8x8 и полем 8x8

idx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
зигзаг	c ₀₀	c ₀₁	c ₁₀	c ₂₀	c ₁₁	c ₀₂	c ₀₃	c ₁₂	c ₂₁	c ₃₀	c ₄₀	c ₃₁	c ₂₂	c ₁₃	c ₀₄	c ₀₅
поле	c ₀₀	c ₁₀	c ₂₀	c ₀₁	c ₁₁	c ₃₀	c ₄₀	c ₂₁	c ₀₂	c ₃₁	c ₅₀	c ₆₀	c ₇₀	c ₄₁	c ₁₂	c ₀₃

Таблица 8-14 (продолжение) – Спецификация отображения idx в c_{ij} для сканирования зигзагом 8x8 и полем 8x8

idx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
зигзаг	c ₁₄	c ₂₃	c ₃₂	c ₄₁	c ₅₀	c ₆₀	c ₅₁	c ₄₂	c ₃₃	c ₂₄	c ₁₅	c ₀₆	c ₀₇	c ₁₆	c ₂₅	c ₃₄
поле	c ₂₂	c ₅₁	c ₆₁	c ₇₁	c ₃₂	c ₁₃	c ₀₄	c ₂₃	c ₄₂	c ₅₂	c ₆₂	c ₇₂	c ₃₃	c ₁₄	c ₀₅	c ₂₄

Таблица 8-14 (продолжение) – Спецификация отображения idx в c_{ij} для сканирования зигзагом 8x8 и полем 8x8

idx	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
зигзаг	c ₄₃	c ₅₂	c ₆₁	c ₇₀	c ₇₁	c ₆₂	c ₅₃	c ₄₄	c ₃₅	c ₂₆	c ₁₇	c ₂₇	c ₃₆	c ₄₅	c ₅₄	c ₆₃
поле	c ₄₃	c ₅₃	c ₆₃	c ₇₃	c ₃₄	c ₁₅	c ₀₆	c ₂₅	c ₄₄	c ₅₄	c ₆₄	c ₇₄	c ₃₅	c ₁₆	c ₂₆	c ₄₅

Таблица 8-14 (продолжение) – Спецификация отображения idx в c_{ij} для сканирования зигзагом 8x8 и полем 8x8

idx	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
зигзаг	c ₇₂	c ₇₃	c ₆₄	c ₅₅	c ₄₆	c ₃₇	c ₄₇	c ₅₆	c ₆₅	c ₇₄	c ₇₅	c ₆₆	c ₅₇	c ₆₇	c ₇₆	c ₇₇
поле	c ₅₅	c ₆₅	c ₇₅	c ₃₆	c ₀₇	c ₁₇	c ₄₆	c ₅₆	c ₆₆	c ₇₆	c ₂₇	c ₃₇	c ₄₇	c ₅₇	c ₆₇	c ₇₇

8.5.7 Процесс вывода параметров квантования цветности и функция масштабирования

Выходы из этого процесса:

- QP_C: параметр квантования цветности для каждого компонента цветности C_b и C_r;
- QS_C: дополнительный параметр квантования цветности для каждого компонента цветности C_b и C_r, требующийся для декодирования секций SP и SI (если применимо).

ПРИМЕЧАНИЕ 1. – Значения QP параметров квантования QP_Y и QS_Y всегда находятся в диапазоне от –QpBdOffset_Y до 51 включительно. Значения QP параметров квантования QP_C и QS_C всегда находятся в диапазоне от –QpBdOffset_C до 51 включительно.

Значение QP_C для компонента цветности определяют из текущего значения QP_Y и значения chroma_qp_index_offset (для C_b) или second_chroma_qp_index_offset (для C_r).

ПРИМЕЧАНИЕ 2. – Уравнения для масштабирования определены таким образом, что эквивалентный множитель масштабирования уровня коэффициента преобразования удваивается при каждом приращении на 6 в QP_Y. Следовательно, возрастание множителя масштабирования происходит приблизительно на 12% при каждом возрастании на 1 значения QP_Y.

Значение QP_C для каждого компонента цветности определяют согласно таблице 8-15, базируясь на индексе, обозначенном qP₁.

Переменную qP_{Offset} для каждого компонента цветности получают следующим образом.

- Если компонентом цветности является компонент C_b, qP_{Offset} определяется как:

$$qP_{Offset} = chroma_qp_index_offset. \quad (8-307)$$

- Иначе (компонентом цветности является компонент C_r) qP_{Offset} определяется как:

$$qP_{Offset} = second_chroma_qp_index_offset. \quad (8-308)$$

Значение qP₁ для каждого компонента цветности находят следующим образом:

$$qP_1 = Clip3(-QpBdOffset_C, 51, QP_Y + qP_{Offset}). \quad (8-309)$$

Значение QP'_C для компонентов цветности находят следующим образом:

$$QP'_C = QP_C + QpBdOffset_C. \quad (8-310)$$

Значение $BitDepth'_C$ для компонентов цветности находят следующим образом:

$$BitDepth'_C = BitDepth_C + residual_colour_transform_flag. \quad (8-311)$$

Таблица 8-15 – Спецификация QP_C как функции qP_I

qP_I	<30	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
QP_C	$=qP_I$	29	30	31	32	32	33	34	34	35	35	36	36	37	37	37	38	38	38	39	39	39	39

Если текущая секция – это секция SP или SI, то QS_C находят, используя вышеуказанный процесс, заменяя QP_Y на QS_Y и QP_C на QS_C .

Функция $LevelScale(m, i, j)$ определяется следующим образом.

- Матрица 4×4 $weightScale(i, j)$ определяется следующим образом.
 - Переменную $mbIsInterFlag$ находят следующим образом:
 - если текущий макроблок кодирован с использованием режимов Inter предсказания макроблоков, $mbIsInterFlag$ устанавливается равным 1;
 - иначе (текущий макроблок кодирован с использованием режимов Intra предсказания макроблоков) $mbIsInterFlag$ устанавливается равным 0.
 - Переменную $iYCbCr$ находят следующим образом:
 - если массив s на входе соотносится с остаточным блоком яркости, $iYCbCr$ устанавливается равным 0;
 - иначе, если массив s на входе соотносится с остаточным блоком цветности, и компонент цветности равен Cb , $iYCbCr$ устанавливается равным 1;
 - иначе (массив s на входе соотносится с остаточным блоком цветности, и компонент цветности равен Cr) $iYCbCr$ устанавливается равным 2.
 - Процесс инверсного сканирования для коэффициентов преобразования, который описан в п. 8.5.5, активируется с $ScalingList4x4[iYCbCr + (mbIsInterFlag == 1) ? 3 : 0]$ в качестве входа, а выход назначается матрице 4×4 $weightScale$.

$$LevelScale(m, i, j) = weightScale(i, j) * normAdjust(m, i, j), \quad (8-312)$$

где:

$$normAdjust(m, i, j) = \begin{cases} v_{m0} & \text{для } (i\%2, j\%2), \text{ равного } (0,0), \\ v_{m1} & \text{для } (i\%2, j\%2), \text{ равного } (1,1), \\ v_{m2} & \text{иначе,} \end{cases} \quad (8-313)$$

где первое и второе описания v – это, соответственно, ряд и столбец матрицы, определенной как:

$$v = \begin{bmatrix} 10 & 16 & 13 \\ 11 & 18 & 14 \\ 13 & 20 & 16 \\ 14 & 23 & 18 \\ 16 & 25 & 20 \\ 18 & 29 & 23 \end{bmatrix}. \quad (8-314)$$

Функция $\text{LevelScale8x8}(m, i, j)$ определяется следующим образом.

- Матрица 8×8 $\text{weightScale8x8}(i, j)$ определяется следующим образом.
 - Переменную mbIsInterFlag находят следующим образом:
 - если текущий макроблок кодирован с использованием режимов Inter предсказания макроблоков, mbIsInterFlag устанавливается равным 1;
 - иначе (текущий макроблок кодирован с использованием режимов Intra предсказания макроблоков) mbIsInterFlag устанавливается равным 0.
- Процесс инверсного сканирования для коэффициентов преобразования яркости 8×8 , который описан в п. 8.5.5, активируется с $\text{ScalingList8x8}[\text{mbIsInterFlag}]$ в качестве входа, а выход назначается матрице weightScale8x8 .

$$\text{LevelScale8x8}(m, i, j) = \text{weightScale8x8}(i, j) * \text{normAdjust8x8}(m, i, j), \quad (8-315)$$

где:

$$\text{normAdjust8x8}(m, i, j) = \begin{cases} v_{m0} \text{ для } (i\%4, j\%4), \text{ равного } (0,0), \\ v_{m1} \text{ для } (i\%2, j\%2), \text{ равного } (1,1), \\ v_{m2} \text{ для } (i\%4, j\%4), \text{ равного } (2,2), \\ v_{m3} \text{ для } (i\%4, j\%2), \text{ равного } (0,1), \text{ или } (i\%2, j\%4), \text{ равного } (1,0), \\ v_{m4} \text{ для } (i\%4, j\%4), \text{ равного } (0,2), \text{ или } (i\%4, j\%4), \text{ равного } (2,0), \\ v_{m5} \text{ иначе,} \end{cases} \quad (8-316)$$

где первое и второе описания v – это, соответственно, ряд и столбец матрицы, определенной как:

$$v = \begin{bmatrix} 20 & 18 & 32 & 19 & 25 & 24 \\ 22 & 19 & 35 & 21 & 28 & 26 \\ 26 & 23 & 42 & 24 & 33 & 31 \\ 28 & 25 & 45 & 26 & 35 & 33 \\ 32 & 28 & 51 & 30 & 40 & 38 \\ 36 & 32 & 58 & 34 & 46 & 43 \end{bmatrix}. \quad (8-317)$$

8.5.8 Процесс масштабирования и преобразования DC коэффициентов преобразования яркости макроблоков типа Intra_16x16

Входы в этот процесс – значения уровней коэффициентов преобразования для DC коэффициентов преобразования яркости макроблоков Intra_16x16 как массива с 4×4 с элементами c_{ij} , где i и j формируют двумерный индекс частоты.

Выходы из этого процесса – 16-уровневые DC значения блоков яркости 4×4 макроблоков Intra_16x16 как массива 4×4 dcY с элементами dcY_{ij} .

В зависимости от значений $\text{qpprime_y_zero_transform_bypass_flag}$ и QP'_Y применяется следующее.

- Если $\text{qpprime_y_zero_transform_bypass_flag}$ равно 1 и QP'_Y равно 0, dcY на выходе находят следующим образом:

$$dcY_{ij} = c_{ij} \quad c \quad i, j = 0..3. \quad (8-318)$$

- Иначе ($\text{qpprime_y_zero_transform_bypass_flag}$ равно 0 или QP'_Y не равно 0) выход определяется следующим текстом этого процесса.

Инверсное преобразование DC коэффициентов преобразования яркости 4x4 определяется как:

$$f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \quad (8-319)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент f_{ij} из f с $i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{BitDepth}_Y)}$ до $2^{(7 + \text{BitDepth}_Y)} - 1$ включительно.

После инверсного преобразования масштабирование выполняют следующим образом.

- Если QP'_Y больше или равно 36, масштабированный результат получают следующим образом:

$$dcY_{ij} = (f_{ij} * \text{LevelScale}(QP'_Y \% 6, 0, 0)) \ll (QP'_Y / 6 - 6), \quad c \quad i, j = 0..3. \quad (8-320)$$

- Иначе (QP'_Y меньше 36) масштабированный результат получают следующим образом:

$$dcY_{ij} = (f_{ij} * \text{LevelScale}(QP'_Y \% 6, 0, 0) + 2^{5 - QP'_Y / 6}) \gg (6 - QP'_Y / 6), \quad c \quad i, j = 0..3. \quad (8-321)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент dcY_{ij} из dcY с $i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{BitDepth}_Y)}$ до $2^{(7 + \text{BitDepth}_Y)} - 1$ включительно.

ПРИМЕЧАНИЕ 1. – Если `entropy_coding_mode_flag` равно 0, и QP'_Y меньше 10, и `profile_idc` равно 66, 77 или 88, диапазон значений, который может быть представлен для элементов c_{ij} из c может оказаться недостаточным для представления полного диапазона значений элементов dcY_{ij} из dcY , что может быть необходимым для формирования хорошей аппроксимации содержания любого возможного источника изображения при использовании макроблока типа `Intra_16x16`.

ПРИМЕЧАНИЕ 2. – Поскольку этот диапазон накладывает ограничение на элементы dcY_{ij} из dcY после сдвига вправо в уравнении 8-321, в декодере перед сдвигом вправо должен поддерживаться более широкий диапазон значений.

8.5.9 Процесс масштабирования и преобразования для DC коэффициентов преобразования цветности

Входы в этот процесс – значения уровней коэффициентов преобразования DC коэффициентов преобразования цветности одного из компонентов цветности макроблока как массива с $(\text{MbWidthC} / 4) \times (\text{MbHeightC} / 4)$ с элементами c_{ij} , где i и j формируют двумерный индекс частоты.

Выходы из этого процесса – 4-уровневые DC значения в форме массива dcC $(\text{MbWidthC} / 4) \times (\text{MbHeightC} / 4)$ с элементами dcC_{ij} .

В зависимости от значений `qpprime_y_zero_transform_bypass_flag` и QP'_Y применяется следующее.

- Если `qpprime_y_zero_transform_bypass_flag` равно 1 и QP'_Y равно 0, dcC на выходе находят следующим образом:

$$dcC_{ij} = c_{ij} \quad c \quad i = 0..(\text{MbWidthC} / 4) - 1 \quad \text{и} \quad j = 0..(\text{MbHeightC} / 4) - 1. \quad (8-322)$$

- Иначе (`qpprime_y_zero_transform_bypass_flag` равно 0 или QP'_Y не равно 0) выход определяется следующим текстом этого процесса.

В зависимости от переменной `chroma_format_idc` инверсное преобразование определяется следующим образом.

- Если `chroma_format_idc` равно 1, инверсное преобразования для DC коэффициентов преобразования цветности 2x2 определяется как:

$$f = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (8-323)$$

- Иначе, если `chroma_format_idc` равно 2, инверсное преобразование для DC коэффициентов преобразования цветности 2x4 определяется как:

$$f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \\ c_{20} & c_{21} \\ c_{30} & c_{31} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (8-324)$$

– Иначе (chroma_format_idc равно 3) инверсное преобразование для DC коэффициентов преобразования цветности 4x4 определяется следующим образом.

– Если residual_colour_transform_flag равно 1, и режимом предсказания текущего макроблока MbPartPredMode(mb_type, 0) является Intra_4x4 или Intra_8x8, инверсное преобразование для DC коэффициентов преобразования цветности 4x4 определяется как:

$$f_{ij} = c_{ij} \ll 2 \quad c \quad i, j = 0..3. \quad (8-325)$$

– Иначе инверсное преобразование для DC коэффициентов преобразования цветности 4x4 определяется как:

$$f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00} & c_{01} & c_{02} & c_{03} \\ c_{10} & c_{11} & c_{12} & c_{13} \\ c_{20} & c_{21} & c_{22} & c_{23} \\ c_{30} & c_{31} & c_{32} & c_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \quad (8-326)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент f_{ij} из f с $i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{BitDepth}'_C)}$ до $2^{(7 + \text{BitDepth}'_C)} - 1$ включительно.

После инверсного преобразования масштабирование, в зависимости от переменной chroma_format_idc, выполняют следующим образом.

– Если chroma_format_idc равно 1, масштабированный результат получают следующим образом:

$$dcC_{ij} = ((f_{ij} * \text{LevelScale}(QP'_C \% 6, 0, 0)) \ll (QP'_C / 6)) \gg 5, \quad c \quad i, j = 0, 1. \quad (8-327)$$

– Если chroma_format_idc равно 2, применяется следующее.

– Переменную $QP'_{C,DC}$ находят как:

$$QP'_{C,DC} = QP'_C + 3. \quad (8-328)$$

– В зависимости от значения $QP'_{C,DC}$ применяется следующее.

– Если $QP'_{C,DC}$ больше или равно 36, масштабированный результат получают следующим образом:

$$dcC_{ij} = (f_{ij} * \text{LevelScale}(QP'_{C,DC} \% 6, 0, 0)) \ll (QP'_{C,DC} / 6 - 6), \quad c \quad i = 0..3, j = 0, 1. \quad (8-329)$$

– Иначе ($QP'_{C,DC}$ меньше 36) масштабированный результат получают следующим образом:

$$dcC_{ij} = (f_{ij} * \text{LevelScale}(QP'_{C,DC} \% 6, 0, 0) + 2^{5 - QP'_{C,DC} / 6}) \gg (6 - QP'_{C,DC} / 6), \quad c \quad i = 0..3, j = 0, 1. \quad (8-330)$$

– Иначе (chroma_format_idc равно 3) применяется следующее.

– Если QP'_C больше или равно 36, масштабированный результат получают следующим образом:

$$dcC_{ij} = (f_{ij} * \text{LevelScale}(QP'_C \% 6, 0, 0)) \ll (QP'_C / 6 - 6), \quad c \quad i, j = 0..3. \quad (8-331)$$

– Иначе (QP'_C меньше 36) масштабированный результат получают следующим образом:

$$dcC_{ij} = (f_{ij} * \text{LevelScale}(QP'_C \% 6, 0, 0) + 2^{5 - QP'_C / 6}) \gg (6 - QP'_C / 6), \quad c \quad i, j = 0..3. \quad (8-332)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент dcC_{ij} из dcC с $i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{BitDepth}'_C)}$ до $2^{(7 + \text{BitDepth}'_C)} - 1$ включительно.

ПРИМЕЧАНИЕ 1. – Если $\text{entropy_coding_mode_flag}$ равно 0, и QP'_C меньше 4, и profile_idc равно 66, 77 или 88, диапазон значений, который может быть представлен для элементов c_{ij} из c , может оказаться недостаточным для представления полного диапазона значений элементов dcC_{ij} из dcC , что может быть необходимым для формирования хорошей аппроксимации содержания любого возможного источника изображения.

ПРИМЕЧАНИЕ 2. – Поскольку этот диапазон накладывает ограничение на элементы dcC_{ij} из dcC после сдвига вправо в уравнении 8-327, 8-330 или 8-332 в декодере перед сдвигом вправо должен поддерживаться более широкий диапазон значений.

8.5.10 Процесс масштабирования и преобразования остаточных блоков 4x4

Вход в этот процесс – массив c 4x4 с элементами c_{ij} , которые либо являются массивом, связанным с остаточным блоком компонента яркости, либо массивом, связанным с остаточным блоком компонента цветности.

Выходы из этого процесса – значения остаточных образцов в форме массива g 4x4 с элементами g_{ij} .

В зависимости от значений $\text{qpprime_y_zero_transform_bypass_flag}$ и QP'_Y применяется следующее.

- Если $\text{qpprime_y_zero_transform_bypass_flag}$ равно 1 и QP'_Y равно 0, выход находят следующим образом:

$$g_{ij} = c_{ij} \quad \text{с } i, j = 0..3. \quad (8-333)$$

- Иначе ($\text{qpprime_y_zero_transform_bypass_flag}$ равно 0 или QP'_Y не равно 0) выход определяется следующим текстом этого процесса.

Переменную bitDepth находят следующим образом:

- если массив c на входе соотносится с остаточным блоком яркости, bitDepth устанавливается равным BitDepth_Y ;
- иначе (массив c на входе соотносится с остаточным блоком цветности) bitDepth устанавливается равным $\text{BitDepth}'_C$.

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент c_{ij} из c с $i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{bitDepth})}$ до $2^{(7 + \text{bitDepth})} - 1$ включительно.

Переменную sMbFlag находят следующим образом:

- если mb_type равно SI или режим предсказания макроблока в секции SP равен Inter, sMbFlag устанавливают равным 1;
- иначе (mb_type не равно SI и режим предсказания макроблока в секции SP не равен Inter) sMbFlag устанавливают равным 0.

Переменную qP находят следующим образом.

- Если массив c на входе соотносится с остаточным блоком яркости, и sMbFlag равно 0:

$$qP = QP'_Y. \quad (8-334)$$

- Иначе, если массив c на входе соотносится с остаточным блоком яркости, и sMbFlag равно 1:

$$qP = QS_Y. \quad (8-335)$$

- Иначе, если массив c на входе соотносится с остаточным блоком цветности, и sMbFlag равно 0,

$$qP = QP'_C. \quad (8-336)$$

- Иначе (массив c на входе соотносится с остаточным блоком цветности, и sMbFlag равно 1):

$$qP = QS_C. \quad (8-337)$$

Масштабирование уровней c_{ij} коэффициентов преобразования блока 4x4 происходит следующим образом.

- Если все из следующих условий истинны:

- i равно 0;
- j равно 0;
- c соотносится с остаточным блоком яркости, кодированному с использованием режима предсказания Intra_16x16, или соотносится с остаточным блоком цветности;

переменную d_{00} находят как:

$$d_{00} = c_{00}. \quad (8-338)$$

– Иначе применяется следующее.

– Если qP больше или равно 24, масштабированный результат получают следующим образом:

$$d_{ij} = (c_{ij} * \text{LevelScale}(qP \% 6, i, j)) \ll (qP / 6 - 4) \text{ с } i, j = 0..3, \text{ кроме случаев, упомянутых выше.} \quad (8-339)$$

– Иначе (qP меньше 24) масштабированный результат получают следующим образом:

$$d_{ij} = (c_{ij} * \text{LevelScale}(qP \% 6, i, j)) \ll (qP / 6), \text{ с } i, j = 0..3, \text{ кроме случаев, упомянутых выше.} \quad (8-340)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент d_{ij} из d с $i, j = 0..3$ превышает диапазон целых значений от $2^{(7 + \text{bitDepth})}$ до $2^{(7 + \text{bitDepth})} - 1$ включительно.

Процесс преобразования должен превращать блок масштабированных коэффициентов преобразования в блок образцов на выходе способом, математически эквивалентным следующему.

Во-первых, каждый (горизонтальный) ряд масштабированных коэффициентов преобразуют, используя одномерное инверсное преобразование следующим образом.

Вычисляют набор предварительных значений следующим образом:

$$e_{i0} = d_{i0} + d_{i2}, \text{ с } i = 0..3; \quad (8-341)$$

$$e_{i1} = d_{i0} - d_{i2}, \text{ с } i = 0..3; \quad (8-342)$$

$$e_{i2} = (d_{i1} \gg 1) - d_{i3}, \text{ с } i = 0..3; \quad (8-343)$$

$$e_{i3} = d_{i1} + (d_{i3} \gg 1), \text{ с } i = 0..3. \quad (8-344)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент e_{ij} из e с $i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{bitDepth})}$ до $2^{(7 + \text{bitDepth})} - 1$ включительно.

Затем из этих предварительных значений вычисляют преобразованный результат:

$$f_{i0} = e_{i0} + e_{i3}, \text{ с } i = 0..3; \quad (8-345)$$

$$f_{i1} = e_{i1} + e_{i2}, \text{ с } i = 0..3; \quad (8-346)$$

$$f_{i2} = e_{i1} - e_{i2}, \text{ с } i = 0..3; \quad (8-347)$$

$$f_{i3} = e_{i0} - e_{i3}, \text{ с } i = 0..3. \quad (8-348)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент f_{ij} из f с $i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{bitDepth})}$ до $2^{(7 + \text{bitDepth})} - 1$ включительно.

Далее каждый (вертикальный) столбец результирующей матрицы преобразуют, используя то же одномерное инверсное преобразование, следующим образом.

Вычисляют набор предварительных значений:

$$g_{0j} = f_{0j} + f_{2j}, \text{ с } j = 0..3; \quad (8-349)$$

$$g_{1j} = f_{0j} - f_{2j}, \text{ с } j = 0..3; \quad (8-350)$$

$$g_{2j} = (f_{1j} \gg 1) - f_{3j}, \quad c \quad j = 0..3; \quad (8-351)$$

$$g_{3j} = f_{1j} + (f_{3j} \gg 1), \quad c \quad j = 0..3. \quad (8-352)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент g_{ij} из $g \quad c \quad i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{bitDepth})}$ до $2^{(7 + \text{bitDepth})} - 1$ включительно.

Затем из этих предварительных значений вычисляют преобразованный результат:

$$h_{0j} = g_{0j} + g_{3j}, \quad c \quad j = 0..3; \quad (8-353)$$

$$h_{1j} = g_{1j} + g_{2j}, \quad c \quad j = 0..3; \quad (8-354)$$

$$h_{2j} = g_{1j} - g_{2j}, \quad c \quad j = 0..3; \quad (8-355)$$

$$h_{3j} = g_{0j} - g_{3j}, \quad c \quad j = 0..3. \quad (8-356)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент h_{ij} из $h \quad c \quad i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{bitDepth})}$ до $2^{(7 + \text{bitDepth})} - 33$ включительно.

После выполнения обоих инверсных одномерных преобразований (горизонтального и вертикального) для получения массива преобразованных образцов окончательные значения остаточных построенных образцов получают как:

$$r_{ij} = (h_{ij} + 2^5) \gg 6 \quad c \quad i, j = 0..3. \quad (8-357)$$

8.5.11 Процесс масштабирования и преобразования остаточных блоков 8x8

Вход в этот процесс – массив с 8x8 с элементами c_{ij} , который является массивом, связанным с остаточным блоком компонента яркости.

Выходы из этого процесса – значения остаточных образцов в форме массива $r \quad 8x8$ с элементами r_{ij} .

В зависимости от значений `qpprime_y_zero_transform_bypass_flag` и QP'_Y применяется следующее.

- Если `qpprime_y_zero_transform_bypass_flag` равно 1 и QP'_Y равно 0, выход находят следующим образом:

$$r_{ij} = c_{ij} \quad c \quad i, j = 0..7. \quad (8-358)$$

- Иначе (`qpprime_y_zero_transform_bypass_flag` равно 0 или QP'_Y не равно 0) выход определяется следующим текстом этого процесса.

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент c_{ij} из $c \quad c \quad i, j = 0..3$ превышает диапазон целочисленных значений от $-2^{(7 + \text{BitDepth}_Y)}$ до $2^{(7 + \text{BitDepth}_Y)} - 1$ включительно.

Процесс масштабирования уровней c_{ij} коэффициентов преобразования блока 8x8 выполняется следующим образом.

- Если QP'_Y больше или равно 36, масштабированный результат получают следующим образом:

$$d_{ij} = (c_{ij} * \text{LevelScale8x8}(QP'_Y \% 6, i, j)) \ll (QP'_Y / 6 - 6), \quad c \quad i, j = 0..7. \quad (8-359)$$

- Иначе (QP'_Y меньше 36) масштабированный результат получают следующим образом:

$$d_{ij} = (c_{ij} * \text{LevelScale8x8}(QP'_Y \% 6, i, j)) + 2^{5-QP'_Y/6} \gg (6 - QP'_Y / 6), \quad c \quad i, j = 0..7. \quad (8-360)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент d_{ij} из $d \quad c \quad i, j = 0..7$ превышает диапазон целочисленных значений от $-2^{(7 + \text{BitDepth}_Y)}$ до $2^{(7 + \text{BitDepth}_Y)} - 1$ включительно.

Процесс преобразования должен превращать блок масштабированных коэффициентов преобразования в блок образцов на выходе способом, математически эквивалентным следующему.

Во-первых, каждый (горизонтальный) ряд масштабированных коэффициентов преобразуют, используя одномерное инверсное преобразование следующим образом.

– Вычисляют набор предварительных значений следующим образом:

$$e_{i0} = d_{i0} + d_{i4} \text{ с } i = 0..7; \quad (8-361)$$

$$e_{i1} = -d_{i3} + d_{i5} - d_{i7} - (d_{i7} \gg 1) \text{ с } i = 0..7; \quad (8-362)$$

$$e_{i2} = d_{i0} - d_{i4} \text{ с } i = 0..7; \quad (8-363)$$

$$e_{i3} = d_{i1} + d_{i7} - d_{i3} - (d_{i3} \gg 1) \text{ с } i = 0..7; \quad (8-364)$$

$$e_{i4} = (d_{i2} \gg 1) - d_{i6} \text{ с } i = 0..7; \quad (8-365)$$

$$e_{i5} = -d_{i1} + d_{i7} + d_{i5} + (d_{i5} \gg 1) \text{ с } i = 0..7; \quad (8-366)$$

$$e_{i6} = d_{i2} + (d_{i6} \gg 1) \text{ с } i = 0..7; \quad (8-367)$$

$$e_{i7} = d_{i3} + d_{i5} + d_{i1} + (d_{i1} \gg 1) \text{ с } i = 0..7. \quad (8-368)$$

– Второй набор предварительных результатов f_{ij} рассчитывают из предварительных значений e_{ij} следующим образом:

$$f_{i0} = e_{i0} + e_{i6} \text{ с } i = 0..7; \quad (8-369)$$

$$f_{i1} = e_{i1} + (e_{i7} \gg 2) \text{ с } i = 0..7; \quad (8-370)$$

$$f_{i2} = e_{i2} + e_{i4}, \text{ с } i = 0..7; \quad (8-371)$$

$$f_{i3} = e_{i3} + (e_{i5} \gg 2) \text{ с } i = 0..7; \quad (8-372)$$

$$f_{i4} = e_{i2} - e_{i4} \text{ с } i = 0..7; \quad (8-373)$$

$$f_{i5} = (e_{i3} \gg 2) - e_{i5} \text{ с } i = 0..7; \quad (8-374)$$

$$f_{i6} = e_{i0} - e_{i6} \text{ с } i = 0..7; \quad (8-375)$$

$$f_{i7} = e_{i7} - (e_{i1} \gg 2) \text{ с } i = 0..7. \quad (8-376)$$

– Затем из этих предварительных значений f_{ij} вычисляют преобразованный результат g_{ij} :

$$g_{i0} = f_{i0} + f_{i7} \text{ с } i = 0..7; \quad (8-377)$$

$$g_{i1} = f_{i2} + f_{i5} \text{ с } i = 0..7; \quad (8-378)$$

$$g_{i2} = f_{i4} + f_{i3} \text{ с } i = 0..7; \quad (8-379)$$

$$g_{i3} = f_{i6} + f_{i1} \text{ с } i = 0..7; \quad (8-380)$$

$$g_{i4} = f_{i6} - f_{i1} \text{ с } i = 0..7; \quad (8-381)$$

$$g_{i5} = f_{i4} - f_{i3} \text{ с } i = 0..7; \quad (8-382)$$

$$g_{i6} = f_{i2} - f_{i5} \text{ с } i = 0..7; \quad (8-383)$$

$$g_{i7} = f_{i0} - f_{i7} \text{ с } i = 0..7. \quad (8-384)$$

Далее каждый (вертикальный) столбец результирующей матрицы преобразуют, используя то же одномерное инверсное преобразование, следующим образом.

- Вычисляют набор предварительных значений h_{ij} из горизонтально преобразованного значения g_{ij} следующим образом:

$$h_{0j} = g_{0j} + g_{4j} \text{ с } j = 0..7; \quad (8-385)$$

$$h_{1j} = -g_{3j} + g_{5j} - g_{7j} - (g_{7j} \gg 1) \text{ с } j = 0..7; \quad (8-386)$$

$$h_{2j} = g_{0j} - g_{4j} \text{ с } j = 0..7; \quad (8-387)$$

$$h_{3j} = g_{1j} + g_{7j} - g_{3j} - (g_{3j} \gg 1) \text{ с } j = 0..7; \quad (8-388)$$

$$h_{4j} = (g_{2j} \gg 1) - g_{6j} \text{ с } j = 0..7; \quad (8-389)$$

$$h_{5j} = -g_{1j} + g_{7j} + g_{5j} + (g_{5j} \gg 1) \text{ с } j = 0..7; \quad (8-390)$$

$$h_{6j} = g_{2j} + (g_{6j} \gg 1) \text{ с } j = 0..7; \quad (8-391)$$

$$h_{7j} = g_{3j} + g_{5j} + g_{1j} + (g_{1j} \gg 1) \text{ с } j = 0..7. \quad (8-392)$$

- Второй набор предварительных результатов k_{ij} рассчитывают из предварительных значений h_{ij} следующим образом:

$$k_{0j} = h_{0j} + h_{6j} \text{ с } j = 0..7; \quad (8-393)$$

$$k_{1j} = h_{1j} + (h_{7j} \gg 2) \text{ с } j = 0..7; \quad (8-394)$$

$$k_{2j} = h_{2j} + h_{4j} \text{ с } j = 0..7; \quad (8-395)$$

$$k_{3j} = h_{3j} + (h_{5j} \gg 2) \text{ с } j = 0..7; \quad (8-396)$$

$$k_{4j} = h_{2j} - h_{4j} \text{ с } j = 0..7; \quad (8-397)$$

$$k_{5j} = (h_{3j} \gg 2) - h_{5j} \text{ с } j = 0..7; \quad (8-398)$$

$$k_{6j} = h_{0j} - h_{6j} \text{ с } j = 0..7; \quad (8-399)$$

$$k_{7j} = h_{7j} - (h_{1j} \gg 2) \text{ с } j = 0..7. \quad (8-400)$$

– Затем из этих предварительных значений k_{ij} вычисляют преобразованный результат m_{ij} :

$$m_{0j} = k_{0j} + k_{7j} \text{ с } j = 0..7; \quad (8-401)$$

$$m_{1j} = k_{2j} + k_{5j} \text{ с } j = 0..7; \quad (8-402)$$

$$m_{2j} = k_{4j} + k_{3j} \text{ с } j = 0..7; \quad (8-403)$$

$$m_{3j} = k_{6j} + k_{1j} \text{ с } j = 0..7; \quad (8-404)$$

$$m_{4j} = k_{6j} - k_{1j} \text{ с } j = 0..7; \quad (8-405)$$

$$m_{5j} = k_{4j} - k_{3j} \text{ с } j = 0..7; \quad (8-406)$$

$$m_{6j} = k_{2j} - k_{5j} \text{ с } j = 0..7; \quad (8-407)$$

$$m_{7j} = k_{0j} - k_{7j} \text{ с } j = 0..7. \quad (8-408)$$

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент e_{ij} , f_{ij} , g_{ij} , h_{ij} или k_{ij} при i и j в диапазоне 0..7 включительно превышает диапазон целочисленных значений от $-2^{(7 + \text{BitDepth}_Y)}$ до $2^{(7 + \text{BitDepth}_Y)} - 1$ включительно.

Поток битов не должен содержать данных, которые приводят к тому, что любой элемент m_{ij} при i и j в диапазоне 0..7 включительно превышает диапазон целочисленных значений от $-2^{(7 + \text{BitDepth}_Y)}$ до $2^{(7 + \text{BitDepth}_Y)} - 33$ включительно.

После выполнения обоих инверсных одномерных преобразований (горизонтального и вертикального) для получения массива преобразованных образцов окончательные значения остаточных построенных образцов получают как:

$$r_{ij} = (m_{ij} + 2^5) \gg 6 \text{ с } i, j = 0..7. \quad (8-409)$$

8.5.12 Процесс построения изображения до процесса фильтра устранения блочности

Входы в этот процесс:

- luma4x4BlkIdx или chroma4x4BlkIdx или luma8x8BlkIdx;
- массив образцов u 4x4 с элементами u_{ij} , которые являются либо блоком яркости 4x4, либо блоком цветности 4x4, либо блоком яркости 8x8.

Положение верхнего левого образца яркости текущего макроблока находят, активируя процесс инверсного сканирования макроблока, который описан п. 6.4.1, с CurrMbAddr в качестве входа и выходом, которому присваивается (xP, yP).

Если u – это блок яркости, то для каждого образца u_{ij} блока яркости 4x4 применяется следующее.

- В зависимости от размера блока u применяется следующее:
 - если u является блоком яркости 4x4, местоположение верхнего левого образца блока яркости с индексом luma4x4BlkIdx внутри макроблока находят, активируя процесс инверсного сканирования блока яркости 8x8, который описан в п. 6.4.3, с luma4x4BlkIdx в качестве входа и выходом, которому присваивается (xO, yO), при этом переменная nE устанавливается равной 4;
 - иначе (u является блоком яркости 8x8) местоположение верхнего левого образца блока яркости 8x8 с индексом luma8x8BlkIdx внутри макроблока находят, активируя процесс инверсного сканирования блока яркости 4x4, который описан в п. 6.4.4, с luma8x8BlkIdx в качестве входа и выходом, которому присваивается (xO, yO), при этом переменная nE устанавливается равной 8.
- В зависимости от переменной MbaffFrameFlag и текущего макроблока применяется следующее.
 - Если MbaffFrameFlag равно 1, а текущий макроблок – макроблок поля:

$$S'_L[xP + xO + j, yP + 2 * (yO + i)] = u_{ij} \text{ с } i, j = 0 \text{ nE} - 1. \quad (8-410)$$

- Иначе (MbaffFrameFlag равно 0 или текущий макроблок является макроблоком кадра):

$$S'_L[xP + xO + j, yP + yO + i] = u_{ij} \quad c \quad i, j = 0.. nE - 1. \quad (8-411)$$

Если u – блок цветности, то для каждого образца u_{ij} блока цветности 4x4 применяется следующее.

- Описание C в переменной S'_C заменяют на C_b для компоненты цветности C_b и на C_r для компоненты цветности C_r .
- В зависимости от переменной `chroma_format_idc` местоположение верхнего левого образца блока цветности 4x4 с индексом `chroma4x4BlkIdx` внутри макроблока находят следующим образом.

- Если `chroma_format_idc` равно 1 or 2, применяется следующее:

$$xO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 0); \quad (8-412);$$

$$yO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 1); \quad (8-413)$$

- Иначе (`chroma_format_idc` равно 3) применяется следующее:

$$xO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 0) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 0); \quad (8-414)$$

$$yO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 1) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 1); \quad (8-415)$$

$$xO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 0) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 0); \quad (8-414)$$

$$yO = \text{InverseRasterScan}(\text{chroma4x4BlkIdx} / 4, 8, 8, 16, 1) + \text{InverseRasterScan}(\text{chroma4x4BlkIdx} \% 4, 4, 4, 8, 1); \quad (8-415)$$

- В зависимости от переменной `MbaffFrameFlag` и текущего макроблока применяется следующее.

- Если `MbaffFrameFlag` равно 1, а текущий макроблок является макроблоком поля:

$$S'_C[(xP / \text{subWidthC}) + xO + j, ((yP + \text{SubHeightC} - 1) / \text{SubHeightC}) + 2 * (yO + i)] = u_{ij} \quad c \quad i, j = 0..3. \quad (8-416)$$

- Иначе (`MbaffFrameFlag` равно 0 или текущий макроблок является макроблоком кадра):

$$S'_C[(xP / \text{subWidthC}) + xO + j, (yP / \text{SubHeightC}) + yO + i] = u_{ij} \quad c \quad i, j = 0..3. \quad (8-417)$$

8.5.13 Процесс остаточного преобразования цвета

Этот процесс активируют, если `residual_colour_transform_flag` равно 1.

После активации процесс приостанавливается, до тех пор пока полностью не будут получены $R_{Y,ij}$, $R_{Cb,ij}$ и $R_{Cr,ij}$ для $i, j = 0..ijMax$, где $ijMax$ определяется следующим образом:

- если `transform_size_8x8_flag` равно 0, переменная $ijMax$ устанавливается равной 3;
- иначе (`transform_size_8x8_flag` равно 1) переменная $ijMax$ устанавливается равной 7.

При возобновлении этого процесса все значения $R_{Y,ij}$, $R_{Cb,ij}$ и $R_{Cr,ij}$ с $i, j = 0..ijMax$ должны быть доступны до вызова соответствующих процессов, описанных в пп. 8.5.1, 8.5.2, 0 или 8.5.4.

Для каждого $i, j = 0..ijMax$ остаточное преобразование цвета рассчитывается как:

$$t = R_{Y,ij} - (R_{Cb,ij} >> 1); \quad (8-418)$$

$$R_{G,ij} = t + R_{Cb,ij}; \quad (8-419)$$

$$R_{B,ij} = t - (R_{Cr,ij} >> 1); \quad (8-420)$$

$$R_{R,ij} = R_{B,ij} + R_{Cr,ij}. \quad (8-421)$$

ПРИМЕЧАНИЕ. – Остаточное преобразование цвета аналогично преобразованию YCgCo, которое описывается уравнениями E-30–E-33. Вместе с тем, остаточное преобразование цвета работает с декодированными данными остаточной разности в рамках процесса декодирования, а не выполняется как шаг постобработки, который находится за рамками процесса декодирования, описываемого в настоящей Рекомендации | Международном стандарте.

8.6 Процесс декодирования макроблоков P в секциях SP или макроблоков SI

Процесс активируют при декодировании макроблоков типа P в секциях типа SP или макроблоков типа SI в секциях SI.

Входы в этот процесс – предсказанные остаточные уровни коэффициентов преобразования и предсказанные образцы текущего макроблока.

Выходы из этого процесса – декодированные образцы текущего макроблока до процесса фильтра устранения блочности.

Этот пункт определяет процесс декодирования коэффициентов преобразования и построения изображения для макроблоков типов P в секциях SP и макроблоков типов SI в секциях SI.

ПРИМЕЧАНИЕ. – Секции SP используют при кодировании с Inter предсказанием, чтобы воспользоваться временной избыточностью в последовательности, аналогично кодированию секции P. Однако в отличие от кодирования секции P кодирование секции SP допускает идентичную реконструкцию секции, если даже используют различные контрольные изображения. SI секции используют пространственные предсказания аналогично I секциям. Кодирование SI секции допускает идентичную реконструкцию соответствующей SP секции. Свойства секций SP и SI помогают обеспечить функциональность потока битов: коммутацию, стыковку, прямой доступ, устойчивость к ошибкам с прямой и обратной коррекцией.

Секция SP состоит из макроблоков, кодированных либо как тип I, либо как тип P.

Секция SI состоит из макроблоков, кодированных либо как тип I, либо как тип SI.

Активируют процессы декодирования коэффициентов преобразования и построения изображения до процесса фильтра устранения блочности для макроблоков типа I в секциях SI, которые описаны в п. 8.5. Макроблок типа SI декодируется, как описано ниже.

Если текущий макроблок кодирован как P_Skip, то для текущего макроблока все значения LumaLevel, ChromaDCLevel, ChromaACLevel устанавливаются равными 0.

8.6.1 Процесс декодирования SP некоммутируемых изображений

Процесс активируют при декодировании макроблоков типа P в SP секциях, в которых sp_for_switch_flag равно 0.

Входы в этот процесс – образцы Inter предсказания для текущего макроблока из п. 8.4 и предсказания уровней остатка коэффициентов преобразования.

Выходы из этого процесса – декодированные образцы текущего макроблока до процесса фильтра устранения блочности.

Этот пункт применим ко всем макроблокам в SP секциях, в которых sp_for_switch_flag равно 0, за исключением тех макроблоков, режим предсказания которых равен Intra_4x4 или Intra_16x16. Этот пункт не применим к SI секциям.

8.6.1.1 Процесс декодирования коэффициентов преобразования яркости

Входы в этот процесс – образцы Inter предсказания яркости для текущего макроблока pred_L из п. 8.4 и предсказания уровней остатка коэффициентов преобразования, LumaLevel и индекс блока яркости 4x4luma4x4BlkIdx.

Положение верхнего левого образца блока яркости 4x4 с индексом luma4x4BlkIdx внутри текущего макроблока находят, активируя процесс инверсного сканирования блока яркости 4x4, который описан в п. 6.4.3, с luma4x4BlkIdx в качестве входа и выходом, которому присваивается (x, y).

Положим, что переменная p – это массив 4x4 образцов предсказания с элементами p_{ij}, которые находят следующим образом:

$$p_{ij} = \text{pred}_L[x + j, y + i] \quad \text{с } i, j = 0..3. \quad (8-422)$$

Переменную p преобразуют, формируя коэффициент преобразования c^P согласно 8-290:

$$c^P = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \\ p_{30} & p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix}. \quad (8-423)$$

Процесс инверсного сканирования коэффициентов преобразования, как описано в п. 8.5.5, активируют с LumaLevel[luma4x4BlkIdx] в качестве входа и двумерным массивом c^T с элементами c_{ij}^T в качестве выхода.

Предсказанные остаточные коэффициенты преобразования c^r масштабируют, используя параметр квантования QP_Y и добавляя к коэффициентам преобразования предсказанных блоков значения c^p с $i, j = 0..3$, следующим образом:

$$c_{ij}^s = c_{ij}^p + ((c_{ij}^r * \text{LevelScale}(QP_Y \% 6, i, j) * A_{ij}) \ll (QP_Y / 6)) \gg 10), \quad (8-424)$$

где $\text{LevelScale}(m, i, j)$ определено из уравнения 8-252, а A_{ij} определено как:

$$A_{ij} = \begin{cases} 16 & \text{для } (i, j) \in \{(0,0), (0,2), (2,0), (2,2)\}, \\ 25 & \text{для } (i, j) \in \{(1,1), (1,3), (3,1), (3,3)\}, \\ 20 & \text{в других случаях.} \end{cases} \quad (8-425)$$

Функция $\text{LevelScale2}(m, i, j)$, используемая в формулах ниже, определяется как:

$$\text{LevelScale2}(m, i, j) = \begin{cases} w_{m0} & \text{для } (i, j) \in \{(0,0), (0,2), (2,0), (2,2)\}, \\ w_{m1} & \text{для } (i, j) \in \{(1,1), (1,3), (3,1), (3,3)\}, \\ w_{m2} & \text{в других случаях,} \end{cases} \quad (8-426)$$

где первая и вторая надписи w обозначают, соответственно, ряд и столбец индексов матрицы, определенной как:

$$w = \begin{bmatrix} 13107 & 5243 & 8066 \\ 11916 & 4660 & 7490 \\ 10082 & 4194 & 6554 \\ 9362 & 3647 & 5825 \\ 8192 & 3355 & 5243 \\ 7282 & 2893 & 4559 \end{bmatrix}. \quad (8-427)$$

Результирующую сумму, c^s , квантуют с параметром квантования QS_Y и с $i, j = 0..3$ следующим образом:

$$c_{ij} = \text{Sign}(c_{ij}^s) * ((\text{Abs}(c_{ij}^s) * \text{LevelScale2}(QS_Y \% 6, i, j) + (1 \ll (14 + QS_Y / 6))) \gg (15 + QS_Y / 6)). \quad (8-428)$$

Процесс масштабирования и преобразования для остатка блоков 4×4 , который описан в п. 8.5.10, активируют с c в качестве входа и g в качестве выхода.

Массив u 4×4 с элементами u_{ij} находят следующим образом:

$$u_{ij} = \text{Clip1}_Y(r_{ij}) \text{ с } i, j = 0..3. \quad (8-429)$$

Процесс построения изображения до процесса фильтра устранения блочности, который описан в п. 8.5.12, активируют с luma4x4BlkIdx и u в качестве входа.

8.6.1.2 Процесс декодирования коэффициентов преобразования цветности

Входы в этот процесс – образцы Inter предсказания цветности для текущего макроблока из п. 8.4 и предсказания уровней остаточных коэффициентов преобразования ChromaDCLevel и ChromaACLevel .

Процесс активируют дважды: один раз для компонента C_b и другой – для компонента C_r . Компонент находят заменой C на C_b для компонента C_b и C на C_r для компонента C_r . Положим, что $iCbCr$ выбирает текущий компонент цветности.

Для каждого блока 4×4 текущего компонента цветности, индексированного с использованием chroma4x4BlkIdx при значениях chroma4x4BlkIdx , равных $0..3$, применяется следующее.

- Положение верхнего левого образца блока цветности 4×4 с индексом chroma4x4BlkIdx внутри макроблока находят следующим образом:

$$x = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 0); \quad (8-430)$$

$$y = \text{InverseRasterScan}(\text{chroma4x4BlkIdx}, 4, 4, 8, 1). \quad (8-431)$$

- Положим, что p – массив 4×4 образцов предсказания с элементами p_{ij} , полученными следующим образом:

$$p_{ij} = \text{pred}_c[x + j, y + i] \quad c \ i, j = 0..3. \quad (8-432)$$

- Массив ' p ' 4×4 преобразуют, формируя коэффициенты преобразования $c^p(\text{chroma4x4BlkIdx})$ с помощью уравнения 8-423.
- Находят переменную chromaList , которая представляет список из 16 входов, $\text{chromaList}[0]$ устанавливают равным 0, $\text{chromaList}[k]$ с индексом $k = 1..15$ определяют следующим образом:

$$\text{chromaList}[k] = \text{ChromaACLevel}[\text{iCbCr}][\text{chroma4x4BlkIdx}][k - 1]. \quad (8-433)$$

- Процесс инверсного сканирования коэффициентов преобразования активируют, как описано в п. 8.5.5, с chromaList в качестве входа и массивом c^r 4×4 в качестве выхода.
- Предсказанные остаточные коэффициенты преобразования c^r масштабируют, используя параметр квантования QP_c и добавляя к коэффициентам преобразования предсказанных блоков значения c^p с $i, j = 0..3$, за исключением комбинации $i = 0, j = 0$, следующим образом:

$$c_{ij}^s = c_{ij}^p(\text{chroma4x4BlkIdx}) + (((c_{ij}^r * \text{LevelScale}(QP_c \% 6, i, j) * A_{ij}) \ll (QP_c / 6)) \gg 10). \quad (8-434)$$

- Результирующую сумму, c^s , квантуют с параметром квантования QS_c и с $i, j = 0..3$, за исключением комбинации $i = 0, j = 0$, следующим образом. Нахождение $c_{00}(\text{chroma4x4BlkIdx})$ описано ниже в этом пункте:

$$c_{ij}(\text{chroma4x4BlkIdx}) = (\text{Sign}(c_{ij}^s) * (\text{Abs}(c_{ij}^s) * \text{LevelScale2}(QS_c \% 6, i, j) + (1 \ll (14 + QS_c / 6)))) \gg (15 + QS_c / 6). \quad (8-435)$$

- Процесс масштабирования и преобразования для остаточных блоков 4×4 , который описан в п. 8.5.10, активируют с $c(\text{chroma4x4BlkIdx})$ в качестве входа и g в качестве выхода.
- Массив u 4×4 с элементами u_{ij} находят следующим образом:

$$u_{ij} = \text{Clip1}(g_{ij}) \quad c \ i, j = 0..3. \quad (8-436)$$

- Процесс построения изображения до процесса фильтра устранения блочности, который описан в п. 8.5.12, активируют с chroma4x4BlkIdx и u в качестве входа.

Вывод уровней DC коэффициентов преобразования $c_{00}(\text{chroma4x4BlkIdx})$ определено следующим образом: DC коэффициенты преобразования 4-х предсказанных блоков цветности 4×4 текущего компонента макроблока объединяют в матрицу 2×2 с элементами $c_{00}^p(\text{chroma4x4BlkIdx})$ и выполняют преобразование 2×2 над DC коэффициентами преобразования следующим образом:

$$dc^p = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} c_{00}^p(0) & c_{00}^p(1) \\ c_{00}^p(2) & c_{00}^p(3) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (8-437)$$

Предсказанные остаточные уровни DC коэффициента преобразования цветности, $\text{ChromaDCLevel}[\text{iCbCr}][k]$ с $k = 0..3$, масштабируют, используя параметр квантования QP и добавляя к DC коэффициентам преобразования следующее:

$$dc_{ij}^s = dc_{ij}^p + (((\text{ChromaDCLevel}[\text{iCbCr}][j * 2 + i] * \text{LevelScale}(QP \% 6, 0, 0) * A_{00}) \ll (QP_c / 6)) \gg 9) \quad c \ i, j = 0, 1. \quad (8-438)$$

Массив dc^s 2×2 квантуют, используя параметр квантования QS_c :

$$dc_{ij}^r = (\text{Sign}(dc_{ij}^s) * (\text{Abs}(dc_{ij}^s) * \text{LevelScale2}(QS_c \% 6, 0, 0) + (1 \ll (15 + QS_c / 6)))) \gg (16 + QS_c / 6) \quad c \ i, j = 0, 1. \quad (8-439)$$

Находят массив 'f' 2x2 с элементами f_{ij} и $i, j = 0..1$:

$$f = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} dc_{00}^r & dc_{01}^r \\ dc_{10}^r & dc_{11}^r \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (8-440)$$

Элементы f_{ij} из f масштабируют следующим образом:

$$c_{00}(j * 2 + i) = ((f_{ij} * \text{LevelScale}(QS_C \% 6, 0, 0)) \ll (QS_C / 6)) \gg 5 \text{ с } i, j = 0, 1. \quad (8-441)$$

8.6.2 Процесс декодирования секций SP и SI для коммутации изображений

Процесс активируют при декодировании макроблоков типа P в секциях SP, в которых `sp_for_switch_flag` равно 1, и при декодировании макроблоков типа SI в секциях SI.

Входы в этот процесс – предсказанные уровни остатков коэффициентов преобразования и массивы предсказанных образцов pred_L , pred_{Cb} и pred_{Cr} для текущего макроблока.

8.6.2.1 Процесс декодирования коэффициентов преобразования яркости

Входы в этот процесс – предсказанные образцы яркости pred_L и предсказанные уровни остатков коэффициентов преобразования яркости, `LumaLevel`.

Массив 'p' 4x4 с элементами p_{ij} с $i, j = 0..3$ находят, как указано в п. 8.6.1.1, преобразованием согласно уравнению 8-423, для вывода коэффициента преобразования c^p . Затем эти коэффициенты преобразования квантуют с параметром квантования QS_Y :

$$c_{ij}^s = \text{Sign}(c_{ij}^p) * ((\text{Abs}(c_{ij}^p) * \text{LevelScale2}(QS_Y \% 6, i, j) + (1 \ll (14 + QS_Y / 6))) \gg (15 + QS_Y / 6)) \text{ с } i, j = 0..3. \quad (8-442)$$

Процесс инверсного сканирования коэффициентов преобразования, который описан в п. 8.5.5, активируют с `LumaLevel[luma4x4BlkIdx]` в качестве входа и двумерным массивом c^r с элементами c_{ij}^r в качестве выхода.

Массив с 4x4 с элементами c_{ij} с $i, j = 0..3$ находят следующим образом:

$$c_{ij} = c_{ij}^r + c_{ij}^s \text{ с } i, j = 0..3. \quad (8-443)$$

Процесс масштабирования и преобразования остаточных блоков 4x4, который описан в п. 8.5.10, активируют с с в качестве входа и г в качестве выхода.

Массив u 4x4 с элементами u_{ij} находят следующим образом:

$$u_{ij} = \text{Clip1}(g_{ij}) \text{ с } i, j = 0..3. \quad (8-444)$$

Процесс построения изображения до процесса фильтра устранения блочности, который описан в п. 8.5.12, активируют с `luma4x4BlkIdx` и u в качестве входа.

8.6.2.2 Процесс декодирования коэффициентов преобразования цветности

Входы в этот процесс – предсказанные образцы цветности для текущего макроблока из п. 8.4 и предсказанные уровни остатка коэффициентов преобразования, `ChromaDCLevel` и `ChromaACLevel`.

Процесс активируют дважды: первый раз для компонента Cb, а второй – для компонента Cr. Компонент находят заменой C на Cb для компонента Cb и C на Cr для компонента Cr. Положим, что `iCbCr` выбирает текущий компонент цветности.

Каждый блок 4x4 текущего компонента цветности индексируют, используя `chroma4x4BlkIdx`, со значениями `chroma4x4BlkIdx`, равными 0..3. Действуют следующие условия:

1. Массив 'p' 4x4 с элементами p_{ij} с $i, j = 0..3$ находят, как указано в п. 8.6.1.2, преобразованием, согласно уравнению 8-423, для вывода коэффициентов преобразования $c^p(\text{chroma4x4BlkIdx})$. Затем эти коэффициенты преобразования квантуют, за исключением комбинации $i = 0, j = 0$, с параметром квантования QS_C с $i, j = 0..3$. Обработка $c_{00}^p(\text{chroma4x4BlkIdx})$ описана ниже в этом пункте:

$$c_{ij}^s = (\text{Sign}(c_{ij}^p(\text{chroma4x4BlkIdx})) * (\text{Abs}(c_{ij}^p(\text{chroma4x4BlkIdx})) * \text{LevelScale2}(\text{QS}_C \% 6, i, j) + (1 \ll (14 + \text{QS}_C / 6)))) \gg (15 + \text{QS}_C / 6). \quad (8-445)$$

- Находят переменную chromaList, которая имеется в списке из 16 входов. Значение chromaList[0] устанавливают равным 0. Значение chromaList[k] с индексами k = 1..15 определяют следующим образом:

$$\text{chromaList}[k] = \text{ChromaACLevel}[\text{iCbCr}][\text{chroma4x4BlkIdx}][k - 1]. \quad (8-446)$$

- Процесс инверсного сканирования коэффициентов преобразования, как это описано в п. 8.5.5, активируют с chromaList в качестве входа и двумерным массивом $c^r(\text{chroma4x4BlkIdx})$ с элементами $c_{ij}^r(\text{chroma4x4BlkIdx})$ в качестве выхода.
- Массив $c(\text{chroma4x4BlkIdx})$ 4x4 с элементами $c_{ij}(\text{chroma4x4BlkIdx})$ с $i, j = 0..3$, за исключением комбинации $i = 0, j = 0$, находят следующим образом. Вывод $c_{00}(\text{chroma4x4BlkIdx})$ описан ниже:

$$c_{ij}(\text{chroma4x4BlkIdx}) = c_{ij}^r(\text{chroma4x4BlkIdx}) + c_{ij}^s. \quad (8-447)$$

- Процесс масштабирования и преобразования остаточный блоков 4x4, который описан в п. 8.5.10, активируют с $c(\text{chroma4x4BlkIdx})$ в качестве входа и g в качестве выхода.
- Массив u 4x4 с элементами u_{ij} находят следующим образом:

$$u_{ij} = \text{Clip1}(r_{ij}) \text{ с } i, j = 0..3. \quad (8-448)$$

- Процесс построения изображения до процесса фильтра устранения блочности, который описан в п. 8.5.12, активируют с chroma4x4BlkIdx и u в качестве входа.

Уровни DC коэффициентов преобразования $c_{00}(\text{chroma4x4BlkIdx})$ находят следующим образом. DC коэффициенты преобразования 4-х предсказанных блоков цветности 4x4 текущих компонентов макроблока, $c_{00}^p(\text{chroma4x4BlkIdx})$, объединяют в матрицу 2x2, и преобразование 2x2 добавляют к DC коэффициентам преобразования этих блоков согласно уравнению 8-437, получая в результате DC коэффициенты преобразования dc_{ij}^p .

Затем эти DC коэффициенты преобразования квантуют с параметром квантования QS_C , заданным как:

$$dc_{ij}^s = (\text{Sign}(dc_{ij}^p) * (\text{Abs}(dc_{ij}^p) * \text{LevelScale2}(\text{QS}_C \% 6, 0, 0) + (1 \ll (15 + \text{QS}_C / 6)))) \gg (16 + \text{QS}_C / 6) \text{ с } i, j = 0, 1. \quad (8-449)$$

Проанализированные предсказанные остаточные DC коэффициенты преобразования цветности, $\text{ChromaDCLevel}[\text{iCbCr}][k]$ с $k = 0..3$ добавляют к этим квантованным DC коэффициентам преобразования предсказанного блока, как показано ниже:

$$dc_{ij}^r = dc_{ij}^s + \text{ChromaDCLevel}[\text{iCbCr}][j * 2 + i] \text{ с } i, j = 0, 1. \quad (8-450)$$

Массив 'f' 2x2 с элементами f_{ij} и $i, j = 0..1$ находят, используя уравнение 8-440.

Массив 'f' 2x2 с элементами f_{ij} и $i, j = 0..1$ копируют следующим образом:

$$c_{00}(j * 2 + i) = f_{ij} \text{ с } i, j = 0, 1. \quad (8-451)$$

8.7 Процесс фильтра устранения блочности

Процесс условной фильтрации применяется ко всем блокам NxN (где N=4 или N=8 для яркости и N=4 для цветности) краев изображения, за исключением краев на границе изображения и любых краев, для которых процесс фильтра устранения блочности аннулирован с помощью команды `disable_deblocking_filter_idc`, как определено ниже. Этот процесс фильтрации выполняют на основе макроблока после завершения процесса построения изображения и до процесса фильтра устранения блочности (как определено в пп. 8.5 и 8.6) полностью декодированного изображения. Все макроблоки изображения обрабатывают в порядке возрастания адресов макроблоков.

ПРИМЕЧАНИЕ 1. – Перед операцией фильтра устранения блочности каждого макроблока разделенные на блоки образцы макроблоков или пары макроблоков выше (если такие имеются) и макроблоки или пары макроблоков слева (если имеются) от текущего макроблока всегда доступны, поскольку процесс фильтра устранения блочности выполняют после завершения процесса построения изображения до процесса фильтра устранения блочности для всего декодированного изображения. Однако для целей определения того, какой край должен подвергаться фильтрации, когда

disable_deblocking_filter_idc равно 2, в течение выполнения описанных шагов процесса фильтра устранения блочности макроблоки в других секциях рассматриваются как недоступные.

Процесс фильтра устранения блочности для компонентов яркости и цветности активируют отдельно. Для каждого макроблока и каждого компонента вертикальные края фильтруют первыми, начиная от края с левой стороны макроблока и перемещаясь к краям на правой стороне макроблока, следуя их геометрическому порядку, а затем фильтруют горизонтальные края, начиная от края сверху макроблока и перемещаясь к нижним краям, следуя их геометрическому порядку. На рисунке 8-10 показаны края макроблока, которые могут быть интерпретированы как края яркости или цветности.

При интерпретации краев на рисунке 8-10 как краев яркости в зависимости от transform_size_8x8_flag применяется следующее:

- если transform_size_8x8_flag равно 0, фильтруются оба типа краев яркости, показанные сплошной линией и пунктирной линией;
- иначе (transform_size_8x8_flag равно 1) фильтруются только края яркости, показанные сплошной линией.

При интерпретации краев на рисунке 8-10 как краев цветности в зависимости от chroma_format_idc применяется следующее:

- если chroma_format_idc равно 1 (формат 4:2:0), фильтруются только края цветности, показанные сплошной линией;
- иначе, если chroma_format_idc равно 2 (формат 4:2:2), фильтруются вертикальные края цветности, показанные сплошной линией, а также оба типа горизонтальных краев цветности, показанные сплошной линией и пунктирной линией;
- иначе, если chroma_format_idc равно 3 (формат 4:4:4), фильтруются оба типа краев цветности, показанные сплошной линией и пунктирной линией;
- иначе (chroma_format_idc равно 0 (монохромное изображение)) все края цветности фильтрации не подлежат.

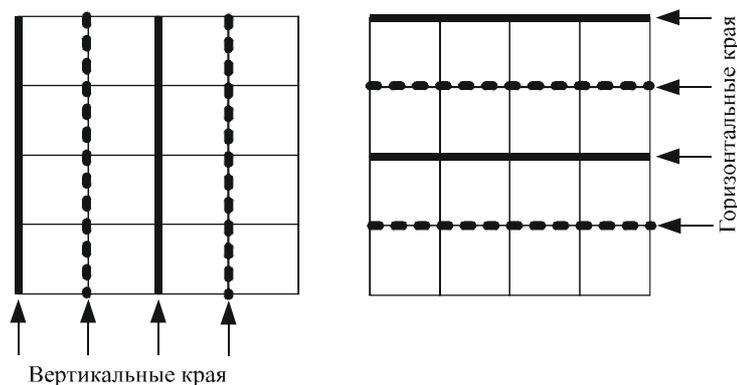


Рисунок 8-10 – Границы в макроблоке, которые следует фильтровать

Для текущего макроблока с адресом CurrMbAddr, принимающим значения 0..PicSizeInMbs – 1, применяется следующее.

1. Активируют процесс создания смежных блоков, который описан в п. 6.4.8.1, а выходу присваивается mbAddrA и mbAddrB.
2. Находят переменные fieldModeMbFlag, filterInternalEdgesFlag, filterLeftMbEdgeFlag и filterTopMbEdgeFlag.
 - Переменную fieldModeMbFlag находят следующим образом.
 - Если любое из следующих условий – истина, то fieldModeMbFlag устанавливают равным 1:
 - field_pic_flag равно 1;
 - MbaffFrameFlag равно 1, а макроблок CurrMbAddr является макроблоком поля.
 - Иначе fieldModeMbFlag устанавливают равным 0.
 - Переменную filterInternalEdgesFlag находят следующим образом.

- Если `disable_deblocking_filter_idc` для секции, которая содержит макроблок `mbAddr`, равно 1, то переменную `filterInternalEdgesFlag` устанавливают на 0.
 - Иначе (`disable_deblocking_filter_idc` для секции, которая содержит макроблок `mbAddr`, не равно 1) переменную `filterInternalEdgesFlag` устанавливают на 1.
 - Переменную `filterLeftMbEdgeFlag` находят следующим образом.
 - Если любое из следующих условий – истина, то переменную `filterLeftMbEdgeFlag` устанавливают на 0:
 - `MbaffFrameFlag` равно 0 и `CurrMbAddr % PicWidthInMbs` равно 0;
 - `MbaffFrameFlag` равно 1, а $(CurrMbAddr \gg 1) \% PicWidthInMbs$ равно 0;
 - `disable_deblocking_filter_idc` для секции, которая содержит макроблок `CurrMbAddr`, равно 1;
 - `disable_deblocking_filter_idc` для секции, которая содержит макроблок `CurrMbAddr`, равно 2, а макроблок `mbAddrA` не доступен.
 - Иначе переменную `filterLeftMbEdgeFlag` устанавливают равной 1.
 - Переменную `filterTopMbEdgeFlag` находят следующим образом.
 - Если любое из следующих условий – истина, то переменную `filterTopMbEdgeFlag` устанавливают на 0:
 - `MbaffFrameFlag` равно 0, и `CurrMbAddr` меньше `PicWidthInMbs`;
 - `MbaffFrameFlag` равно 1, $(CurrMbAddr \gg 1)$ меньше `PicWidthInMbs`, и макроблок `CurrMbAddr` является макроблоком поля;
 - `MbaffFrameFlag` равно 1, $(CurrMbAddr \gg 1)$ меньше `PicWidthInMbs`, макроблок `CurrMbAddr` является макроблоком кадра, и `CurrMbAddr \% 2` равно 0;
 - `disable_deblocking_filter_idc` для секции, которая содержит макроблок `CurrMbAddr`, равно 1;
 - `disable_deblocking_filter_idc` для секции, которая содержит макроблок `CurrMbAddr`, равно 2, а макроблок `mbAddrB` недоступен.
 - Иначе переменную `filterTopMbEdgeFlag` устанавливают равной 1.
3. С заданными переменными `fieldModeMbFlag`, `filterInternalEdgesFlag`, `filterLeftMbEdgeFlag` и `filterTopMbEdgeFlag` фильтрацией для устранения блочности управляют следующим образом.
- Если `filterLeftMbEdgeFlag` равно 1, фильтрация левого вертикального края яркости определяется следующим образом:
 - `Frnbbdhet.n ghjwcc`, определенный в п. 8.7.1, с `chromaEdgeFlag = 0`, `verticalEdgeFlag = 1`, `fieldModeFilteringFlag = fieldModeMbFlag` и $(xE_k, yE_k) = (0, k)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода.
 - Если `filterInternalEdgesFlag` равно 1, фильтрация внутренних вертикальных краев яркости определяется следующим образом.
 - Если `transform_size_8x8_flag` равно 0, активируется процесс, определенный в п. 8.7.1, с `chromaEdgeFlag = 0`, `verticalEdgeFlag = 1`, `fieldModeFilteringFlag = fieldModeMbFlag` и $(xE_k, yE_k) = (4, k)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода.
 - Активируется процесс, определенный в п. 8.7.1, с `chromaEdgeFlag = 0`, `verticalEdgeFlag = 1`, `fieldModeFilteringFlag = fieldModeMbFlag` и $(xE_k, yE_k) = (8, k)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода.
 - Если `transform_size_8x8_flag` равно 0, активируется процесс, определенный в п. 8.7.1, с `chromaEdgeFlag = 0`, `verticalEdgeFlag = 1`, `fieldModeFilteringFlag = fieldModeMbFlag` и $(xE_k, yE_k) = (12, k)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода.
 - Если `filterTopMbEdgeFlag` равно 1, фильтрация верхнего горизонтального края яркости определяется следующим образом.
 - Если `MbaffFrameFlag` равно 1, $(CurrMbAddr \% 2)$ равно 0, `CurrMbAddr` больше или равно $2 * PicWidthInMbs$, макроблок `CurrMbAddr` является макроблоком кадра, и макроблок $(CurrMbAddr - 2 * PicWidthInMbs + 1)$ является макроблоком поля, применяется следующее:
 - активируется процесс, определенный в п. 8.7.1, с `chromaEdgeFlag = 0`, `verticalEdgeFlag = 0`, `fieldModeFilteringFlag = 1` и $(xE_k, yE_k) = (k, 0)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода;

- активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 0$, $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = 1$ и $(x_{E_k}, y_{E_k}) = (k, 1)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода.
- Иначе, активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 0$, $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (k, 0)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода.
- Если $\text{filterInternalEdgesFlag}$ равно 1, фильтрация внутренних горизонтальных краев яркости определяется следующим образом.
 - Если $\text{transform_size_8x8_flag}$ равно 0, активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 0$, $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (k, 4)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода.
 - Активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 0$, $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (k, 8)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода.
 - Если $\text{transform_size_8x8_flag}$ равно 0, активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 0$, $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (k, 12)$ с $k = 0..15$ в качестве входа и S'_L в качестве выхода.
- Для фильтрация обоих компонентов с $iCbCr = 0$ для Cb и $iCbCr = 1$ для Cr применяется следующее.
 - Если $\text{filterLeftMbEdgeFlag}$ равно 1, фильтрация левого вертикального края цветности определяется следующим образом.
 - Активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, $iCbCr$, $\text{verticalEdgeFlag} = 1$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (0, k)$ с $k = 0..MbHeightC - 1$ в качестве входа и S'_C с C, которое заменяется на Cb для $iCbCr = 0$, и C, которое заменяется на Cr для $iCbCr = 1$, в качестве выхода.
 - Если $\text{filterInternalEdgesFlag}$ равно 1, фильтрация внутреннего вертикального края цветности определяется следующим образом:
 - активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, $iCbCr$, $\text{verticalEdgeFlag} = 1$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (4, k)$ с $k = 0..MbHeightC - 1$ в качестве входа и S'_C с C, которое заменяется на Cb для $iCbCr = 0$, и C, которое заменяется на Cr для $iCbCr = 1$, в качестве выхода;
 - если chroma_format_idc равно 3, активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, $iCbCr$, $\text{verticalEdgeFlag} = 1$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (8, k)$ с $k = 0..MbHeightC - 1$ в качестве входа и S'_C с C, которое заменяется на Cb для $iCbCr = 0$, и C, которое заменяется на Cr для $iCbCr = 1$, в качестве выхода;
 - если chroma_format_idc равно 3, активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, $iCbCr$, $\text{verticalEdgeFlag} = 1$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (12, k)$ с $k = 0..MbHeightC - 1$ в качестве входа и S'_C с C, которое заменяется на Cb для $iCbCr = 0$, и C, которое заменяется на Cr для $iCbCr = 1$, в качестве выхода.
 - Если $\text{filterTopMbEdgeFlag}$ равно 1, фильтрация верхнего горизонтального края цветности определяется следующим образом.
 - Если MbaffFrameFlag равно 1, $(\text{CurrMbAddr} \% 2)$ равно 0, CurrMbAddr больше или равно $2 * \text{PicWidthInMbs}$, макроблок CurrMbAddr является макроблоком кадра, а макроблок $(\text{CurrMbAddr} - 2 * \text{PicWidthInMbs} + 1)$ является макроблоком поля, применяется следующее:
 - активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, $iCbCr$, $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = 1$ и $(x_{E_k}, y_{E_k}) = (k, 0)$ с $k = 0..MbWidthC - 1$ в качестве входа и S'_C с C, которое заменяется на Cb для $iCbCr = 0$, и C, которое заменяется на Cr для $iCbCr = 1$, в качестве выхода;
 - активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, $iCbCr$, $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = 1$ и $(x_{E_k}, y_{E_k}) = (k, 1)$ с $k = 0..MbWidthC - 1$ в качестве входа и S'_C с C, которое заменяется на Cb для $iCbCr = 0$, и C, которое заменяется на Cr для $iCbCr = 1$, в качестве выхода.

- Иначе активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, iCbCr , $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (k, 0)$ с $k = 0..MbWidthC - 1$ в качестве входа и S'_C с C , которое заменяется на C_b для $\text{iCbCr} = 0$, и C , которое заменяется на C_r для $\text{iCbCr} = 1$, в качестве выхода.
- Если $\text{filterInternalEdgesFlag}$ равно 1, фильтрация внутреннего горизонтального края цветности определяется следующим образом.
 - Активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, iCbCr , $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (k, 4)$ с $k = 0..MbWidthC - 1$ в качестве входа и S'_C с C , которое заменяется на C_b для $\text{iCbCr} = 0$, и C , которое заменяется на C_r для $\text{iCbCr} = 1$, в качестве выхода.
 - Если chroma_format_idc не равно 1, активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, iCbCr , $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (k, 8)$ с $k = 0..MbWidthC - 1$ в качестве входа и S'_C с C , которое заменяется на C_b для $\text{iCbCr} = 0$, и C , которое заменяется на C_r для $\text{iCbCr} = 1$, в качестве выхода.
 - Если chroma_format_idc не равно 1, активируется процесс, определенный в п. 8.7.1, с $\text{chromaEdgeFlag} = 1$, iCbCr , $\text{verticalEdgeFlag} = 0$, $\text{fieldModeFilteringFlag} = \text{fieldModeMbFlag}$ и $(x_{E_k}, y_{E_k}) = (k, 12)$ с $k = 0..MbWidthC - 1$ в качестве входа и S'_C с C , которое заменяется на C_b для $\text{iCbCr} = 0$, и C , которое заменяется на C_r для $\text{iCbCr} = 1$, в качестве выхода.

ПРИМЕЧАНИЕ 2 – Если фильтрация в режиме поля ($\text{fieldModeFilteringFlag}$ равно 1) применяется по верхним горизонтальным краям макроблока кадра, эта вертикальная фильтрация по верхней или нижней границе макроблока может использовать некоторые образцы, перекрывающие внутренний край блока, который также внутренне фильтруется в режиме кадра.

ПРИМЕЧАНИЕ 3 – Например, в формате цветности 4:2:0, если $\text{transform_size_8x8_flag}$ равно 0, применяется следующее. Фильтруют 3 горизонтальных края яркости, 1 горизонтальный край цветности для C_b и 1 горизонтальный край цветности для C_r , которые являются внутренними для макроблока. Если фильтрация в режиме поля ($\text{fieldModeFilteringFlag}$ равно 1) применяется к верхним краям макроблока кадра, фильтруют 2 горизонтальных края яркости, 2 горизонтальных края цветности для C_b и 2 горизонтальных края цветности для C_r между макроблоком кадра и верхней парой макроблоков, применяя фильтрацию в режиме поля, в сумме до 5 горизонтальных краев яркости, 3 горизонтальных краев цветности для C_b и 3 горизонтальных краев цветности для C_r , которые считаются находящимися под управлением этого макроблока кадра. Во всех других случаях фильтруют не более 4 горизонтальных краев яркости, 2 горизонтальных краев цветности для C_b и 2 – для C_r , которые считаются находящимися под управлением конкретного макроблока.

Наконец, значения массивов S'_L , S'_{Cb} , S'_{Cr} присваивают массивам S_L , S_{Cb} , S_{Cr} (которые представляют декодированное изображение), соответственно.

8.7.1 Процесс фильтрации краев блоков

Входы в этот процесс – mbAddr , chromaEdgeFlag , индекс компонента цветности iCbCr (если chromaEdgeFlag равно 1), verticalEdgeFlag , $\text{fieldModeFilteringFlag}$, и множество местоположений образцов nE (x_{E_k}, y_{E_k}) с $k = 0..nE - 1$, выраженных относительно верхнего левого угла макроблока CurrMbAddr . Множество местоположений образцов (x_{E_k}, y_{E_k}) представляют местоположения образцов непосредственно справа от вертикального края (если verticalEdgeFlag равно 1) или непосредственно ниже вертикального края (если verticalEdgeFlag равно 0).

Переменную nE находят следующим образом:

- если chromaEdgeFlag равно 0, то nE устанавливают равным 16;
- иначе (chromaEdgeFlag равно 1) nE устанавливается равным $(\text{verticalEdgeFlag} == 1) ? MbHeightC : MbWidthC$.

Положим, что s' – переменная, определяющая массив образцов яркости или цветности, которую находят следующим образом:

- если chromaEdgeFlag равно 0, то s' представляет массив S'_L образцов яркости текущего изображения;
- иначе, если chromaEdgeFlag равно 1, а iCbCr равно 0, то s' представляет массив S'_{Cb} образцов цветности компонента цветности C_b текущего изображения;
- иначе (chromaEdgeFlag равно 1 и iCbCr равно 1) s' представляет массив S'_{Cr} образцов цветности компонента цветности C_r текущего изображения.

Переменную dy находят следующим образом:

- если $\text{fieldModeFilteringFlag}$ равно 1 и MbaffFrameFlag равно 1, dy устанавливают равным 2;
- иначе ($\text{fieldModeFilteringFlag}$ равно 0 или MbaffFrameFlag равно 0) dy устанавливают равным 1.

Местоположение верхнего левого образца яркости макроблока CurrMbAddr находят, активируя процесс инверсного сканирования макроблока, который описан в п. 6.4.1, с mbAddr = CurrMbAddr в качестве входа, и выходом, которому присваивается (xI, yI).

Переменные xP и yP находят следующим образом:

- если chromaEdgeFlag равно 0, xP устанавливается равным xI, а yP устанавливается равным yI;
- иначе (chromaEdgeFlag равно 1) xP устанавливается равным xI / SubWidthC, а yP устанавливается равным (yI + SubHeightC – 1) / SubHeightC.

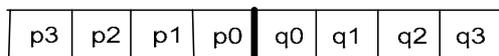


Рисунок 8-11 – Условные обозначения для описания образцов блоков 4x4 вдоль горизонтальной или вертикальной границы

Для каждого местоположения образца (xE_k, yE_k), k = 0 .. nE – 1 применяется следующее.

- Процесс фильтрации применяют к множеству из восьми образцов блоков 4x4 вдоль горизонтального и вертикального краев, обозначенных как p_i и q_i с i = 0..3, как показано на рисунке 8-11, с краем, расположенным между p₀ и q₀. Значения p_i и q_i с i = 0..3 определены следующим образом.

- Если verticalEdgeFlag равно 1:

$$q_i = s'[xP + xE_k + i, yP + dy * yE_k]; \quad (8-452)$$

$$p_i = s'[xP + xE_k - i - 1, yP + dy * yE_k]. \quad (8-453)$$

- Иначе (verticalEdgeFlag равно 0):

$$q_i = s'[xP + xE_k, yP + dy * (yE_k + i) - (yE_k \% 2)]; \quad (8-454)$$

$$p_i = s'[xP + xE_k, yP + dy * (yE_k - i - 1) - (yE_k \% 2)]. \quad (8-455)$$

- Процесс, определенный в п. 8.7.2, активируют со значениями образца p_i и q_i (i = 0..3), chromaEdgeFlag, verticalEdgeFlag и fieldModeFilteringFlag в качестве входа и с выходом, присвоенным отфильтрованным результирующим значениям образца p'_i и q'_i с i = 0..2.

- Вход значений образца p_i и q_i с i = 0..2 заменяют соответствующими отфильтрованными результирующими значениями образца p'_i и q'_i с i = 0..2 внутри массива s' образцов следующим образом.

- Если verticalEdgeFlag равно 1:

$$s'[xP + xE_k + i, yP + dy * yE_k] = q'_i; \quad (8-456)$$

$$s'[xP + xE_k - i - 1, yP + dy * yE_k] = p'_i \quad (8-457)$$

- Иначе (verticalEdgeFlag равно 0):

$$s'[xP + xE_k, yP + dy * (yE_k + i) - (yE_k \% 2)] = q'_i; \quad (8-458)$$

$$s'[xP + xE_k, yP + dy * (yE_k - i - 1) - (yE_k \% 2)] = p'_i. \quad (8-459)$$

8.7.2 Процесс фильтрации набора образцов вдоль горизонтального и вертикального краев блока

Входы в этот процесс – вход значений образца p_i и q_i с i в диапазоне 0..3 для одного множества образцов вдоль края, который следует фильтровать, `chromaEdgeFlag`, `verticalEdgeFlag` и `fieldModeFilteringFlag`.

Выходы этого процесса – отфильтрованные результирующие значения образца p'_i и q'_i с i в диапазоне 0..2.

Переменную `bS`, которую определяют интенсивностью фильтрации границы в зависимости от степени яркости, находят следующим образом:

- если `chromaEdgeFlag` равно 0, процесс вывода интенсивности фильтрации границы, определенный в п. 8.7.2.1, активируют с p_0 , q_0 и `verticalEdgeFlag` в качестве входа и с выходом, присвоенным параметру `bS`;
- иначе (`chromaEdgeFlag` равно 1) `bS`, используемое для фильтрации набора образцов горизонтального или вертикального края цветности, устанавливается равным значению `bS` для фильтрации набора образцов горизонтального или вертикального краев яркости, соответственно, которое содержит образец яркости в местоположении (`SubWidthC * x`, `SubHeightC * y`) внутри массива яркости того же поля, где (x, y) – это местоположение образца цветности q_0 внутри массива цветности этого поля.

Процесс, определенный в п. 8.7.2.2, активируют с p_0 , q_0 , p_1 , q_1 , `chromaEdgeFlag` и `bS` в качестве входа, и выходом, который присваивается `filterSamplesFlag`, `indexA`, α и β .

В зависимости от переменной `filterSamplesFlag` применяется следующее.

- Если `filterSamplesFlag` равно 1, применяется следующее:
 - если `bS` меньше 4, активируют процесс, определенный в п. 8.7.2.3, с p_i и q_i ($i = 0..2$), `chromaEdgeFlag`, `bS`, β и `indexA`, заданными в качестве входа, а выход – присвоенные значения p'_i и q'_i ($i = 0..2$);
 - иначе (`bS` равно 4) процесс, определенный в п. 8.7.2.4, активируют с p_i и q_i ($i = 0..3$), `chromaEdgeFlag`, α и β , заданными в качестве входа, а выход – присвоенные значения p'_i и q'_i ($i = 0..2$).
- Иначе (`filterSamplesFlag` равно 0) результирующие отфильтрованные образцы p'_i и q'_i ($i = 0..2$) заменяют соответствующими образцами входа p_i и q_i :

$$\text{для } i = 0..2, \quad p'_i = p_i; \quad (8-460)$$

$$\text{для } i = 0..2, \quad q'_i = q_i. \quad (8-461)$$

8.7.2.1 Процесс вывода интенсивности фильтрации границы в зависимости от степени яркости

Входы в этот процесс – входные значения образцов p_0 и q_0 единственного множества образцов вдоль края, который следует фильтровать, а также `verticalEdgeFlag`.

Выход этого процесса – переменная `bS`.

Положим, что переменную `mixedModeEdgeFlag` можно найти следующим образом:

- если `MbaffFrameFlag` равно 1 и образцы p_0 и q_0 – разные пары макроблоков, одна из которых – пара макроблоков поля, а другая – пара макроблоков кадра, то `mixedModeEdgeFlag` устанавливают равным 1;
- иначе `mixedModeEdgeFlag` устанавливают равным 0.

Переменную `bS` находят следующим образом.

- Если край блока является также и краем макроблока, а любое из следующих условий – истина, то значение `bS`, равное 4, является выходом:
 - образцы p_0 и q_0 находятся оба в макроблоках кадра, и любой из них или оба, p_0 или q_0 , расположены в макроблоке, кодированном с использованием режима Intra предсказания макроблока;
 - образцы p_0 и q_0 находятся оба в макроблоках кадра, и любой из них или оба, p_0 или q_0 , расположены в макроблоке, который входит в секцию со значением `slice_type`, равным SP или SI;
 - `MbaffFrameFlag` равно 1 или `field_pic_flag` равно 1, и `verticalEdgeFlag` равно 1, и любой из образцов или оба, p_0 или q_0 , расположены в макроблоке, кодированном с использованием режима Intra предсказания макроблока;
 - `MbaffFrameFlag` равно 1 или `field_pic_flag` равно 1, и `verticalEdgeFlag` равно 1, и любой из образцов или оба, p_0 или q_0 , расположены в макроблоке, который входит в секцию со значением `slice_type`, равным SP или SI.

- Иначе, если любое из следующих условий – истина, то значение bS, равно 3, является выходом:
 - mixedModeEdgeFlag равно 0, и любой из образцов или оба, p₀ или q₀, расположены в макроблоке, кодированном с использованием режима Intra предсказания макроблока;
 - mixedModeEdgeFlag равно 0, и любой из образцов или оба, p₀ или q₀, расположены в макроблоке, который входит в секцию со значением slice_type, равным SP или SI;
 - mixedModeEdgeFlag равно 1, verticalEdgeFlag равно 0, и любой из образцов или оба, p₀ или q₀, расположены в макроблоке, кодированном с использованием режима Intra предсказания макроблока;
 - mixedModeEdgeFlag равно 1, verticalEdgeFlag равно 0, и любой из образцов или оба, p₀ или q₀, расположены в макроблоке, который входит в секцию со значением slice_type, равным SP или SI.
- Иначе, если следующее условие – истина, то значение bS, равно 2, является выходом:
 - блок яркости, содержащий образец p₀, или блок яркости, содержащий образец q₀, с ненулевыми уровнями коэффициентов преобразования.
- Иначе, если следующее условие – истина, то значение bS, равно 1, является выходом:
 - mixedModeEdgeFlag равно 1;
 - mixedModeEdgeFlag равно 0, и предсказание разделения макроблока/субмакроблока, содержащего образец p₀ с контрольными изображениями или числом векторов движения, отличными от тех, которые используют для предсказания разделения макроблока/субмакроблока, содержащего образец q₀;

ПРИМЕЧАНИЕ 1. – Определение того, являются ли контрольные изображения, которые используются для двух разделений макроблоков/субмакроблоков, одинаковыми или разными, базируется только на том, какие изображения являются контрольными, не учитывая формируется ли предсказание с использованием индекса в списке контрольных изображений 0 или индекса в списке контрольных изображений 1, а также не учитывая, было или не было различным местоположение индекса в списке контрольных изображений.
 - mixedModeEdgeFlag равно 0, и один вектор движения используют, чтобы предсказать разделение макроблока/субмакроблока, содержащего образец p₀, а другой вектор движения – чтобы предсказать разделение макроблока/субмакроблока, содержащего образец q₀. При этом учитывают абсолютную разность между горизонтальными или вертикальными компонентами векторов движения, большую или равную 4 в единицах четверти яркости образцов кадра;
 - mixedModeEdgeFlag равно 0, и два вектора движения и два различных контрольных изображения используют, чтобы предсказать разделение макроблока/субмакроблока, содержащего образец p₀, а два вектора движения тех же контрольных изображений – чтобы предсказать разделение макроблока/субмакроблока, содержащего образец q₀. При этом учитывают абсолютную разность между горизонтальными или вертикальными компонентами двух векторов движения для предсказания разделения двух макроблоков/субмакроблоков для этого же контрольного изображения. Эта разность должна быть больше или равна 4 в единицах четверти яркости образцов кадра;
 - mixedModeEdgeFlag равно 0, и два вектора движения для того же контрольного изображения используют, чтобы предсказать разделение макроблока/субмакроблока, содержащего образец p₀, а два вектора движения того же контрольного изображения используют, чтобы предсказать разделение макроблока/субмакроблока, содержащего образец q₀, а оба из следующих условий являются истиной:
 - абсолютная разность между горизонтальными или вертикальными компонентами векторов движения списка 0, которую используют для предсказания разделения двух макроблоков/субмакроблоков, больше или равна 4 в единицах четверти яркости образцов кадра, или абсолютная разность между горизонтальными или вертикальными компонентами векторов движения списка 1, которую используют для предсказания разделения двух макроблоков/субмакроблоков, больше или равна 4 в единицах четверти яркости образцов кадра;
 - абсолютная разность между горизонтальными или вертикальными компонентами векторов движения списка 0, которую используют для предсказания разделения двух макроблоков/субмакроблоков, содержащих образец p₀, и такая же разность для предсказания разделения двух макроблоков/субмакроблоков, содержащих образец q₀ списка 1, больше или равна 4 в единицах четверти яркости образцов кадра. Или абсолютная разность между горизонтальными или вертикальными компонентами векторов движения списка 1, которую используют для предсказания разделения двух макроблоков/субмакроблоков, содержащих образец p₀, и такая же разность для предсказания разделения двух макроблоков/субмакроблоков, содержащих образец q₀ списка 0, больше или равна 4 в единицах четверти яркости образцов кадра.

ПРИМЕЧАНИЕ 2. – Вертикальная разность в 4 единицы четверти яркости образцов кадра соответствует разности в 2 единицы четверти яркости образцов поля.
- Иначе выходом является значение bS, равно 0.

8.7.2.2 Процесс вывода порогов для края каждого блока

Входы в этот процесс – вход значений образцов p_0 , q_0 , p_1 и q_1 одного множества образцов вдоль края, предназначенного для фильтрации, chromaEdgeFlag и bS для множества входов образцов, как это определено в п. 8.7.2.

Выходы из этого процесса – переменная filterSamplesFlag , которая указывает, были ли фильтрованы входы образцов, значение indexA и значения пороговых переменных α и β .

Положим, что qP_p и qP_q – переменные, определяющие значения параметров квантования для макроблоков, содержащих образцы p_0 и q_0 соответственно. Переменные qP_z (с заменой z на p или q) находят следующим образом.

- Если chromaEdgeFlag равно 0, применяется следующее:
 - если макроблок, содержащий образец z_0 , это макроблок I_PCM , то qP_z устанавливают на 0;
 - иначе (макроблок, содержащий образец z_0 , это не макроблок I_PCM) qP_z устанавливают на значение QP_Y макроблока, содержащего образец z_0 .
- Иначе (chromaEdgeFlag равно 1) применяется следующее:
 - если макроблок, содержащий образец z_0 , это макроблок I_PCM , то qP_z устанавливают на значение QP_C , которое соответствует значению 0 для QP_Y , как это определено в п. 8.5.7;
 - иначе (макроблок, содержащий образец z_0 , это не макроблок I_PCM) qP_z устанавливают на значение QP_C , которое соответствует значению QP_Y макроблока, содержащего образец z_0 , как это определено в п. 8.5.7.

Положим, что qP_{av} – переменная, определяющая средний параметр квантования. Ее находят следующим образом:

$$qP_{av} = (qP_p + qP_q + 1) \gg 1. \quad (8-462)$$

ПРИМЕЧАНИЕ. – В секциях SP и SI значение qP_{av} находят таким же образом, как и в других типах секций. При фильтре устранения блочности значение QS_Y из уравнения 7-28 не используют.

Положим, что indexA – переменная, которую используют для доступа к таблице α (таблица 8-16), а также к таблице t_{c0} (таблица 8-17), которые применяют для фильтрации краев с помощью значения bS меньше 4, как определено в п. 8.7.2.3. Положим, что indexB – переменная, которую используют для доступа к таблице β (таблица 8-16). Переменные indexA и indexB находят следующим образом. Значения FilterOffsetA и FilterOffsetB – это значения переменных, определенных в п. 7.4.3 для секции, которая содержит макроблок с образцом q_0 :

$$\text{indexA} = \text{Clip3}(0, 51, qP_{av} + \text{FilterOffsetA}); \quad (8-463)$$

$$\text{indexB} = \text{Clip3}(0, 51, qP_{av} + \text{FilterOffsetB}). \quad (8-464)$$

Переменных α' и β' определены в таблице 8-16 в зависимости от значений indexA и indexB . В зависимости от chromaEdgeFlag пороговые переменные α и β находят следующим образом:

- Если chromaEdgeFlag равно 0:

$$\alpha = \alpha' * (1 \ll (\text{BitDepth}_Y - 8)); \quad (8-465)$$

$$\beta = \beta' * (1 \ll (\text{BitDepth}_Y - 8)). \quad (8-466)$$

- Иначе (chromaEdgeFlag равно 1):

$$\alpha = \alpha' * (1 \ll (\text{BitDepth}_C - 8)); \quad (8-467)$$

$$\beta = \beta' * (1 \ll (\text{BitDepth}_C - 8)). \quad (8-468)$$

Переменную filterSamplesFlag находят как:

$$\text{filterSamplesFlag} = (\text{bS} \neq 0 \ \&\& \ \text{Abs}(p_0 - q_0) < \alpha \ \&\& \ \text{Abs}(p_1 - p_0) < \beta \ \&\& \ \text{Abs}(q_1 - q_0) < \beta). \quad (8-469)$$

Таблица 8-16 – Вывод пороговых переменных α' и β' по indexA и indexB в зависимости от смещения

	indexA (для α') или indexB (для β')																									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
α'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	5	6	7	8	9	10	12	13
β'	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	3	3	3	3	4	4	4

Таблица 8-16 (окончание) – Вывод пороговых переменных α' и β' по indexA и indexB в зависимости от смещения

	indexA (для α') или indexB (для β')																														
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51					
α'	15	17	20	22	25	28	32	36	40	45	50	56	63	71	80	90	101	113	127	144	162	182	203	226	255	255					
β'	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17	17	18	18					

8.7.2.3 Процесс фильтрации краев при значении bS, меньшем 4

Входы в этот процесс – вход значений образцов p_i и q_i ($i = 0..2$) одного множества образцов вдоль края, предназначенного для фильтрации, переменная chromaEdgeFlag, bS, β и indexA для множества входов образцов, как это определено в 8.7.2.

Выходы этого процесса – результирующие значения фильтрации образцов p'_i и q'_i ($i = 0..2$) для множества значений входов образцов.

Результирующие значения фильтрации образцов p'_0 и q'_0 находят как:

$$\Delta = \text{Clip3}(-t_c, t_c, (((q_0 - p_0) \ll 2) + (p_1 - q_1) + 4) \gg 3); \quad (8-470)$$

$$p'_0 = \text{Clip1}(p_0 + \Delta); \quad (8-471)$$

$$q'_0 = \text{Clip1}(q_0 - \Delta), \quad (8-472)$$

где порог t_c определяют следующим образом:

– если chromaEdgeFlag равно 0:

$$t_c = t_{c0} + ((a_p < \beta) ? 1 : 0) + ((a_q < \beta) ? 1 : 0); \quad (8-473)$$

– иначе (chromaEdgeFlag равно 1):

$$t_c = t_{c0} + 1. \quad (8-474)$$

Переменная t'_{c0} определена в таблице 8-17 в зависимости от значений indexA и bS. В зависимости от chromaEdgeFlag соответствующую пороговую переменную t_{c0} находят следующим образом:

– если chromaEdgeFlag равно 0:

$$t_{c0} = t'_{c0} * (1 \ll (\text{BitDepth}_Y - 8)); \quad (8-475)$$

– иначе (chromaEdgeFlag равно 1):

$$t_{c0} = t'_{c0} * (1 \ll (\text{BitDepth}_C - 8)). \quad (8-476)$$

Положим, что a_p и a_q – две пороговые переменные, определяемые как:

$$a_p = \text{Abs}(p_2 - p_0); \quad (8-477)$$

$$a_q = \text{Abs}(q_2 - q_0). \quad (8-478)$$

Результирующие значения фильтрации образца p'_1 находят следующим образом.

- Если `chromaEdgeFlag` равно 0, а a_p меньше чем β :

$$p'_1 = p_1 + \text{Clip3}(-t_{c0}, t_{c0}, (p_2 + ((p_0 + q_0 + 1) \gg 1) - (p_1 \ll 1)) \gg 1). \quad (8-479)$$

- Иначе (`chromaEdgeFlag` равно 1 или a_p больше или равно β):

$$p'_1 = p_1. \quad (8-480)$$

Результирующие значения фильтрации образца q'_1 находят следующим образом.

- Если `chromaEdgeFlag` равно 0, а a_q меньше чем β :

$$q'_1 = q_1 + \text{Clip3}(-t_{c0}, t_{c0}, (q_2 + ((p_0 + q_0 + 1) \gg 1) - (q_1 \ll 1)) \gg 1). \quad (8-481)$$

- Иначе (`chromaEdgeFlag` равно 1 или a_q больше или равно β):

$$q'_1 = q_1. \quad (8-482)$$

Результирующие значения фильтрации образцов p'_2 и q'_2 всегда устанавливают равными входным образцам p_2 и q_2 :

$$p'_2 = p_2; \quad (8-483)$$

$$q'_2 = q_2. \quad (8-484)$$

Таблица 8-17 – Значение переменной t_{c0} как функции `indexA` и `bS`

	IndexA																									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
<code>bS = 1</code>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
<code>bS = 2</code>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
<code>bS = 3</code>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Таблица 8-17 (окончание) – Значение переменной t_{c0} как функции `indexA` и `bS`

	IndexA																									
	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
<code>bS = 1</code>	1	1	1	1	1	1	1	2	2	2	2	3	3	3	4	4	4	5	6	6	7	8	9	10	11	13
<code>bS = 2</code>	1	1	1	1	1	2	2	2	2	3	3	3	4	4	5	5	6	7	8	8	10	11	12	13	15	17
<code>bS = 3</code>	1	2	2	2	2	3	3	3	4	4	4	5	6	6	7	8	9	10	11	13	14	16	18	20	23	25

8.7.2.4 Процесс фильтрации краев при значении bS, равном 4

Входы в этот процесс – значения входов образцов p_i и q_i ($i = 0..3$) одного множества образцов вдоль края, предназначенного для фильтрации, переменная `chromaEdgeFlag` и значения пороговых переменных α и β для этого множества образцов, как это определено в п. 8.7.2.

Выходы этого процесса – результирующие значения фильтрации образцов p'_i и q'_i ($i = 0..2$) для этого множества значений входов образцов.

Положим, что a_p и a_q – две пороговые переменные, определенные в п. 8.7.2.3, в уравнениях 8-477 и 8-478, соответственно.

Результирующие значения фильтрации образца p'_i ($i = 0..2$) находят следующим образом.

- Если `chromaEdgeFlag` равно 0 и выполняются следующие условия:

$$a_p < \beta \ \&\& \ \text{Abs}(p_0 - q_0) < ((\alpha \gg 2) + 2), \quad (8-485)$$

то переменные p'_0 , p'_1 , и p'_2 находят как:

$$p'_0 = (p_2 + 2*p_1 + 2*p_0 + 2*q_0 + q_1 + 4) \gg 3; \quad (8-486)$$

$$p'_1 = (p_2 + p_1 + p_0 + q_0 + 2) \gg 2; \quad (8-487)$$

$$p'_2 = (2*p_3 + 3*p_2 + p_1 + p_0 + q_0 + 4) \gg 3. \quad (8-488)$$

- Иначе (`chromaEdgeFlag` равно 1 или условия в 8-485 не выполняются) переменные p'_0 , p'_1 и p'_2 находят как:

$$p'_0 = (2*p_1 + p_0 + q_1 + 2) \gg 2; \quad (8-489)$$

$$p'_1 = p_1; \quad (8-490)$$

$$p'_2 = p_2. \quad (8-491)$$

Результирующие значения фильтрации образцов q'_i ($i = 0..2$) находят следующим образом.

- Если `chromaEdgeFlag` равно 0 и выполняются следующие условия:

$$a_q < \beta \ \&\& \ \text{Abs}(p_0 - q_0) < ((\alpha \gg 2) + 2), \quad (8-492)$$

то переменные q'_0 , q'_1 и q'_2 находят как:

$$q'_0 = (p_1 + 2*p_0 + 2*q_0 + 2*q_1 + q_2 + 4) \gg 3; \quad (8-493)$$

$$q'_1 = (p_0 + q_0 + q_1 + q_2 + 2) \gg 2; \quad (8-494)$$

$$q'_2 = (2*q_3 + 3*q_2 + q_1 + q_0 + p_0 + 4) \gg 3. \quad (8-495)$$

- Иначе (`chromaEdgeFlag` равно 1 или условия в 8-492 не выполняются) переменные q'_0 , q'_1 и q'_2 находят как:

$$q'_0 = (2*q_1 + q_0 + p_1 + 2) \gg 2; \quad (8-496)$$

$$q'_1 = q_1; \quad (8-497)$$

$$q'_2 = q_2. \quad (8-498)$$

9 Процесс анализа

Входы в этот процесс – биты из RBSP.

Выходы этого процесса – значения элементов синтаксиса.

Процесс активируют, если дескрипторы элемента синтаксиса в таблицах синтаксиса в п. 7.3 равны $ue(v)$, $me(v)$, $se(v)$, $te(v)$ (см. п. 9.1), $ce(v)$ (см. п. 9.2) или $ae(v)$ (см. п. 9.3).

9.1 Процесс анализа кодов Exp-Golomb

Процесс активируют, если дескрипторы элементов синтаксиса в таблицах синтаксиса в п. 7.3 равны $ue(v)$, $me(v)$, $se(v)$, или $te(v)$. Для элементов синтаксиса в пп. 7.3.4 и 7.3.5 этот процесс активируют, только если `entropy_coding_mode_flag` равно 0.

Входы в этот процесс – биты из RBSP.

Выходы этого процесса – значения элементов синтаксиса.

Элементы синтаксиса, закодированные как $ue(v)$, $me(v)$ или $se(v)$, – это Exp-Golomb-coded. Элементы синтаксиса, закодированные как $te(v)$ – это усеченные элементы Exp-Golomb-coded. Процесс анализа этих элементов синтаксиса начинают с чтения стартовых битов в текущем местоположении в потоке битов до и включая появление первого ненулевого бита, и подсчитывая число продвинутых битов, которые равны 0. Этот процесс описывается следующим образом:

```

leadingZeroBits = -1;
for( b = 0; !b; leadingZeroBits++ )
    b = read_bits( 1 ).
    
```

Переменную `codeNum` определяют как:

$$\text{codeNum} = 2^{\text{leadingZeroBits}} - 1 + \text{read_bits}(\text{leadingZeroBits}),$$

где возвратное значение от `read_bits(leadingZeroBits)` интерпретируют как бинарное представление целого без знака с старшим значащим битом, записанным первым.

В таблице 9-1 показана структура кода Exp-Golomb с разделением строки битов на биты "префикса" и "суффикса". Биты "префикса" – это те биты, которые проанализированы в вышеуказанном псевдокоде для вычисления `leadingZeroBits` и показаны в виде 0 или 1 в строке битов столбца таблицы 9-1. Биты "суффикса" – это те биты, которые проанализированы при вычислении `codeNum` и показаны в виде x_i в таблице 9-1, с i в диапазоне от 0 до `leadingZeroBits - 1` включительно. Каждый x_i может принимать значения 0 или 1.

Таблица 9-1 – Строки битов с битами "префикса" и "суффикса" и с присвоением диапазону кода `codeNum` (для информации)

Форма строки битов	Диапазон <code>codeNum</code>
1	0
0 1 x_0	1–2
0 0 1 $x_1 x_0$	3–6
0 0 0 1 $x_2 x_1 x_0$	7–14
0 0 0 0 1 $x_3 x_2 x_1 x_0$	15–30
0 0 0 0 0 1 $x_4 x_3 x_2 x_1 x_0$	31–62
...	...

Таблица 9-2 иллюстрирует однозначное присвоение строк битов значениям codeNum.

Таблица 9-2 – Строки битов Exp-Golomb и codeNum в явной форме, использованные как ue(v) (для информации)

Строка битов	codeNum
1	0
0 1 0	1
0 1 1	2
0 0 1 0 0	3
0 0 1 0 1	4
0 0 1 1 0	5
0 0 1 1 1	6
0 0 0 1 0 0 0	7
0 0 0 1 0 0 1	8
0 0 0 1 0 1 0	9
...	...

В зависимости от дескриптора, значение элемента синтаксиса находят следующим образом.

- Если элемент синтаксиса кодирован как ue(v), значение этого элемента синтаксиса равно codeNum.
- Иначе, если элемент синтаксиса кодирован как se(v), значение этого элемента синтаксиса находят, активируя процесс отображения для имеющих знак кодов Exp-Golomb, как это определено в п. 9.1.1 с codeNum в качестве входа.
- Иначе, если элемент синтаксиса кодирован как me(v), значение элемента синтаксиса находят, активируя процесс отображения для кодированного образца блока, как это определено в п. 9.1.2 с codeNum в качестве входа.
- Иначе (элемент синтаксиса кодирован как te(v)) диапазон возможных значений для элемента синтаксиса определяется в первую очередь. Диапазон этого элемента синтаксиса может лежать между 0 и x, с x больше или равным 1, и его используют для вывода значения элемента синтаксиса следующим образом:
 - если x больше чем 1, codeNum и значение элемента синтаксиса находится тем же способом, что и для элементов синтаксиса, кодированных как ue(v);
 - иначе (x равно 1) процесс анализа codeNum, который равен значению элемента синтаксиса, задан процессом, эквивалентным следующему:

```
b = read_bits(1)
codeNum = !b.
```

9.1.1 Процесс отображения кодов Exp-Golomb со знаком

Вход в этот процесс – codeNum, как это определено в п. 9.1.

Выход этого процесса – значение элемента синтаксиса, кодированного как se(v).

Элемент синтаксиса присваивают параметру codeNum упорядочением элементов синтаксиса по их абсолютным значениям в возрастающем порядке и представляя положительную величину для заданного абсолютного значения наименьшим codeNum. В таблице 9-3 приведены правила присвоения.

Таблица 9-3 – Присвоение элементов синтаксиса значениям codeNum для элементов синтаксиса se(v), кодированных кодом Exp-Golomb со знаком

codeNum	Значение элемента синтаксиса
0	0
1	1
2	-1
3	2
4	-2
5	3
6	-3
k	$(-1)^{k+1} \text{Ceil}(k \div 2)$

9.1.2 Процесс отображения кодированного блока образца

Вход в этот процесс – codeNum, как это определено в п. 9.1.

Выход этого процесса – значение элементов синтаксиса coded_block_pattern, кодированных как me(v).

В таблице 9-4 показано присвоение coded_block_pattern параметру codeNum в зависимости от того, является ли режим предсказания макроблока Intra_4x4, Intra_8x8 или Inter.

Таблица 9-4 – Присвоение codeNum параметру coded_block_pattern для режимов предсказания макроблока

(a) chroma_format_idc не равно 0

codeNum	coded_block_pattern	
	Intra_4x4, Intra_8x8	Inter
0	47	0
1	31	16
2	15	1
3	0	2
4	23	4
5	27	8
6	29	32
7	30	3
8	7	5
9	11	10
10	13	12
11	14	15
12	39	47
13	43	7
14	45	11
15	46	13

codeNum	coded_block_pattern	
	Intra_4x4, Intra_8x8	Inter
16	16	14
17	3	6
18	5	9
19	10	31
20	12	35
21	19	37
22	21	42
23	26	44
24	28	33
25	35	34
26	37	36
27	42	40
28	44	39
29	1	43
30	2	45
31	4	46
32	8	17
33	17	18
34	18	20
35	20	24
36	24	19
37	6	21
38	9	26
39	22	28
40	25	23
41	32	27
42	33	29
43	34	30
44	36	22
45	40	25
46	38	38
47	41	41

(b) chroma_format_idc равно 0

codeNum	coded_block_pattern	
	Intra_4x4, Intra_8x8	Inter
0	15	0
1	0	1
2	7	2
3	11	4
4	13	8
5	14	3
6	3	5
7	5	10
8	10	12
9	12	15
10	1	7
11	2	11
12	4	13
13	8	14
14	6	6
15	9	9

9.2 Процесс анализа CAVLC для уровней коэффициентов преобразования

Процесс активируют, если элементы синтаксического анализа с дескриптором равны $se(v)$ (согласно п. 7.3.5.3.1) и если `entropy_coding_mode_flag` равно 0.

Входы в этот процесс – биты из данных секции, максимальное число ненулевых уровней коэффициентов преобразования `maxNumCoeff`, индекс блока яркости `luma4x4BlkIdx` или индекс блока цветности `chroma4x4BlkIdx` текущего блока уровней коэффициентов преобразования.

Выход этого процесса – список `coeffLevel`, содержащий уровни коэффициентов преобразования блока яркости с индексом блока `luma4x4BlkIdx` или блока цветности с индексом блока `chroma4x4BlkIdx`.

Процесс описывают в следующем порядке.

1. Все уровни коэффициентов преобразования, с индексами от 0 до `maxNumCoeff - 1`, в списке `coeffLevel` устанавливают равными 0.
2. Общее число ненулевых уровней коэффициентов преобразования `TotalCoeff(coeff_token)` и число концевых уровней коэффициентов преобразования `TrailingOnes(coeff_token)` находят из анализа `coeff_token` (см. п. 9.2.1) следующим образом.
 - Если число ненулевых уровней коэффициентов преобразования `TotalCoeff(coeff_token)` равно 0, список `coeffLevel`, содержащий значения 0, возвращают и далее не производят никаких действий.
 - Иначе выполняют следующие шаги.
 - a. Ненулевые уровни коэффициентов преобразования находят анализом `trailing_ones_sign_flag`, `level_prefix`, и `level_suffix` (см. п. 9.2.2).
 - b. Однократные проходы нулевых уровней коэффициентов преобразования перед каждым ненулевым уровнем коэффициентов преобразования находят анализом `total_zeros` и `run_before` (см. п. 9.2.3).
 - c. Информацию об уровнях и проходах собирают в список `coeffLevel` (см. п. 9.2.4).

9.2.1 Процесс анализа общего числа уровней коэффициентов преобразования и конечных уровней коэффициентов преобразования

Входы в этот процесс – биты из данных секции, максимальное число ненулевых уровней коэффициентов преобразования `maxNumCoeff`, индекс блока яркости `luma4x4BlkIdx` или блока цветности `chroma4x4BlkIdx` текущего блока преобразования.

Выходы этого процесса – `TotalCoeff(coeff_token)` и `TrailingOnes(coeff_token)`.

Элемент синтаксиса `coeff_token` декодируют, используя одно из шести значений VLC, определенных в шести крайних правых столбцах таблицы 9-5. Каждое значение VLC определяет `TotalCoeff(coeff_token)` и `TrailingOnes(coeff_token)` для заданного кодового слова `coeff_token`. Выбор VLC зависит от переменной `nC`, которую находят следующим образом.

- Если процесс анализа CAVLC активируют для `ChromaDCLevel`, то `nC` находят следующим образом:
 - если `chroma_format_idc` равно 1, `nC` устанавливается равным -1;
 - иначе, если `chroma_format_idc` равно 2, `nC` устанавливается равным -2;
 - иначе (`chroma_format_idc` равно 3) `nC` устанавливается равным 0.
- Иначе применяют следующее.
 - Если процесс анализа CAVLC активируют для `Intra16x16DCLevel`, то `luma4x4BlkIdx` устанавливают равным 0.
 - Переменные `blkA` и `blkB` находят следующим образом.
 - Если процесс анализа CAVLC активируют для `Intra16x16DCLevel`, `Intra16x16ACLevel` или `LumaLevel`, то активируют процесс, определенный в п. 6.4.8.3, с `luma4x4BlkIdx` в качестве входа, и присваивают выходу значения `mbAddrA`, `mbAddrB`, `luma4x4BlkIdxA` и `luma4x4BlkIdxB`. Блоку яркости 4x4, определенному `mbAddrA\luma4x4BlkIdxA`, присваивают `blkA`, а блоку яркости 4x4, определенному `mbAddrB\luma4x4BlkIdxB`, присваивают `blkB`.
 - Иначе (процесс анализа CAVLC активируют для `ChromaACLevel`) процесс, определенный в п. 6.4.8.4, активируют с `chroma4x4BlkIdx` в качестве входа, а выходу присваивают значения `mbAddrA`, `mbAddrB`, `chroma4x4BlkIdxA` и `chroma4x4BlkIdxB`. Блоку цветности 4x4, определенному `mbAddrA\iCbCr\chroma4x4BlkIdxA`, присваивают `blkA`, а блоку цветности 4x4, определенному `mbAddrB\iCbCr\chroma4x4BlkIdxB`, присваивают `blkB`.
 - Допустим, что `nA` и `nB` – число ненулевых уровней коэффициентов преобразования (заданных `TotalCoeff(coeff_token)`) в блоке уровней коэффициентов преобразования `blkA`, расположенном слева от текущего блока, и в блоке уровней коэффициентов преобразования `blkB`, расположенном выше текущего блока, соответственно.
 - С заменой `N` на `A` и `B` в `mbAddrN`, `blkN` и `nN` применяют следующее.
 - Если любое из следующих условий истина, то `nN` устанавливают равным 0:
 - `mbAddrN` недоступно;
 - текущий макроблок кодирован с использованием режима `Intra` предсказания, `constrained_intra_pred_flag` равно 1, а `mbAddrN` кодирован с использованием режима `Inter` предсказания, а также используют данные разделения секции (`nal_unit_type` находится в диапазоне от 2 до 4 включительно);
 - макроблок `mbAddrN` имеет значения `mb_type`, равные `P_Skip` или `B_Skip`;
 - все остаточные AC уровни коэффициентов преобразования смежного блока `blkN` равны 0, благодаря соответствующим битам `CodedBlockPatternLuma` или `CodedBlockPatternChroma`, равным 0.
 - Иначе, если `mbAddrN` – макроблок `I_PCM`, то `nN` устанавливают равным 16.
 - Иначе `nN` устанавливают равным значению `TotalCoeff(coeff_token)` смежного блока `blkN`.

ПРИМЕЧАНИЕ 1. – Значения `nA` и `nB`, которые находят, используя `TotalCoeff(coeff_token)`, не включают DC уровни коэффициентов преобразования в макроблоках `Intra_16x16` или DC уровни коэффициентов преобразования в блоках цветности, т. к. эти уровни коэффициентов преобразования декодируют отдельно. Если блоки выше или слева принадлежат макроблоку `Intra_16x16` или являются блоками цветности, то `nA` и `nB` – это число декодированных ненулевых AC уровней коэффициентов преобразования.

ПРИМЕЧАНИЕ 2. – При анализе `Intra16x16DCLevel` значения `nA` и `nB` основаны на числе ненулевых уровней коэффициентов преобразования в смежных блоках 4x4 и на числе ненулевых DC уровней коэффициентов преобразования в смежных блоках 16x16.

- При заданных значениях nA и nB, переменную nC находят следующим образом.
 - Если доступны оба значения mbAddrA и mbAddrB, переменную nC устанавливают равной $(nA + nB + 1) \gg 1$.
 - Иначе (mbAddrA недоступно или mbAddrB недоступно) переменную nC устанавливают равной $nA + nB$.

Результирующее значение TotalCoeff(coeff_token) от декодирования coeff_token должно лежать в диапазоне от 0 до maxNumCoeff включительно.

Таблица 9-5 – Отображение coeff_token в TotalCoeff(coeff_token) и в TrailingOnes(coeff_token)

TrailingOnes (coeff_token)	TotalCoeff (coeff_token)	$0 \leq nC < 2$	$2 \leq nC < 4$	$4 \leq nC < 8$	$8 \leq nC$	$nC == -1$	$nC == -2$
0	0	1	11	1111	0000 11	01	1
0	1	0001 01	0010 11	0011 11	0000 00	0001 11	0001 111
1	1	01	10	1110	0000 01	1	01
0	2	0000 0111	0001 11	0010 11	0001 00	0001 00	0001 110
1	2	0001 00	0011 1	0111 1	0001 01	0001 10	0001 101
2	2	001	011	1101	0001 10	001	001
0	3	0000 0011 1	0000 111	0010 00	0010 00	0000 11	0000 0011 1
1	3	0000 0110	0010 10	0110 0	0010 01	0000 011	0001 100
2	3	0000 101	0010 01	0111 0	0010 10	0000 010	0001 011
3	3	0001 1	0101	1100	0010 11	0001 01	0000 1
0	4	0000 0001 11	0000 0111	0001 111	0011 00	0000 10	0000 0011 0
1	4	0000 0011 0	0001 10	0101 0	0011 01	0000 0011	0000 0010 1
2	4	0000 0101	0001 01	0101 1	0011 10	0000 0010	0001 010
3	4	0000 11	0100	1011	0011 11	0000 000	0000 01
0	5	0000 0000 111	0000 0100	0001 011	0100 00	-	0000 0001 11
1	5	0000 0001 10	0000 110	0100 0	0100 01	-	0000 0001 10
2	5	0000 0010 1	0000 101	0100 1	0100 10	-	0000 0010 0
3	5	0000 100	0011 0	1010	0100 11	-	0001 001
0	6	0000 0000 0111 1	0000 0011 1	0001 001	0101 00	-	0000 0000 111
1	6	0000 0000 110	0000 0110	0011 10	0101 01	-	0000 0000 110
2	6	0000 0001 01	0000 0101	0011 01	0101 10	-	0000 0001 01
3	6	0000 0100	0010 00	1001	0101 11	-	0001 000
0	7	0000 0000 0101 1	0000 0001 111	0001 000	0110 00	-	0000 0000 0111
1	7	0000 0000 0111 0	0000 0011 0	0010 10	0110 01	-	0000 0000 0110
2	7	0000 0000 101	0000 0010 1	0010 01	0110 10	-	0000 0000 101
3	7	0000 0010 0	0001 00	1000	0110 11	-	0000 0001 00
0	8	0000 0000 0100 0	0000 0001 011	0000 1111	0111 00	-	0000 0000 0011 1
1	8	0000 0000 0101 0	0000 0001 110	0001 110	0111 01	-	0000 0000 0101
2	8	0000 0000 0110 1	0000 0001 101	0001 101	0111 10	-	0000 0000 0100
3	8	0000 0001 00	0000 100	0110 1	0111 11	-	0000 0000 100
0	9	0000 0000 0011 11	0000 0000 1111	0000 1011	1000 00	-	

TrailingOnes (coeff_token)	TotalCoeff (coeff_token)	$0 \leq nC < 2$	$2 \leq nC < 4$	$4 \leq nC < 8$	$8 \leq nC$	$nC == -1$	$nC == -2$
1	9	0000 0000 0011 10	0000 0001 010	0000 1110	1000 01	-	
2	9	0000 0000 0100 1	0000 0001 001	0001 010	1000 10	-	
3	9	0000 0000 100	0000 0010 0	0011 00	1000 11	-	
0	10	0000 0000 0010 11	0000 0000 1011	0000 0111 1	1001 00	-	
1	10	0000 0000 0010 10	0000 0000 1110	0000 1010	1001 01	-	
2	10	0000 0000 0011 01	0000 0000 1101	0000 1101	1001 10	-	
3	10	0000 0000 0110 0	0000 0001 100	0001 100	1001 11	-	
0	11	0000 0000 0001 111	0000 0000 1000	0000 0101 1	1010 00	-	
1	11	0000 0000 0001 110	0000 0000 1010	0000 0111 0	1010 01	-	
2	11	0000 0000 0010 01	0000 0000 1001	0000 1001	1010 10	-	
3	11	0000 0000 0011 00	0000 0001 000	0000 1100	1010 11	-	
0	12	0000 0000 0001 011	0000 0000 0111 1	0000 0100 0	1011 00	-	
1	12	0000 0000 0001 010	0000 0000 0111 0	0000 0101 0	1011 01	-	
2	12	0000 0000 0001 101	0000 0000 0110 1	0000 0110 1	1011 10	-	
3	12	0000 0000 0010 00	0000 0000 1100	0000 1000	1011 11	-	
0	13	0000 0000 0000 1111	0000 0000 0101 1	0000 0011 01	1100 00	-	
1	13	0000 0000 0000 001	0000 0000 0101 0	0000 0011 1	1100 01	-	
2	13	0000 0000 0001 001	0000 0000 0100 1	0000 0100 1	1100 10	-	
3	13	0000 0000 0001 100	0000 0000 0110 0	0000 0110 0	1100 11	-	
0	14	0000 0000 0000 1011	0000 0000 0011 1	0000 0010 01	1101 00	-	
1	14	0000 0000 0000 1110	0000 0000 0010 11	0000 0011 00	1101 01	-	
2	14	0000 0000 0000 1101	0000 0000 0011 0	0000 0010 11	1101 10	-	
3	14	0000 0000 0001 000	0000 0000 0100 0	0000 0010 10	1101 11	-	
0	15	0000 0000 0000 0111	0000 0000 0010 01	0000 0001 01	1110 00	-	
1	15	0000 0000 0000 1010	0000 0000 0010 00	0000 0010 00	1110 01	-	
2	15	0000 0000 0000 1001	0000 0000 0010 10	0000 0001 11	1110 10	-	
3	15	0000 0000 0000 1100	0000 0000 0000 1	0000 0001 10	1110 11	-	
0	16	0000 0000 0000 0100	0000 0000 0001 11	0000 0000 01	1111 00	-	
1	16	0000 0000 0000 0110	0000 0000 0001 10	0000 0001 00	1111 01	-	
2	16	0000 0000 0000 0101	0000 0000 0001 01	0000 0000 11	1111 10	-	
3	16	0000 0000 0000 1000	0000 0000 0001 00	0000 0000 10	1111 11	-	

9.2.2 Процесс анализа информационного уровня

Входы в этот процесс – биты из данных секции, число ненулевых уровней коэффициентов преобразования TotalCoeff(coeff_token) и число концевых ненулевых уровней коэффициентов преобразования TrailingOnes(coeff_token).

Выход этого процесса – список с именами уровней, в котором содержатся уровни коэффициентов преобразования.

Первоначально индекс i устанавливают равным 0. Затем следующую процедуру итеративно применяют $\text{TrailingOnes}(\text{coeff_token})$ раз, чтобы декодировать концевые уровни коэффициентов преобразования (если таковые есть).

- 1-битовый элемент синтаксиса `trailing_ones_sign_flag` декодируют и оценивают следующим образом:
 - если `trailing_ones_sign_flag` равно 0, то значение +1 присваивают `level[i]`;
 - иначе (`trailing_ones_sign_flag` равно 1) значение –1 присваивают `level[i]`.
- Индекс i возрастает на 1.

После декодирования концевого уровня коэффициентов преобразования переменную `suffixLength` инициализируют следующим образом.

- Если $\text{TotalCoeff}(\text{coeff_token})$ больше 10, а $\text{TrailingOnes}(\text{coeff_token})$ меньше 3, `suffixLength` устанавливают равным 1.
- Иначе ($\text{TotalCoeff}(\text{coeff_token})$ меньше или равно 10 или $\text{TrailingOnes}(\text{coeff_token})$ равно 3), `suffixLength` устанавливают равным 0.

Следующую процедуру применяют итеративно ($\text{TotalCoeff}(\text{coeff_token}) - \text{TrailingOnes}(\text{coeff_token})$) раз, чтобы декодировать остаток уровней (если таковые есть).

- Элемент синтаксиса `level_prefix` декодируют, как это описано в п. 9.2.2.1.
- Переменную `levelSuffixSize` устанавливают равной переменной `suffixLength`, за исключением следующих двух случаев.
- Если `level_prefix` равно 14 и `suffixLength` равно 0, `levelSuffixSize` устанавливают равным 4.
- Если `level_prefix` больше или равно 15, `levelSuffixSize` устанавливают равным `level_prefix - 3`.
- Элемент синтаксиса `level_suffix` декодируют следующим образом:
 - если `levelSuffixSize` больше 0, элемент синтаксиса `level_suffix` декодируют как целое без знака представление $u(v)$ с `levelSuffixSize` битов;
 - иначе (`levelSuffixSize` равно 0) элемент синтаксиса `level_suffix` принимается равным 0.
- Переменную `levelCode` устанавливают равной $(\text{Min}(15, \text{level_prefix}) \ll \text{suffixLength}) + \text{level_suffix}$.
- Если `level_prefix` больше или равно 15, а `suffixLength` равно 0, `levelCode` возрастает на 15.
- Если `level_prefix` больше или равно 16, `levelCode` увеличивается на $(1 \ll (\text{level_prefix} - 3)) - 4096$.
- Если индекс i равен $\text{TrailingOnes}(\text{coeff_token})$, а $\text{TrailingOnes}(\text{coeff_token})$ меньше 3, `levelCode` возрастает на 2.
- Переменную `level[i]` находят следующим образом:
 - если `levelCode` – четное число, то значение $(\text{levelCode} + 2) \gg 1$ присваивают `level[i]`;
 - иначе (`levelCode` нечетное число) значение $(-\text{levelCode} - 1) \gg 1$ присваивают `level[i]`.
- Если `suffixLength` равно 0, `suffixLength` устанавливают равным 1.
- Если абсолютное значение `level[i]` больше $(3 \ll (\text{suffixLength} - 1))$, а `suffixLength` меньше 6, `suffixLength` возрастает на 1.
- Индекс i возрастает на 1.

9.2.2.1 Процесс анализа для `level_prefix`

Входами в этот процесс являются биты из данных секции.

Выходом этого процесса является `level_prefix`.

Процесс анализа для элемента синтаксиса состоит из чтения битов начиная с текущего местоположения в потоке битов до первого ненулевого бита, включая его, и подсчета количества ведущих битов, равных 0. Этот процесс описывается следующим образом:

```
leadingZeroBits = -1
for( b = 0; !b; leadingZeroBits++ )
    b = read_bits( 1 )
level_prefix = leadingZeroBits.
```

Таблица 9-6 является таблицей кодовых слов для level_prefix.

Таблица 9-6 – Таблица кодовых слов для level_prefix (для информации)

level_prefix	Строка битов
0	1
1	01
2	001
3	0001
4	0000 1
5	0000 01
6	0000 001
7	0000 0001
8	0000 0000 1
9	0000 0000 01
10	0000 0000 001
11	0000 0000 0001
12	0000 0000 0000 1
13	0000 0000 0000 01
14	0000 0000 0000 001
15	0000 0000 0000 0001
...	...

9.2.3 Процесс анализа информации проходов

Входы в этот процесс – биты из данных секции, число ненулевых уровней коэффициентов преобразования TotalCoeff(coeff_token) и максимальное число ненулевых уровней коэффициентов преобразования maxNumCoeff.

Выход этого процесса – список проходов нулевых уровней коэффициентов преобразования, которые предшествовали ненулевым уровням коэффициентов преобразования в результате проходов.

Первоначально индекс i устанавливают равным 0.

Переменную zerosLeft находят следующим образом.

- Если число ненулевых уровней коэффициентов преобразования TotalCoeff(coeff_token) равно максимальному числу ненулевых уровней коэффициентов преобразования maxNumCoeff, то переменную zerosLeft устанавливают равной 0.
- Иначе (число ненулевых уровней коэффициентов преобразования TotalCoeff(coeff_token) меньше максимального числа ненулевых уровней коэффициентов преобразования maxNumCoeff), total_zeros декодируют, а zerosLeft устанавливают равным его значению.

Значение VLC, которое используют для декодирования total_zeros, находят следующим образом.

Если maxNumCoeff равно 4, используют одно из значений VLC, определенных в таблице 9-9(a).

- Иначе, если maxNumCoeff равно 8, используют одно из значений VLC, определенных в таблице 9-9 (b).
- Иначе (maxNumCoeff не равно 4 и не равно 8) используют VLC из таблицы 9-7 и таблицы 9-8.

Следующую процедуру далее применяют итеративно (TotalCoeff(coeff_token) – 1) раз.

- Переменную run[i] находят следующим образом.
 - Если zerosLeft больше нуля, то значение run_before декодируют, основываясь на таблице 9-10 и zerosLeft. Значение run[i] устанавливают равным run_before.
 - Иначе (zerosLeft равно 0) run[i] устанавливают равным 0.
 - Значение run[i] вычитают из zerosLeft, а результат присваивают параметру zerosLeft. Результат вычитания должен быть больше или равен 0.
 - Индекс i возрастает на 1.
- Наконец, значение zerosLeft присваивают run[i].

Таблица 9-7 – Таблицы total_zeros для блоков 4x4 при TotalCoeff(coeff_token) от 1 до 7

total_zeros	TotalCoeff(coeff_token)						
	1	2	3	4	5	6	7
0	1	111	0101	0001 1	0101	0000 01	0000 01
1	011	110	111	111	0100	0000 1	0000 1
2	010	101	110	0101	0011	111	101
3	0011	100	101	0100	111	110	100
4	0010	011	0100	110	110	101	011
5	0001 1	0101	0011	101	101	100	11
6	0001 0	0100	100	100	100	011	010
7	0000 11	0011	011	0011	011	010	0001
8	0000 10	0010	0010	011	0010	0001	001
9	0000 011	0001 1	0001 1	0010	0000 1	001	0000 00
10	0000 010	0001 0	0001 0	0001 0	0001	0000 00	
11	0000 0011	0000 11	0000 01	0000 1	0000 0		
12	0000 0010	0000 10	0000 1	0000 0			
13	0000 0001 1	0000 01	0000 00				
14	0000 0001 0	0000 00					
15	0000 0000 1						

Таблица 9-8 – Таблицы total_zeros для блоков 4x4 при TotalCoeff(coeff_token) от 8 до 15

total_zeros	TotalCoeff(coeff_token)							
	8	9	10	11	12	13	14	15
0	0000 01	0000 01	0000 1	0000	0000	000	00	0
1	0001	0000 00	0000 0	0001	0001	001	01	1
2	0000 1	0001	001	001	01	1	1	
3	011	11	11	010	1	01		
4	11	10	10	1	001			
5	10	001	01	011				
6	010	01	0001					
7	001	0000 1						
8	0000 00							

Таблица 9-9 – Таблицы total_zeros для DC блоков цветности 2x2

(а) DC блок цветности 2x2
(выборка цветности 4:2:0)

total_zeros	TotalCoeff(coeff_token)		
	1	2	3
0	1	1	1
1	01	01	0
2	001	00	
3	000		

(b) DC блок цветности 2x4 block
(выборка цветности 4:2:2)

total_zeros	TotalCoeff(coeff_token)						
	1	2	3	4	5	6	7
0	1	000	000	110	00	00	0
1	010	01	001	00	01	01	1
2	011	001	01	01	10	1	
3	0010	100	10	10	11		
4	0011	101	110	111			
5	0001	110	111				
6	0000 1	111					
7	0000 0						

Таблица 9-10 – Таблицы для run_before

run_before	zerosLeft						
	1	2	3	4	5	6	>6
0	1	1	11	11	11	11	111
1	0	01	10	10	10	000	110
2	–	00	01	01	011	001	101
3	–	–	00	001	010	011	100
4	–	–	–	000	001	010	011
5	–	–	–	–	000	101	010
6	–	–	–	–	–	100	001
7	–	–	–	–	–	–	0001
8		–	–	–	–	–	00001
9	–	–	–	–	–	–	000001
10	–	–	–	–	–	–	0000001
11	–	–	–	–	–	–	00000001
12	–	–	–	–	–	–	000000001
13	–	–	–	–	–	–	0000000001
14	–	–	–	–	–	–	00000000001

9.2.4 Объединение информации об уровне и проходе

Вход в этот процесс – список вызванных уровней коэффициентов преобразования, список вызванных проходов и число ненулевых уровней коэффициентов преобразования TotalCoeff(coeff_token).

Выход этого процесса – список coeffLevel уровней коэффициентов преобразования.

Переменную coeffNum устанавливают равной –1, а индекс i устанавливают равным (TotalCoeff(coeff_token) – 1). Следующую процедуру итеративно применяют TotalCoeff(coeff_token) раз:

- coeffNum возрастает на run[i] + 1;
- coeffLevel[coeffNum] устанавливают равным level[i];
- индекс i уменьшается на 1.

9.3 Процесс анализа САВАС данных секции

Процесс активируют при анализе элементов синтаксиса с дескриптором ae(v) (согласно пп. 7.3.4 и 7.3.5) и при entropy_coding_mode_flag равном 1.

Входы в этот процесс – запрос значения элемента синтаксиса и значений ранее анализированных элементов синтаксиса.

Выход этого процесса – значение элемента синтаксиса.

При начале анализа данных секции согласно п. 7.3.4 активируют процесс инициализации процесса анализа САВАС, как это определено в п. 9.3.1.

Анализ элементов синтаксиса производят следующим образом:

Для каждого запрошенного значения элемента синтаксиса находят бинаризацию, как описано в п. 9.3.2.

Бинаризация элемента синтаксиса и последовательность анализированных бинов определяет протекание процесса декодирования, как это описано в п. 9.3.3.

Для каждого бина в процессе бинаризации элемента синтаксиса, проиндексированного переменной binIdx, индекс контекста ctxIdx находят, как это определено в п. 9.3.3.1.

Для каждого ctxIdx активируют процесс арифметического декодирования, как это определено в п. 9.3.3.2.

Результирующую последовательность ($b_0 \dots b_{binIdx}$) проанализированных бинов сравнивают со множеством строк бинов, заданных процессом бинаризации после декодирования каждого бина. Если последовательность соответствует строке бинов в заданном множестве, то элементу синтаксиса присваивается соответствующее значение.

В случае если запрошенное значение элемента синтаксиса обработано для типа элемента синтаксиса mb_type, а декодированное значение типа mb_type равно I_PCM, устройство декодирования инициализируется после декодирования любого pcm_alignment_zero_bit и всех данных pcm_sample_luma и pcm_sample_chroma, как это определено в п. 9.3.1.2.

Весь процесс анализа САВАС проиллюстрирован на блок-схеме рисунка 9-1 с аббревиатурой SE для элемента синтаксиса.

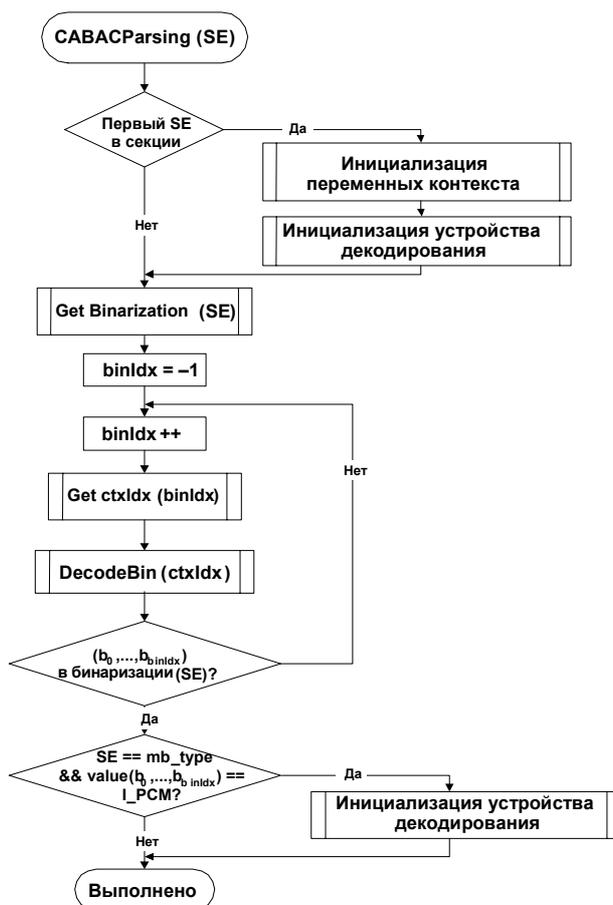


Рисунок 9-1 – Иллюстрация процесса анализа САВАС элемента синтаксиса SE (для информации)

9.3.1 Процесс инициализации

Выходы этого процесса – инициализированные внутренние переменные САВАС.

Процессы в пп. 9.3.1.1 и 9.3.1.2 активируют при начале анализа данных секции типа секций из п. 7.3.4.

Процесс в п. 9.3.1.2 также активируют после декодирования любого pcm_alignment_zero_bit и всех данных pcm_sample_luma и pcm_sample_chroma макроблока типа I_PCM.

9.3.3.1 Процесс инициализации контекстных переменных

Выходы этого процесса – инициализированные контекстные переменные CABAC, индексированные значением ctxIdx.

В таблицах 9-12–9-23 содержатся значения переменных *m* и *n*, использованных при инициализации контекстных переменных, которые присвоены всем элементам синтаксиса в пп. 7.3.4 и 7.3.5, за исключением флага конца секции (end-of-slice flag).

Для каждой контекстной переменной инициализируют две переменные – pStateIdx и valMPS.

ПРИМЕЧАНИЕ 1. – Переменная pStateIdx соответствует индексу состояния вероятности, а переменная valMPS – значению наиболее вероятного символа, как далее описано в п. 9.3.3.2.

Два присвоенных для инициализации значения, pStateIdx и valMPS, находят из SliceQP_Y, которое в свою очередь находят из уравнения 7-27. Заданы две таблицы входов (*m*, *n*):

1. preCtxState = Clip3(1, 126, ((m * Clip3(0, 51, SliceQP_Y)) >> 4) + n);
2. if(preCtxState <= 63) {
 - pStateIdx = 63 – preCtxState
 - valMPS = 0
- } else {
 - pStateIdx = preCtxState – 64
 - valMPS = 1
- }.

В таблице 9-11 перечислены значения ctxIdx, которые требуют инициализации для каждого из типов секций. В этой таблице также перечислены числа, которые включают значения *m* и *n*, необходимые для инициализации. Для типов секций P, SP и B инициализация также зависит от значения элемента синтаксиса cabac_init_idc. Заметим, что имя этого элемента синтаксиса не влияет на процесс инициализации.

Таблица 9-11 – Объединение ctxIdx и элементов синтаксиса для каждого типа секций в процессе инициализации

	Элемент синтаксиса	Таблица	Тип секции			
			SI	I	P, SP	B
slice_data()	mb_skip_flag	Таблица 9-13 Таблица 9-14			11–13	24–26
	mb_field_decoding_flag	Таблица 9-18	70–72	70–72	70–72	70–72
macroblock_layer()	mb_type	Таблица 9-12 Таблица 9-13 Таблица 9-14	0–10	3–10	14–20	27–35
	transform_size_8x8_flag	Таблица 9-16	na	399–401	399–401	399–401
	coded_block_pattern (luma)	Таблица 9-18	73–76	73–76	73–76	73–76
	coded_block_pattern (chroma)	Таблица 9-18	77–84	77–84	77–84	77–84
	mb_qp_delta	Таблица 9-17	60–63	60–63	60–63	60–63
mb_pred()	prev_intra4x4_pred_mode_flag	Таблица 9-17	68	68	68	68
	rem_intra4x4_pred_mode	Таблица 9-17	69	69	69	69
	prev_intra8x8_pred_mode_flag	Таблица 9-17	na	68	68	68
	rem_intra8x8_pred_mode	Таблица 9-17	na	69	69	69
	intra_chroma_pred_mode	Таблица 9-17	64–67	64–67	64–67	64–67

Таблица 9-11 – Объединение ctxIdx и элементов синтаксиса для каждого типа секций в процессе инициализации

	Элемент синтаксиса	Таблица	Тип секции			
			SI	I	P, SP	B
mb_pred() and sub_mb_pred()	ref_idx_10	Таблица 9-16			54–59	54–59
	ref_idx_11	Таблица 9-16				54–59
	mvd_10[][0]	Таблица 9-15			40–46	40–46
	mvd_11[][0]	Таблица 9-15				40–46
	mvd_10[][1]	Таблица 9-15			47–53	47–53
	mvd_11[][1]	Таблица 9-15				47–53
sub_mb_pred()	sub_mb_type	Таблица 9-13			21–23	36–39
		Таблица 9-14				
residual_block_cabac()	coded_block_flag	Таблица 9-18	85–104	85–104	85–104	85–104
	significant_coeff_flag[]	Таблица 9-19	105–165 277–337	105–165 277–337	105–165 277–337	105–165 277–337
		Таблица 9-22				
		Таблица 9-24				
last_significant_coeff_flag[]	Таблица 9-20	166–226 338–398	166–226 338–398	166–226 338–398	166–226 338–398	
	Таблица 9-23					
coeff_abs_level_minus1[]	Таблица 9-24	227–275	227–275	227–275	227–275	
	Таблица 9-24					

ПРИМЕЧАНИЕ 2. – Значение ctxIdx, равное 276, объединяют с end_of_slice_flag и с бином mb_type, который определяет тип макроблока I_PCM. Процесс декодирования, определенный в п. 9.3.3.2.4, применяют к ctxIdx, равному 276. Однако этот процесс декодирования можно также применять, используя процесс декодирования, определенный в п. 9.3.3.2.1. В этом случае первоначальные значения, объединенные с ctxIdx, равным 276, определены как pStatIdx = 63 и valMPS = 0, где pStatIdx = 63 представляет неадаптивное состояние вероятности.

Таблица 9-12 – Значения переменных m и n для ctxIdx от 0 до 10

Переменные инициализации	ctxIdx										
	0	1	2	3	4	5	6	7	8	9	10
M	20	2	3	20	2	3	-28	-23	-6	-1	7
N	-15	54	74	-15	54	74	127	104	53	54	51

Таблица 9-13 – Значения переменных m и n для ctxIdx от 11 до 23

Значение cabac_init_idc	Переменные инициализации	ctxIdx												
		11	12	13	14	15	16	17	18	19	20	21	22	23
0	m	23	23	21	1	0	-37	5	-13	-11	1	12	-4	17
	n	33	2	0	9	49	118	57	78	65	62	49	73	50
1	m	22	34	16	-2	4	-29	2	-6	-13	5	9	-3	10
	n	25	0	0	9	41	118	65	71	79	52	50	70	54
2	m	29	25	14	-10	-3	-27	26	-4	-24	5	6	-17	14
	n	16	0	0	51	62	99	16	85	102	57	57	73	57

Таблица 9-14 – Значения переменных m и n для ctxIdx от 24 до 39

Значение cabac_init_idc	Переменные инициализации	ctxIdx															
		24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
0	m	18	9	29	26	16	9	-46	-20	1	-13	-11	1	-6	-17	-6	9
	n	64	43	0	67	90	104	127	104	67	78	65	62	86	95	61	45
1	m	26	19	40	57	41	26	-45	-15	-4	-6	-13	5	6	-13	0	8
	n	34	22	0	2	36	69	127	101	76	71	79	52	69	90	52	43
2	m	20	20	29	54	37	12	-32	-22	-2	-4	-24	5	-6	-14	-6	4
	n	40	10	0	0	42	97	127	117	74	85	102	57	93	88	44	55

Таблица 9-15 – Значения переменных m и n для ctxIdx от 40 до 53

Значение cabac_init_idc	Переменные инициализации	ctxIdx													
		40	41	42	43	44	45	46	47	48	49	50	51	52	53
0	m	-3	-6	-11	6	7	-5	2	0	-3	-10	5	4	-3	0
	n	69	81	96	55	67	86	88	58	76	94	54	69	81	88
1	m	-2	-5	-10	2	2	-3	-3	1	-3	-6	0	-3	-7	-5
	n	69	82	96	59	75	87	100	56	74	85	59	81	86	95
2	m	-11	-15	-21	19	20	4	6	1	-5	-13	5	6	-3	-1
	n	89	103	116	57	58	84	96	63	85	106	63	75	90	101

Таблица 9-16 – Значения переменных m и n для ctxIdx от 54 до 59 и от 399 до 401

Значение sabac_init_idc	Переменные инициализации	ctxIdx								
		54	55	56	57	58	59	399	400	401
Секции I	m	na	na	na	na	na	na	31	31	25
	n	na	na	na	na	na	na	21	31	50
0	m	-7	-5	-4	-5	-7	1	12	11	14
	n	67	74	74	80	72	58	40	51	59
1	m	-1	-1	1	-2	-5	0	25	21	21
	n	66	77	70	86	72	61	32	49	54
2	m	3	-4	-2	-12	-7	1	21	19	17
	n	55	79	75	97	50	60	33	50	61

Таблица 9-17 – Значения переменных m и n для ctxIdx от 60 до 69

Переменные инициализации	ctxIdx									
	60	61	62	63	64	65	66	67	68	69
m	0	0	0	0	-9	4	0	-7	13	3
n	41	63	63	63	83	86	97	72	41	62

Таблица 9-18 – Значения переменных m и n для ctxIdx от 70 до 104

ctxIdx	Секции I и SI		Значение sabac_init_idc						ctxIdx	Секции I и SI		Значение sabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n		
70	0	11	0	45	13	15	7	34	88	-11	115	-13	108	-4	92	5	78
71	1	55	-4	78	7	51	-9	88	89	-12	63	-3	46	0	39	-6	55
72	0	69	-3	96	2	80	-20	127	90	-2	68	-1	65	0	65	4	61
73	-17	127	-27	126	-39	127	-36	127	91	-15	84	-1	57	-15	84	-14	83
74	-13	102	-28	98	-18	91	-17	91	92	-13	104	-9	93	-35	127	-37	127
75	0	82	-25	101	-17	96	-14	95	93	-3	70	-3	74	-2	73	-5	79
76	-7	74	-23	67	-26	81	-25	84	94	-8	93	-9	92	-12	104	-11	104
77	-21	107	-28	82	-35	98	-25	86	95	-10	90	-8	87	-9	91	-11	91
78	-27	127	-20	94	-24	102	-12	89	96	-30	127	-23	126	-31	127	-30	127
79	-31	127	-16	83	-23	97	-17	91	97	-1	74	5	54	3	55	0	65
80	-24	127	-22	110	-27	119	-31	127	98	-6	97	6	60	7	56	-2	79
81	-18	95	-21	91	-24	99	-14	76	99	-7	91	6	59	7	55	0	72
82	-27	127	-18	102	-21	110	-18	103	100	-20	127	6	69	8	61	-4	92
83	-21	114	-13	93	-18	102	-13	90	101	-4	56	-1	48	-3	53	-6	56
84	-30	127	-29	127	-36	127	-37	127	102	-5	82	0	68	0	68	3	68
85	-17	123	-7	92	0	80	11	80	103	-7	76	-4	69	-7	74	-8	71
86	-12	115	-5	89	-5	89	5	76	104	-22	125	-8	88	-9	88	-13	98
87	-16	122	-7	96	-7	94	2	84									

Таблица 9-19 – Значения переменных m и n для ctxIdx от 105 до 165

ctxIdx	Секции I и SI		Значение sabac_init_idc						ctxIdx	Секции I и SI		Значение sabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n		
105	-7	93	-2	85	-13	103	-4	86	136	-13	101	5	53	0	58	-5	75
106	-11	87	-6	78	-13	91	-12	88	137	-13	91	-2	61	-1	60	-8	80
107	-3	77	-1	75	-9	89	-5	82	138	-12	94	0	56	-3	61	-21	83
108	-5	71	-7	77	-14	92	-3	72	139	-10	88	0	56	-8	67	-21	64
109	-4	63	2	54	-8	76	-4	67	140	-16	84	-13	63	-25	84	-13	31
110	-4	68	5	50	-12	87	-8	72	141	-10	86	-5	60	-14	74	-25	64
111	-12	84	-3	68	-23	110	-16	89	142	-7	83	-1	62	-5	65	-29	94
112	-7	62	1	50	-24	105	-9	69	143	-13	87	4	57	5	52	9	75
113	-7	65	6	42	-10	78	-1	59	144	-19	94	-6	69	2	57	17	63
114	8	61	-4	81	-20	112	5	66	145	1	70	4	57	0	61	-8	74
115	5	56	1	63	-17	99	4	57	146	0	72	14	39	-9	69	-5	35
116	-2	66	-4	70	-78	127	-4	71	147	-5	74	4	51	-11	70	-2	27
117	1	64	0	67	-70	127	-2	71	148	18	59	13	68	18	55	13	91
118	0	61	2	57	-50	127	2	58	149	-8	102	3	64	-4	71	3	65
119	-2	78	-2	76	-46	127	-1	74	150	-15	100	1	61	0	58	-7	69
120	1	50	11	35	-4	66	-4	44	151	0	95	9	63	7	61	8	77
121	7	52	4	64	-5	78	-1	69	152	-4	75	7	50	9	41	-10	66
122	10	35	1	61	-4	71	0	62	153	2	72	16	39	18	25	3	62
123	0	44	11	35	-8	72	-7	51	154	-11	75	5	44	9	32	-3	68
124	11	38	18	25	2	59	-4	47	155	-3	71	4	52	5	43	-20	81
125	1	45	12	24	-1	55	-6	42	156	15	46	11	48	9	47	0	30
126	0	46	13	29	-7	70	-3	41	157	-13	69	-5	60	0	44	1	7
127	5	44	13	36	-6	75	-6	53	158	0	62	-1	59	0	51	-3	23
128	31	17	-10	93	-8	89	8	76	159	0	65	0	59	2	46	-21	74
129	1	51	-7	73	-34	119	-9	78	160	21	37	22	33	19	38	16	66
130	7	50	-2	73	-3	75	-11	83	161	-15	72	5	44	-4	66	-23	124
131	28	19	13	46	32	20	9	52	162	9	57	14	43	15	38	17	37
132	16	33	9	49	30	22	0	67	163	16	54	-1	78	12	42	44	-18
133	14	62	-7	100	-44	127	-5	90	164	0	62	0	60	9	34	50	-34
134	-13	108	9	53	0	54	1	67	165	12	72	9	69	0	89	-22	127
135	-15	100	2	53	-5	61	-15	72									

Таблица 9-20 – Значения переменных m и n для ctxIdx от 166 до 226

ctxIdx	Секции I и SI		Значение sabac_init_idc						ctxIdx	Секции I и SI		Значение f sabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n		
166	24	0	11	28	4	45	4	39	197	26	-17	28	3	36	-28	28	-3
167	15	9	2	40	10	28	0	42	198	30	-25	28	4	38	-28	24	10
168	8	25	3	44	10	31	7	34	199	28	-20	32	0	38	-27	27	0
169	13	18	0	49	33	-11	11	29	200	33	-23	34	-1	34	-18	34	-14
170	15	9	0	46	52	-43	8	31	201	37	-27	30	6	35	-16	52	-44
171	13	19	2	44	18	15	6	37	202	33	-23	30	6	34	-14	39	-24
172	10	37	2	51	28	0	7	42	203	40	-28	32	9	32	-8	19	17
173	12	18	0	47	35	-22	3	40	204	38	-17	31	19	37	-6	31	25
174	6	29	4	39	38	-25	8	33	205	33	-11	26	27	35	0	36	29
175	20	33	2	62	34	0	13	43	206	40	-15	26	30	30	10	24	33
176	15	30	6	46	39	-18	13	36	207	41	-6	37	20	28	18	34	15
177	4	45	0	54	32	-12	4	47	208	38	1	28	34	26	25	30	20
178	1	58	3	54	102	-94	3	55	209	41	17	17	70	29	41	22	73
179	0	62	2	58	0	0	2	58	210	30	-6	1	67	0	75	20	34
180	7	61	4	63	56	-15	6	60	211	27	3	5	59	2	72	19	31
181	12	38	6	51	33	-4	8	44	212	26	22	9	67	8	77	27	44
182	11	45	6	57	29	10	11	44	213	37	-16	16	30	14	35	19	16
183	15	39	7	53	37	-5	14	42	214	35	-4	18	32	18	31	15	36
184	11	42	6	52	51	-29	7	48	215	38	-8	18	35	17	35	15	36
185	13	44	6	55	39	-9	4	56	216	38	-3	22	29	21	30	21	28
186	16	45	11	45	52	-34	4	52	217	37	3	24	31	17	45	25	21
187	12	41	14	36	69	-58	13	37	218	38	5	23	38	20	42	30	20
188	10	49	8	53	67	-63	9	49	219	42	0	18	43	18	45	31	12
189	30	34	-1	82	44	-5	19	58	220	35	16	20	41	27	26	27	16
190	18	42	7	55	32	7	10	48	221	39	22	11	63	16	54	24	42
191	10	55	-3	78	55	-29	12	45	222	14	48	9	59	7	66	0	93
192	17	51	15	46	32	1	0	69	223	27	37	9	64	16	56	14	56
193	17	46	22	31	0	0	20	33	224	21	60	-1	94	11	73	15	57
194	0	89	-1	84	27	36	8	63	225	12	68	-2	89	10	67	26	38
195	26	-19	25	7	33	-25	35	-18	226	2	97	-9	108	-10	116	-24	127
196	22	-17	30	-7	34	-30	33	-25									

Таблица 9-21 – Значения переменных m и n для ctxIdx от 227 до 275

ctxIdx	Секции I и SI		Значение sabac_init_idc						ctxIdx	Секции I и SI		Значение sabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n		
227	-3	71	-6	76	-23	112	-24	115	252	-12	73	-6	55	-16	72	-14	75
228	-6	42	-2	44	-15	71	-22	82	253	-8	76	0	58	-7	69	-10	79
229	-5	50	0	45	-7	61	-9	62	254	-7	80	0	64	-4	69	-9	83
230	-3	54	0	52	0	53	0	53	255	-9	88	-3	74	-5	74	-12	92
231	-2	62	-3	64	-5	66	0	59	256	-17	110	-10	90	-9	86	-18	108
232	0	58	-2	59	-11	77	-14	85	257	-11	97	0	70	2	66	-4	79
233	1	63	-4	70	-9	80	-13	89	258	-20	84	-4	29	-9	34	-22	69
234	-2	72	-4	75	-9	84	-13	94	259	-11	79	5	31	1	32	-16	75
235	-1	74	-8	82	-10	87	-11	92	260	-6	73	7	42	11	31	-2	58
236	-9	91	-17	102	-34	127	-29	127	261	-4	74	1	59	5	52	1	58
237	-5	67	-9	77	-21	101	-21	100	262	-13	86	-2	58	-2	55	-13	78
238	-5	27	3	24	-3	39	-14	57	263	-13	96	-3	72	-2	67	-9	83
239	-3	39	0	42	-5	53	-12	67	264	-11	97	-3	81	0	73	-4	81
240	-2	44	0	48	-7	61	-11	71	265	-19	117	-11	97	-8	89	-13	99
241	0	46	0	55	-11	75	-10	77	266	-8	78	0	58	3	52	-13	81
242	-16	64	-6	59	-15	77	-21	85	267	-5	33	8	5	7	4	-6	38
243	-8	68	-7	71	-17	91	-16	88	268	-4	48	10	14	10	8	-13	62
244	-10	78	-12	83	-25	107	-23	104	269	-2	53	14	18	17	8	-6	58
245	-6	77	-11	87	-25	111	-15	98	270	-3	62	13	27	16	19	-2	59
246	-10	86	-30	119	-28	122	-37	127	271	-13	71	2	40	3	37	-16	73
247	-12	92	1	58	-11	76	-10	82	272	-10	79	0	58	-1	61	-10	76
248	-15	55	-3	29	-10	44	-8	48	273	-12	86	-3	70	-5	73	-13	86
249	-10	60	-1	36	-10	52	-8	61	274	-13	90	-6	79	-1	70	-9	83
250	-6	62	1	38	-10	57	-8	66	275	-14	97	-8	85	-4	78	-10	87
251	-4	65	2	43	-9	58	-7	70									

Таблица 9-22 – Значения переменных m и n для ctxIdx от 277 до 337

ctxIdx	Секции I и SI		Значение sabac_init_idc						ctxIdx	Секции I и SI		Значение sabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n		
277	-6	93	-13	106	-21	126	-22	127	308	-16	96	-1	51	-16	77	-10	67
278	-6	84	-16	106	-23	124	-25	127	309	-7	88	7	49	-2	64	1	68
279	-8	79	-10	87	-20	110	-25	120	310	-8	85	8	52	2	61	0	77
280	0	66	-21	114	-26	126	-27	127	311	-7	85	9	41	-6	67	2	64
281	-1	71	-18	110	-25	124	-19	114	312	-9	85	6	47	-3	64	0	68
282	0	62	-14	98	-17	105	-23	117	313	-13	88	2	55	2	57	-5	78
283	-2	60	-22	110	-27	121	-25	118	314	4	66	13	41	-3	65	7	55
284	-2	59	-21	106	-27	117	-26	117	315	-3	77	10	44	-3	66	5	59
285	-5	75	-18	103	-17	102	-24	113	316	-3	76	6	50	0	62	2	65
286	-3	62	-21	107	-26	117	-28	118	317	-6	76	5	53	9	51	14	54
287	-4	58	-23	108	-27	116	-31	120	318	10	58	13	49	-1	66	15	44
288	-9	66	-26	112	-33	122	-37	124	319	-1	76	4	63	-2	71	5	60
289	-1	79	-10	96	-10	95	-10	94	320	-1	83	6	64	-2	75	2	70
290	0	71	-12	95	-14	100	-15	102	321	-7	99	-2	69	-1	70	-2	76
291	3	68	-5	91	-8	95	-10	99	322	-14	95	-2	59	-9	72	-18	86
292	10	44	-9	93	-17	111	-13	106	323	2	95	6	70	14	60	12	70
293	-7	62	-22	94	-28	114	-50	127	324	0	76	10	44	16	37	5	64
294	15	36	-5	86	-6	89	-5	92	325	-5	74	9	31	0	47	-12	70
295	14	40	9	67	-2	80	17	57	326	0	70	12	43	18	35	11	55
296	16	27	-4	80	-4	82	-5	86	327	-11	75	3	53	11	37	5	56
297	12	29	-10	85	-9	85	-13	94	328	1	68	14	34	12	41	0	69
298	1	44	-1	70	-8	81	-12	91	329	0	65	10	38	10	41	2	65
299	20	36	7	60	-1	72	-2	77	330	-14	73	-3	52	2	48	-6	74
300	18	32	9	58	5	64	0	71	331	3	62	13	40	12	41	5	54
301	5	42	5	61	1	67	-1	73	332	4	62	17	32	13	41	7	54
302	1	48	12	50	9	56	4	64	333	-1	68	7	44	0	59	-6	76
303	10	62	15	50	0	69	-7	81	334	-13	75	7	38	3	50	-11	82
304	17	46	18	49	1	69	5	64	335	11	55	13	50	19	40	-2	77
305	9	64	17	54	7	69	15	57	336	5	64	10	57	3	66	-2	77
306	-12	104	10	41	-7	69	1	67	337	12	70	26	43	18	50	25	42
307	-11	97	7	46	-6	67	0	68									

Таблица 9-23 – Значения переменных m и n для ctxIdx от 338 до 398

ctxIdx	Секции I и SI		Значение sabac_init_idc						ctxIdx	Секции I и SI		Значение sabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	n	m	n	m	n		
338	15	6	14	11	19	-6	17	-13	369	32	-26	31	-4	40	-37	37	-17
339	6	19	11	14	18	-6	16	-9	370	37	-30	27	6	38	-30	32	1
340	7	16	9	11	14	0	17	-12	371	44	-32	34	8	46	-33	34	15
341	12	14	18	11	26	-12	27	-21	372	34	-18	30	10	42	-30	29	15
342	18	13	21	9	31	-16	37	-30	373	34	-15	24	22	40	-24	24	25
343	13	11	23	-2	33	-25	41	-40	374	40	-15	33	19	49	-29	34	22
344	13	15	32	-15	33	-22	42	-41	375	33	-7	22	32	38	-12	31	16
345	15	16	32	-15	37	-28	48	-47	376	35	-5	26	31	40	-10	35	18
346	12	23	34	-21	39	-30	39	-32	377	33	0	21	41	38	-3	31	28
347	13	23	39	-23	42	-30	46	-40	378	38	2	26	44	46	-5	33	41
348	15	20	42	-33	47	-42	52	-51	379	33	13	23	47	31	20	36	28
349	14	26	41	-31	45	-36	46	-41	380	23	35	16	65	29	30	27	47
350	14	44	46	-28	49	-34	52	-39	381	13	58	14	71	25	44	21	62
351	17	40	38	-12	41	-17	43	-19	382	29	-3	8	60	12	48	18	31
352	17	47	21	29	32	9	32	11	383	26	0	6	63	11	49	19	26
353	24	17	45	-24	69	-71	61	-55	384	22	30	17	65	26	45	36	24
354	21	21	53	-45	63	-63	56	-46	385	31	-7	21	24	22	22	24	23
355	25	22	48	-26	66	-64	62	-50	386	35	-15	23	20	23	22	27	16
356	31	27	65	-43	77	-74	81	-67	387	34	-3	26	23	27	21	24	30
357	22	29	43	-19	54	-39	45	-20	388	34	3	27	32	33	20	31	29
358	19	35	39	-10	52	-35	35	-2	389	36	-1	28	23	26	28	22	41
359	14	50	30	9	41	-10	28	15	390	34	5	28	24	30	24	22	42
360	10	57	18	26	36	0	34	1	391	32	11	23	40	27	34	16	60
361	7	63	20	27	40	-1	39	1	392	35	5	24	32	18	42	15	52
362	-2	77	0	57	30	14	30	17	393	34	12	28	29	25	39	14	60
363	-4	82	-14	82	28	26	20	38	394	39	11	23	42	18	50	3	78
364	-3	94	-5	75	23	37	18	45	395	30	29	19	57	12	70	-16	123
365	9	69	-19	97	12	55	15	54	396	34	26	22	53	21	54	21	53
366	-12	109	-35	125	11	65	0	79	397	29	39	22	61	14	71	22	56
367	36	-35	27	0	37	-33	36	-16	398	19	66	11	86	11	83	25	61
368	36	-34	28	0	39	-36	37	-14									

Таблица 9-24 – Значения переменных m и n для ctxIdx от 402 до 459

ctxIdx	Секции I		Значение sabac_init_idc						ctxIdx	Секции I		Значение sabac_init_idc					
			0		1		2					0		1		2	
	m	n	m	n	m	n	m	n		m	N	m	n	m	n	m	n
402	-17	120	-4	79	-5	85	-3	78	431	-2	55	-12	56	-9	57	-12	59
403	-20	112	-7	71	-6	81	-8	74	432	0	61	-6	60	-6	63	-8	63
404	-18	114	-5	69	-10	77	-9	72	433	1	64	-5	62	-4	65	-9	67
405	-11	85	-9	70	-7	81	-10	72	434	0	68	-8	66	-4	67	-6	68
406	-15	92	-8	66	-17	80	-18	75	435	-9	92	-8	76	-7	82	-10	79
407	-14	89	-10	68	-18	73	-12	71	436	-14	106	-5	85	-3	81	-3	78
408	-26	71	-19	73	-4	74	-11	63	437	-13	97	-6	81	-3	76	-8	74
409	-15	81	-12	69	-10	83	-5	70	438	-15	90	-10	77	-7	72	-9	72
410	-14	80	-16	70	-9	71	-17	75	439	-12	90	-7	81	-6	78	-10	72
411	0	68	-15	67	-9	67	-14	72	440	-18	88	-17	80	-12	72	-18	75
412	-14	70	-20	62	-1	61	-16	67	441	-10	73	-18	73	-14	68	-12	71
413	-24	56	-19	70	-8	66	-8	53	442	-9	79	-4	74	-3	70	-11	63
414	-23	68	-16	66	-14	66	-14	59	443	-14	86	-10	83	-6	76	-5	70
415	-24	50	-22	65	0	59	-9	52	444	-10	73	-9	71	-5	66	-17	75
416	-11	74	-20	63	2	59	-11	68	445	-10	70	-9	67	-5	62	-14	72
417	23	-13	9	-2	17	-10	9	-2	446	-10	69	-1	61	0	57	-16	67
418	26	-13	26	-9	32	-13	30	-10	447	-5	66	-8	66	-4	61	-8	53
419	40	-15	33	-9	42	-9	31	-4	448	-9	64	-14	66	-9	60	-14	59
420	49	-14	39	-7	49	-5	33	-1	449	-5	58	0	59	1	54	-9	52
421	44	3	41	-2	53	0	33	7	450	2	59	2	59	2	58	-11	68
422	45	6	45	3	64	3	31	12	451	21	-10	21	-13	17	-10	9	-2
423	44	34	49	9	68	10	37	23	452	24	-11	33	-14	32	-13	30	-10
424	33	54	45	27	66	27	31	38	453	28	-8	39	-7	42	-9	31	-4
425	19	82	36	59	47	57	20	64	454	28	-1	46	-2	49	-5	33	-1
426	-3	75	-6	66	-5	71	-9	71	455	29	3	51	2	53	0	33	7
427	-1	23	-7	35	0	24	-7	37	456	29	9	60	6	64	3	31	12
428	1	34	-7	42	-1	36	-8	44	457	35	20	61	17	68	10	37	23
429	1	43	-8	45	-2	42	-11	49	458	29	36	55	34	66	27	31	38
430	0	54	-5	48	-2	52	-10	56	459	14	67	42	62	47	57	20	64

9.3.1.2 Процесс инициализации устройства арифметического декодирования

Процесс активируют перед декодированием первого макроблока секции или после декодирования любого pcm_alignment_zero_bit и всех данных pcm_sample_luma и pcm_sample_chroma макроблока типа I_PCM.

Выходы этого процесса – регистры инициализированного устройства декодирования codIRange и codIOffset – оба с точностью 16-битового регистра.

Статус устройства арифметического декодирования представлен переменными codIRange и codIOffset. В процедуре инициализации процесса арифметического декодирования codIRange устанавливают равным 0x01FE, а codIOffset устанавливают равным возвратному значению от read_bits(9), которое интерпретируют как 9 битов бинарного представления целого без знака со старшим значащим битом, записанным первым.

Поток битов не должен содержать данных, в результате которых значение `codIOffset` становится равным `0x01FE` или `0x01FF`.

ПРИМЕЧАНИЕ. – В настоящей Рекомендации | Международном стандарте при описании устройства арифметического декодирования используют точность 16-битового регистра. Однако минимальная точность регистра для переменных `codIRange` и `codIOffset` составляет 9 битов.

9.3.2 Процесс бинаризации

Вход в этот процесс – запрос элемента синтаксиса.

Выход этого процесса – бинаризация элементов синтаксиса, `maxBinIdxCtx`, `ctxIdxOffset` и `bypassFlag`.

Таблица 9-24 определяет тип процесса бинаризации, `maxBinIdxCtx` и `ctxIdxOffset`, связанный с каждым элементом синтаксиса.

Спецификации унарного (U) процесса бинаризации, усеченного унарного (TU) процесса бинаризации, связанного унарного процесса бинаризации *k*-ого порядка с кодом `Exp-Golomb (UEGk)` и процесса бинаризации с фиксированной длиной (FL), даны в пп. 9.3.2.1–9.3.2.4, соответственно. Прочие бинаризации определены в пп. 9.3.2.5–9.3.2.7.

За исключением секций I, бинаризация элемента синтаксиса `mb_type`, как это определено в п. 9.3.2.5, состоит из строк бинов, заданных связью со строками битов префиксов и суффиксов. `UEGk` бинаризация, как это определено в п. 9.3.2.3, которую используют для элементов синтаксиса `mvd_IX` ($X = 0, 1$) и `coeff_abs_level_minus1`, и бинаризация `coded_block_pattern` также состоит из связи со строками битов префиксов и суффиксов. Для этих процессов бинаризации строки битов префиксов и суффиксов индексируют отдельно, используя переменную `binIdx`, как это определено далее в п. 9.3.3. Два множества строк битов префиксов и суффиксов называют частью бинаризации префикса и частью бинаризации суффикса, соответственно.

Связанный с каждой бинаризацией или частью бинаризации элемент синтаксиса является определенным значением переменной сдвига (`ctxIdxOffset`) индекса контекста и определенным значением переменной `maxBinIdxCtx`, заданной в таблице 9-25. Если два значения для каждой из этих переменных определены для одного элемента синтаксиса в таблице 9-25, значение в верхнем ряду относится к части префикса, а значение в нижнем ряду – к части суффикса бинаризации соответствующего элемента синтаксиса.

Использование процесса `DecodeBypass` и переменной `bypassFlag` происходит следующим образом.

- Если значение `ctxIdxOffset` не присвоено соответствующей бинаризации или части бинаризации в таблице 9-25, что обозначено как "na", все бины в строке битов соответствующей бинаризации или части бинаризации префикса/суффикса декодируются с помощью процесса `DecodeBypass`, как это определено в п. 9.3.3.2.3. В этом случае `bypassFlag` устанавливают равным 1, и `bypassFlag` используют для указания на то, что для анализа значения бина из потока битов применяется процесс `DecodeBypass`.
- Иначе для каждого возможного значения `binIdx`, вплоть до определенного значения `maxBinIdxCtx`, заданного в таблице 9-25, особое значение переменной `ctxIdx` определено далее в п. 9.3.3. В этом случае `bypassFlag` устанавливают равным 0.

Возможные значения индекса контекста `ctxIdx` находятся в диапазоне от 0 до 459 включительно. Значение, присвоенное `ctxIdxOffset`, определяет нижнее значение диапазона `ctxIdx`, присвоенное соответствующей бинаризации или части бинаризации элемента синтаксиса.

Значение `ctxIdx = ctxIdxOffset = 276` присвоено элементу синтаксиса `end_of_slice_flag` и бину `mb_type`, который определяет тип макроблока I_PCM, как описано далее в п. 9.3.3.1. Для анализа значения соответствующего бина из потока битов, используется процесс арифметического декодирования для принятия решений перед окончанием (`DecodeTerminate`), как это определено в п. 9.3.3.2.4.

ПРИМЕЧАНИЕ. – Бины `mb_type` в I секциях и бины суффикса `mb_type` в SI секциях, которые соответствуют тем же значениям `binIdx`, имеют общее значение `ctxIdx`. Последний бин префикса `mb_type` и первый бин суффикса `mb_type` в секциях P, SP и B могут иметь общее значение `ctxIdx`.

Таблица 9-25 – Элементы синтаксиса и объединенные типы бинаризации, maxBinIdxCtx и ctxIdxOffset

Элемент синтаксиса	Тип бинаризации	maxBinIdxCtx	ctxIdxOffset
mb_type (только SI секции)	префикс и суффикс, как это определено в п. 9.3.2.5	префикс: 0 суффикс: 6	префикс: 0 суффикс: 3
mb_type (только I секции)	как это определено в п. 9.3.2.5	6	3
mb_skip_flag (только P, SP секции)	FL, cMax = 1	0	11
mb_type (только P, SP секции)	префикс и суффикс, как это определено в п. 9.3.2.5	префикс: 2 суффикс: 5	префикс: 14 суффикс: 17
sub_mb_type (только P, SP секции)	как это определено в п. 9.3.2.5	2	21
mb_skip_flag (только B секции)	FL, cMax = 1	0	24
mb_type (только B секции)	префикс и суффикс, как это определено в п. 9.3.2.5	префикс: 3 суффикс: 5	префикс: 27 суффикс: 32
sub_mb_type (только B секции)	как это определено в п. 9.3.2.5	3	36
mvd_10[][][0], mvd_11[][][0]	префикс и суффикс, как заданные UEG3 с signedValFlag = 1, uCoff = 9	префикс: 4 суффикс: na	префикс: 40 суффикс: na (использование DecodeBypass)
mvd_10[][][1], mvd_11[][][1]		префикс: 4 суффикс: na	префикс: 47 суффикс: na (использование DecodeBypass)
ref_idx_10, ref_idx_11	U	2	54
mb_qp_delta	как это определено в п. 9.3.2.7	2	60
intra_chroma_pred_mode	TU, cMax = 3	1	64
prev_intra4x4_pred_mode_flag, prev_intra8x8_pred_mode_flag	FL, cMax = 1	0	68
rem_intra4x4_pred_mode, rem_intra8x8_pred_mode	FL, cMax = 7	0	69
mb_field_decoding_flag	FL, cMax = 1	0	70
coded_block_pattern	префикс и суффикс, как это определено в п. 9.3.2.6	префикс: 3 суффикс: 1	префикс: 73 суффикс: 77
coded_block_flag	FL, cMax = 1	0	85
significant_coeff_flag (блоки, кодированные кадрами, с ctxBlockCat < 5)	FL, cMax = 1	0	105
last_significant_coeff_flag (блоки, кодированные кадрами, с ctxBlockCat < 5)	FL, cMax = 1	0	166
coeff_abs_level_minus1 (блоки с ctxBlockCat < 5)	префикс и суффикс, как заданные UEG0 с signedValFlag = 0, uCoff = 14	префикс: 1 суффикс: na	префикс: 227 суффикс: na, (использование DecodeBypass)
coeff_sign_flag	FL, cMax = 1	0	na, (использование DecodeBypass)
end_of_slice_flag	FL, cMax = 1	0	276
significant_coeff_flag (блоки, кодированные полями, с ctxBlockCat < 5)	FL, cMax = 1	0	277

Элемент синтаксиса	Тип бинаризации	maxBinIdxCtx	ctxIdxOffset
Last_significant_coeff_flag (блоки, кодированные полями, с ctxBlockCat < 5)	FL, cMax = 1	0	338
transform_size_8x8_flag	FL, cMax=1	0	399
significant_coeff_flag (блоки, кодированные кадрами, с ctxBlockCat == 5)	FL, cMax=1	0	402
last_significant_coeff_flag (блоки, кодированные кадрами, с ctxBlockCat == 5)	FL, cMax=1	0	417
coeff_abs_level_minus1 (блоки с ctxBlockCat == 5)	префикс и суффикс, как заданные UEG0 с signedValFlag=0, uCoff=14	префикс: 1 суффикс: na	префикс: 426 суффикс: na, (использование DecodeBypass)
significant_coeff_flag (блоки, кодированные полями, с ctxBlockCat == 5)	FL, cMax=1	0	436
last_significant_coeff_flag (блоки, кодированные полями, с ctxBlockCat == 5)	FL, cMax=1	0	451

9.3.2.1 Унарный (U) процесс бинаризации

Вход в этот процесс – запрос U бинаризации элемента синтаксиса.

Выход этого процесса – U бинаризация элемента синтаксиса.

Строка бинов элемента синтаксиса, принимающего значение (целое без знака) synElVal, – это строка битов длиной synElVal + 1, индексированная BinIdx. Бины binIdx, меньшие synElVal, равны 1. Бин со значением binIdx, равным synElVal, равен 0.

Таблица 9-26 иллюстрирует строку бинов унарной бинаризации элемента синтаксиса.

Таблица 9-26 – Строка бинов унарной бинаризации (для информации)

Значение элемента синтаксиса	Строка бинов					
0 (I_NxN)	0					
1	1	0				
2	1	1	0			
3	1	1	1	0		
4	1	1	1	1	0	
5	1	1	1	1	1	0
...						
binIdx	0	1	2	3	4	5

9.3.2.2 Усеченный унарный (TU) процесс бинаризации

Вход в этот процесс – запрос TU бинаризации элемента синтаксиса и cMax.

Выход этого процесса – TU бинаризация элемента синтаксиса.

Для значения элемента синтаксиса (целого без знака), меньшего cMax, активируют U процесс бинаризации, как это определено в п. 9.3.2.1. Для значения элемента синтаксиса равного cMax строка бинов – это строка битов длиной cMax со всеми бинами, равными 1.

ПРИМЕЧАНИЕ. – TU бинаризацию всегда активируют со значением cMax, равным наибольшему возможному значению элемента синтаксиса, подлежащего декодированию.

9.3.2.3 Соединенный процесс бинаризации: унарный и k-ого порядка с экспоненциальным кодом Exp-Golomb (UEGk)

Вход в этот процесс – запрос UEGk бинаризации для элемента синтаксиса, signedValFlag и uCoff.

Выход этого процесса – UEGk бинаризация элемента синтаксиса.

Строка бинов UEGk – соединение префикса строки битов и суффикса строки битов. Префикс бинаризации описывают, активируя TU процесс бинаризации для части префикса $\text{Min}(u\text{Coff}, \text{Abs}(\text{synElVal}))$ значения элемента синтаксиса synElVal, как это определено в п. 9.3.2.2, с $c\text{Max} = u\text{Coff}$, где $u\text{Coff} > 0$.

Строку бинов UEGk находят следующим образом.

- Если одно из следующих условий – истина, то строка бинов элемента синтаксиса со значением synElVal состоит только из префикса строки битов:
 - signedValFlag равно 0, а префикс строки битов не равен строке битов длиной uCoff со всеми битами, равными 1;
 - signedValFlag равно 1, а префикс строки битов равен строке битов, которая состоит из единственного бита со значением, равным 0.
- Иначе строку бинов части суффикса UEGk со значением элемента синтаксиса synElVal описывают процессом, эквивалентным следующему псевдокоду:

```
if( Abs( synElVal ) >= uCoff ) {
    sufS = Abs( synElVal ) - uCoff
    stopLoop = 0
    do {
        if( sufS >= ( 1 << k ) ) {
            put( 1 )
            sufS = sufS - ( 1 << k )
            k++
        } else {
            put( 0 )
            while( k-- )
                put( ( sufS >> k ) & 0x01 )
            stopLoop = 1
        }
    } while( !stopLoop )
}
if( signedValFlag && synElVal != 0 )
    if( synElVal > 0 )
        put( 0 )
    else
        put( 1 ).
```

ПРИМЕЧАНИЕ. – Спецификация для кода Exp-Golomb (EGk) k-ого порядка использует 1 и 0 в обратном значении для унарной части кода Exp-Golomb 0-го порядка, как это определено в п. 9.1.

9.3.2.4 Процесс бинаризации фиксированной длины (FL)

Вход в этот процесс – запрос FL бинаризации элемента синтаксиса и cMax.

Выход этого процесса – FL бинаризация элемента синтаксиса.

FL бинаризацию создают, используя fixedLength-bit, – строку бинов с целыми без знака значения элемента синтаксиса, где $\text{fixedLength} = \text{Ceil}(\text{Log}_2(c\text{Max} + 1))$. Индексацию бинов при FL бинаризации производят таким образом, что binIdx = 0 относится к младшему значащему биту с возрастающим значением binIdx в направлении к старшему значащему биту.

9.3.2.5 Процесс бинаризации типов макроблока и субмакроблока

Вход в этот процесс – запрос бинаризации элементов синтаксиса mb_type или sub_mb_type.

Выход этого процесса – бинаризация элемента синтаксиса.

Схема бинаризации для декодирования типа макроблока в I секции описана в таблице 9-27.

Для типов макроблоков в SI секции, бинаризация состоит из строки бинов, определенных как соединение префикса и суффикса строки битов следующим образом.

Префикс строки битов состоит из единственного бита, который описывают как $b_0 = ((mb_type == SI) ? 0 : 1)$. При значении элемента синтаксиса для каждого b_0 , равного 0, строка бинов состоит только из префикса строки битов. При значении элемента синтаксиса для каждого b_0 , равного 1, бинаризация задана соединением префикса b_0 и

суффикса строки битов, как это определено в таблице 9-27 для типа макроблока в I секции, индексированной вычитанием 1 из значения mb_type в SI секции.

Таблица 9-27 – Бинаризация типов макроблоков в I секции

Значение (имя) mb_type	Строка битов						
	0	1	2	3	4	5	6
0 (I_4x4)	0						
1 (I_16x16_0_0_0)	1	0	0	0	0	0	
2 (I_16x16_1_0_0)	1	0	0	0	0	1	
3 (I_16x16_2_0_0)	1	0	0	0	1	0	
4 (I_16x16_3_0_0)	1	0	0	0	1	1	
5 (I_16x16_0_1_0)	1	0	0	1	0	0	0
6 (I_16x16_1_1_0)	1	0	0	1	0	0	1
7 (I_16x16_2_1_0)	1	0	0	1	0	1	0
8 (I_16x16_3_1_0)	1	0	0	1	0	1	1
9 (I_16x16_0_2_0)	1	0	0	1	1	0	0
10 (I_16x16_1_2_0)	1	0	0	1	1	0	1
11 (I_16x16_2_2_0)	1	0	0	1	1	1	0
12 (I_16x16_3_2_0)	1	0	0	1	1	1	1
13 (I_16x16_0_0_1)	1	0	1	0	0	0	
14 (I_16x16_1_0_1)	1	0	1	0	0	1	
15 (I_16x16_2_0_1)	1	0	1	0	1	0	
16 (I_16x16_3_0_1)	1	0	1	0	1	1	
17 (I_16x16_0_1_1)	1	0	1	1	0	0	0
18 (I_16x16_1_1_1)	1	0	1	1	0	0	1
19 (I_16x16_2_1_1)	1	0	1	1	0	1	0
20 (I_16x16_3_1_1)	1	0	1	1	0	1	1
21 (I_16x16_0_2_1)	1	0	1	1	1	0	0
22 (I_16x16_1_2_1)	1	0	1	1	1	0	1
23 (I_16x16_2_2_1)	1	0	1	1	1	1	0
24 (I_16x16_3_2_1)	1	0	1	1	1	1	1
25 (I_PCM)	1	1					
binIdx	0	1	2	3	4	5	6

Схемы бинаризации типов P макроблоков в P и SP секциях и B макроблоков в B секции показаны в таблице 9-28.

Строка битов типов I макроблоков в P и SP секциях, соответствующих значениям mb_type от 5 до 30, состоит из соединения префикса, который представлен единственным битом со значением, равным 1, как это определено в таблице 9-28, и суффикса, как это определено в таблице 9-26, индексированной вычитанием 5 из значения mb_type.

Значение mb_type, равное 4 (P_8x8ref0), не разрешено.

Для типов I макроблоков в В секциях (значения mb_type от 23 до 48) бинаризация состоит из строки бинов, определенных как соединение префикса строки битов, как это указано в таблице 9-28, и суффикса строки битов, как это указано в таблице 9-27, индексированной вычитанием 23 из значения mb_type.

Таблица 9-28 – Бинаризация типов макроблоков в P, SP и В секциях

Тип секции	Значение (имя) mb_type	Строка бинов					
P, SP секции	0 (P_L0_16x16)	0	0	0			
	1 (P_L0_L0_16x8)	0	1	1			
	2 (P_L0_L0_8x16)	0	1	0			
	3 (P_8x8)	0	0	1			
	4 (P_8x8ref0)	na					
	от 5 до 30 (Intra, только префикс)	1					
В секция	0 (B_Direct_16x16)	0					
	1 (B_L0_16x16)	1	0	0			
	2 (B_L1_16x16)	1	0	1			
	3 (B_Bi_16x16)	1	1	0	0	0	0
	4 (B_L0_L0_16x8)	1	1	0	0	0	1
	5 (B_L0_L0_8x16)	1	1	0	0	1	0
	6 (B_L1_L1_16x8)	1	1	0	0	1	1
	7 (B_L1_L1_8x16)	1	1	0	1	0	0
	8 (B_L0_L1_16x8)	1	1	0	1	0	1
	9 (B_L0_L1_8x16)	1	1	0	1	1	0
	10 (B_L1_L0_16x8)	1	1	0	1	1	1
	11 (B_L1_L0_8x16)	1	1	1	1	1	0
	12 (B_L0_Bi_16x8)	1	1	1	0	0	0
	13 (B_L0_Bi_8x16)	1	1	1	0	0	0
	14 (B_L1_Bi_16x8)	1	1	1	0	0	1
	15 (B_L1_Bi_8x16)	1	1	1	0	0	1
	16 (B_Bi_L0_16x8)	1	1	1	0	1	0
	17 (B_Bi_L0_8x16)	1	1	1	0	1	0
	18 (B_Bi_L1_16x8)	1	1	1	0	1	1
	19 (B_Bi_L1_8x16)	1	1	1	0	1	1
	20 (B_Bi_Bi_16x8)	1	1	1	1	0	0
	21 (B_Bi_Bi_8x16)	1	1	1	1	0	0
22 (B_8x8)	1	1	1	1	1	1	
от 23 до 48 (Intra, только префикс)	1	1	1	1	0	1	
binIdx		0	1	2	3	4	5

Для секций P, SP и В спецификация бинаризации sub_mb_type задана в таблице 9-29.

Таблица 9-29 – Бинаризация типов субмакроблоков в P, SP и B секциях

Тип секции	Значение (имя) sub_mb_type	Строка бинов					
P, SP секции	0 (P_L0_8x8)	1					
	1 (P_L0_8x4)	0	0				
	2 (P_L0_4x8)	0	1	1			
	3 (P_L0_4x4)	0	1	0			
B секция	0 (B_Direct_8x8)	0					
	1 (B_L0_8x8)	1	0	0			
	2 (B_L1_8x8)	1	0	1			
	3 (B_Bi_8x8)	1	1	0	0	0	
	4 (B_L0_8x4)	1	1	0	0	1	
	5 (B_L0_4x8)	1	1	0	1	0	
	6 (B_L1_8x4)	1	1	0	1	1	
	7 (B_L1_4x8)	1	1	1	0	0	0
	8 (B_Bi_8x4)	1	1	1	0	0	1
	9 (B_Bi_4x8)	1	1	1	0	1	0
	10 (B_L0_4x4)	1	1	1	0	1	1
	11 (B_L1_4x4)	1	1	1	1	0	
12 (B_Bi_4x4)	1	1	1	1	1		
binIdx		0	1	2	3	4	5

9.3.2.6 Процесс бинаризации для кодированного образца блока

Вход в этот процесс – запрос бинаризации элемента синтаксиса coded_block_pattern.

Выход этого процесса – бинаризация элемента синтаксиса.

Бинаризация coded_block_pattern состоит из части префикса и части суффикса (если представлена). Часть префикса бинаризации задана длиной FL бинаризации CodedBlockPatternLuma с cMax = 15. Если chroma_format_idc не равно 0 (монохромное изображение), часть суффикса представлена и состоит из TU бинаризации CodedBlockPatternChroma с cMax = 2. Взаимодействие между значением элемента синтаксиса coded_block_pattern и значениями CodedBlockPatternLuma и CodedBlockPatternChroma заданы, как это определено в п. 7.4.5.

9.3.2.7 Процесс бинаризации для mb_qp_delta

Вход в этот процесс – запрос бинаризации элемента синтаксиса mb_qp_delta.

Выход этого процесса – бинаризация элемента синтаксиса.

Строку бинов mb_qp_delta находят U бинаризацией отображенного значения элемента синтаксиса mb_qp_delta, где правило присвоения между значением mb_qp_delta со знаком и его отображенным значением задано в таблице 9-3.

9.3.3 Процесс декодирования потока

Вход в этот процесс – бинаризация запрошенного элемента синтаксиса, maxBinIdxCtx, bypassFlag и ctxIdxOffset, как это определено в п. 9.3.2.

Выход этого процесса – значение данного элемента синтаксиса.

Процесс определяет, как анализируют каждый бит строки битов для каждого элемента синтаксиса.

После анализа каждого бита результирующую строку битов сравнивают со всеми строками бинов процесса бинаризации элемента синтаксиса и применяют следующие условия:

- если строка битов равна одной строке бинов, то соответствующее значение элемента синтаксиса – это выход;
- иначе (строка битов не равна одной строке бинов) анализируют следующий бит.

При анализе каждого бина переменная binIdx возрастает на 1, начиная с binIdx, которое устанавливают на 0 для первого бина.

Если бинаризация соответствующего элемента синтаксиса состоит из бинаризации частей префикса и суффикса, переменную binIdx устанавливают равной 0 для первого бина каждой части строки битов (части префикса или части суффикса). В этом случае после анализа префикса строки битов, анализ суффикса строки битов (связанный с процессами бинаризации из пп. 9.3.2.3 и 9.3.2.5) активируют в зависимости от результирующего префикса строки битов, как это указано в пп. 9.3.2.3 и 9.3.2.5. Отметим, что для бинаризации элемента синтаксиса coded_block_pattern суффикс строки битов представляют независимо от префикса строки битов длиной 4, как это определено в п. 9.3.2.6.

В зависимости от переменной bypassFlag применяют следующее.

- Если bypassFlag равно 1, выполняется процесс обхода декодирования, как это определено в п. 9.3.3.2.3, для анализа значений бинов из потока битов.
- Иначе (bypassFlag равно 0) анализ каждого бина производят по порядку с помощью следующих двух шагов.
 1. Находят заданные значения binIdx, maxBinIdxCtx и ctxIdxOffset, ctxIdx, как это определено в п. 9.3.3.1.
 2. зная ctxIdx, декодируют значение бина из потока битов, как это определено в п. 9.3.3.2.

9.3.3.1 Процесс вывода ctxIdx

Входы в этот процесс – binIdx, maxBinIdxCtx и ctxIdxOffset.

Выход этого процесса – ctxIdx.

В таблице 9-30 оказано присвоение приращений ctxIdx (ctxIdxInc) параметрам binIdx для всех ctxIdxOffset, за исключением тех значений, которые относятся к элементам синтаксиса coded_block_flag, significant_coeff_flag, last_significant_coeff_flag и coeff_abs_level_minus1.

Значение ctxIdx, которое следует использовать со специальным значением binIdx, описано первым определением ctxIdxOffset, связанным с заданной строкой бинов или ее частью. Значение ctxIdx определяют следующим образом.

- Если ctxIdxOffset перечислено в таблице 9-30, то значение ctxIdx для binIdx представляет сумму ctxIdxOffset и ctxIdxInc, которую находят в таблице 9-30. Если в таблице 9-30 перечислено более одного значения binIdx, то процесс присвоения ctxIdxInc для binIdx описан далее в пунктах, указанных в скобках в соответствующей графе таблицы.
- Иначе (ctxIdxOffset не перечислено в таблице 9-30) ctxIdx описывают суммой следующих членов: ctxIdxOffset и ctxIdxBlockCatOffset(ctxBlockCat), как это определено в таблице 9-31, и ctxIdxInc(ctxBlockCat). В п. 9.3.3.1.3 указано, какое значение ctxBlockCat используют. В п. 9.3.3.1.1.9 определено присвоение ctxIdxInc(ctxBlockCat) для coded_block_flag, а в п. 9.3.3.1.3 определено присвоение ctxIdxInc(ctxBlockCat) для significant_coeff_flag, last_significant_coeff_flag и coeff_abs_level_minus1.

Все бины с binIdx больше, чем maxBinIdxCtx, анализируют, используя значение ctxIdx, присвоенное BinIdx, равному maxBinIdxCtx.

Все входы в таблице 9-30, отмеченные как "na", соответствуют значениям binIdx, которые не встречаются для соответствующего значения ctxIdxOffset.

Значение ctxIdx = 276 присваивают параметру binIdx mb_type, указывающему на режим I_PCM. Перед завершением для анализа значений соответствующих бинов из потока битов применяется процесс арифметического декодирования решений, как это определено в п. 9.3.3.2.4.

Таблица 9-30 – Присвоение ctxIdxInc параметру binIdx для всех значений ctxIdxOffset, за исключением тех, которые относятся к элементам синтаксиса coded_block_flag, significant_coeff_flag, last_significant_coeff_flag и coeff_abs_level_minus1

ctxIdxOffset	binIdx						
	0	1	2	3	4	5	>= 6
0	0,1,2 (п. 9.3.3.1.1.3)	na	na	na	na	na	na
3	0,1,2 (п. 9.3.3.1.1.3)	ctxIdx=276	3	4	5,6 (п. 9.3.3.1.2)	6,7 (п. 9.3.3.1.2)	7
11	0,1,2 (п. 9.3.3.1.1.1)	na	na	na	na	na	na
14	0	1	2,3 (п. 9.3.3.1.2)	na	na	na	na
17	0	ctxIdx=276	1	2	2,3 (п. 9.3.3.1.2)	3	3
21	0	1	2	na	na	na	na
24	0,1,2 (п. 9.3.3.1.1.1)	na	na	na	na	na	na
27	0,1,2 (п. 9.3.3.1.1.3)	3	4,5 (п. 9.3.3.1.2)	5	5	5	5
32	0	ctxIdx=276	1	2	2,3 (п. 9.3.3.1.2)	3	3
36	0	1	2,3 (п. 9.3.3.1.2)	3	3	3	na
40	0,1,2 (п. 9.3.3.1.1.7)	3	4	5	6	6	6
47	0,1,2 (п. 9.3.3.1.1.7)	3	4	5	6	6	6
54	0,1,2,3 (п. 9.3.3.1.1.6)	4	5	5	5	5	5
60	0,1 (п. 9.3.3.1.1.5)	2	3	3	3	3	3
64	0,1,2 (п. 9.3.3.1.1.8)	3	3	na	na	na	na
68	0	na	na	na	na	na	na
69	0	0	0	na	na	na	na
70	0,1,2 (п. 9.3.3.1.1.2)	na	na	na	na	na	na
73	0,1,2,3 (п. 9.3.3.1.1.4)	0,1,2,3 (п. 9.3.3.1.1.4)	0,1,2,3 (п. 9.3.3.1.1.4)	0,1,2,3 (п. 9.3.3.1.1.4)	na	na	na
77	0,1,2,3 (п. 9.3.3.1.1.4)	4,5,6,7 (п. 9.3.3.1.1.4)	na	na	na	na	na
276	0	na	na	na	na	na	na
399	0,1,2 (п. 9.3.3.1.1.10)	na	na	na	na	na	na

В таблице 9-31 показаны значения `ctxIdxBlockCatOffset` в зависимости от `ctxBlockCat` для элементов синтаксиса `coded_block_flag`, `significant_coeff_flag`, `last_significant_coeff_flag`, и `coeff_abs_level_minus1`. Спецификация `ctxBlockCat` дана в таблице 9-33.

Таблица 9-31 – Присвоение `ctxIdxBlockCatOffset` параметру `ctxBlockCat` для элементов синтаксиса `coded_block_flag`, `significant_coeff_flag`, `last_significant_coeff_flag` и `coeff_abs_level_minus1`

Элемент синтаксиса	ctxBlockCat (как это определено в таблице 9-33)					
	0	1	2	3	4	5
<code>coded_block_flag</code>	0	4	8	12	16	na
<code>significant_coeff_flag</code>	0	15	29	44	47	0
<code>last_significant_coeff_flag</code>	0	15	29	44	47	0
<code>coeff_abs_level_minus1</code>	0	10	20	30	39	0

9.3.3.1.1 Процесс присвоения `ctxIdxInc` с использованием ближайших элементов синтаксиса

Пункт 9.3.3.1.1.1 определяет процесс вывода `ctxIdxInc` для элемента синтаксиса `mb_skip_flag`.

Пункт 9.3.3.1.1.2 определяет процесс вывода `ctxIdxInc` для элемента синтаксиса `mb_field_decoding_flag`.

Пункт 9.3.3.1.1.3 определяет процесс вывода `ctxIdxInc` для элемента синтаксиса `mb_type`.

Пункт 9.3.3.1.1.4 определяет процесс вывода `ctxIdxInc` для элемента синтаксиса `coded_block_pattern`.

Пункт 9.3.3.1.1.5 определяет процесс вывода `ctxIdxInc` для элемента синтаксиса `mb_qp_delta`.

Пункт 9.3.3.1.1.6 определяет процесс вывода `ctxIdxInc` для элементов синтаксиса `ref_idx_10` и `ref_idx_11`.

Пункт 9.3.3.1.1.7 определяет процесс вывода `ctxIdxInc` для элементов синтаксиса `mvd_10` и `mvd_11`.

Пункт 9.3.3.1.1.8 определяет процесс вывода `ctxIdxInc` для элемента синтаксиса `intra_chroma_pred_mode`.

Пункт 9.3.3.1.1.9 определяет процесс вывода `ctxIdxInc` для элемента синтаксиса `coded_block_flag`.

Пункт 9.3.3.1.1.10 определяет процесс вывода `ctxIdxInc` для элемента синтаксиса `transform_size_8x8_flag`.

9.3.3.1.1.1 Процесс вывода `ctxIdxInc` для элемента синтаксиса `mb_skip_flag`

Выход этого процесса – `ctxIdxInc`.

Если `MbaffFrameFlag` равно 1, а `mb_field_decoding_flag` еще не декодировано для текущей пары макроблоков с адресом верхнего макроблока $address\ 2 * (CurrMbAddr / 2)$, то для элемента синтаксиса `mb_field_decoding_flag` применяется установленное правило, как это определено в п. 7.4.4.

Активируют процесс нахождения смежных макроблоков, определенных в п. 6.4.8.1, и присваивают выходу значения `mbAddrA` и `mbAddrB`.

Предположим, что переменную `condTermFlagN` (с N, принимающим значения A или B) получили следующим образом.

- Если `mbAddrN` недоступно или `mb_skip_flag` макроблока `mbAddrN` равно 1, `condTermFlagN` устанавливают равным 0.
- Иначе (`mbAddrN` доступно, а `mb_skip_flag` макроблока `mbAddrN` равно 0) `condTermFlagN` устанавливают равным 1.

Переменную `ctxIdxInc` находят как:

$$ctxIdxInc = condTermFlagA + condTermFlagB. \quad (9-1)$$

9.3.3.1.1.2 Процесс вывода `ctxIdxInc` элемента синтаксиса `mb_field_decoding_flag`

Выход этого процесса – `ctxIdxInc`.

Активируют процесс нахождения адресов смежных макроблоков и их доступности в кадрах MBAFF, как это определено в п. 6.4.7, и присваивают выходу значения `mbAddrA` и `mbAddrB`.

Если оба макроблока, mbAddrN и mbAddrN + 1, имеют mb_type, равное P_Skip или B_Skip, то для элемента синтаксиса mb_field_decoding_flag применяется установленное правило, как это определено в п. 7.4.4.

Предположим, что переменную condTermFlagN (с N, принимающим значения A или B) получили следующим образом.

- Если любое из следующих условий – истина, то condTermFlagN устанавливают равным 0:
 - mbAddrN недоступно;
 - макроблок mbAddrN – это макроблок кадра.
- Иначе condTermFlagN устанавливают равным 1.

Переменную ctxIdxInc находят как:

$$\text{ctxIdxInc} = \text{condTermFlagA} + \text{condTermFlagB}. \quad (9-2)$$

9.3.3.1.1.3 Процесс вывода ctxIdxInc элемента синтаксиса mb_type

Вход в этот процесс – ctxIdxOffset.

Выход этого процесса – ctxIdxInc.

Активируют процесс нахождения смежных макроблоков, определенных в п. 6.4.8.1, и присваивают выходу значения mbAddrA и mbAddrB.

Предположим, что переменную condTermFlagN (с N, принимающим значения A или B) получили следующим образом.

- Если любое из следующих условий – истина, то condTermFlagN устанавливают равным 0:
 - mbAddrN недоступно;
 - ctxIdxOffset равно 0, а mb_type макроблока mbAddrN равно SI;
 - ctxIdxOffset равно 3, а mb_type макроблока mbAddrN равно I_NxN;
 - ctxIdxOffset равно 27, а mb_type макроблока mbAddrN равно P_Skip, B_Skip или B_Direct_16x16.
- Иначе condTermFlagN устанавливают равным 1.

Переменную ctxIdxInc находят как:

$$\text{ctxIdxInc} = \text{condTermFlagA} + \text{condTermFlagB}. \quad (9-3)$$

9.3.3.1.1.4 Процесс вывода ctxIdxInc элемента синтаксиса coded_block_pattern

Входы в этот процесс – ctxIdxOffset и binIdx.

Выход этого процесса – ctxIdxInc.

В зависимости от значения переменной ctxIdxOffset применяют следующее.

- Если ctxIdxOffset равно 73 делают следующее.
 - Активируют процесс нахождения смежных блоков яркости 8x8, определенных в п. 6.4.8.2, с luma8x8BlkIdx = binIdx в качестве входа и присваивают выходу значения mbAddrA, mbAddrB, luma8x8BlkIdxA и luma8x8BlkIdxB.
 - Предположим, что переменную condTermFlagN (с N, принимающим значения A или B) получили следующим образом.
 - Если любое из следующих условий – истина, то condTermFlagN устанавливают равным 0:
 - mbAddrN недоступно;
 - mb_type макроблока mbAddrN равно I_PCM;
 - макроблок mbAddrN не является текущим макроблоком CurrMbAddr, а макроблок mbAddrN не содержит mb_type, равного P_Skip или B_Skip, и ((CodedBlockPatternLuma >> luma8x8BlkIdxN) & 1) не равно 0 для значения CodedBlockPatternLuma макроблока mbAddrN;
 - макроблок mbAddrN является текущим макроблоком CurrMbAddr, а ранее декодированное значение бина b_k из coded_block_pattern с k = luma8x8BlkIdxN не равно 0.
 - Иначе condTermFlagN устанавливают равным 1.

- Переменную `ctxIdxInc` находят как:

$$\text{ctxIdxInc} = \text{condTermFlagA} + 2 * \text{condTermFlagB}. \quad (9-4)$$

- Иначе (`ctxIdxOffset` равно 77) применяют следующее.
 - Активируют процесс нахождения смежных макроблоков, определенных в п. 6.4.8.1, и присваиваются выходу значения `mbAddrA` и `mbAddrB`.
 - Предположим, что переменную `condTermFlagN` (с `N`, принимающим значения `A` или `B`) получили следующим образом.
 - Если `mbAddrN` доступно, а `mb_type` макроблока `mbAddrN` равно `I_PCM`, то `condTermFlagN` устанавливают равным 1.
 - Иначе, если любое из следующих условий – истина, то `condTermFlagN` устанавливают равным 0:
 - `mbAddrN` недоступно или макроблок `mbAddrN` содержит `mb_type`, равное `P_Skip` или `B_Skip`;
 - `binIdx` равно 0 и `CodedBlockPatternChroma` макроблока `mbAddrN` равно 0;
 - `binIdx` равно 1 и `CodedBlockPatternChroma` макроблока `mbAddrN` не равно 2.
 - Иначе `condTermFlagN` устанавливают равным 1.
 - Переменную `ctxIdxInc` находят как:

$$\text{ctxIdxInc} = \text{condTermFlagA} + 2 * \text{condTermFlagB} + ((\text{binIdx} == 1) ? 4 : 0). \quad (9-5)$$

ПРИМЕЧАНИЕ. – Если для макроблока используют режим предсказания `Intra_16x16`, то значения `CodedBlockPatternLuma` и `CodedBlockPatternChroma` макроблока находят из `mb_type`, как это определено в таблице 7-11.

9.3.3.1.1.5 Процесс вывода `ctxIdxInc` элемента синтаксиса `mb_qp_delta`

Выход этого процесса – `ctxIdxInc`.

Предположим, что `prevMbAddr` – адрес макроблока из той группы макроблоков, которая предшествует текущему макроблоку в порядке декодирования. Если текущий макроблок – первый макроблок секции, то `prevMbAddr` помечают как недоступный.

Предположим, что переменную `ctxIdxInc` получили следующим образом.

- Если любое из перечисленных условий – истина, то `ctxIdxInc` устанавливают равным 0:
 - `prevMbAddr` недоступно или макроблок `prevMbAddr` содержит `mb_type`, равное `P_Skip` или `B_Skip`;
 - `mb_type` макроблока `prevMbAddr` равно `I_PCM`;
 - макроблок `prevMbAddr` не кодирован в режиме предсказания `Intra_16x16`, а оба значения `CodedBlockPatternLuma` и `CodedBlockPatternChroma` макроблока `prevMbAddr` равны 0;
 - `mb_qp_delta` макроблока `prevMbAddr` равно 0.
- Иначе `ctxIdxInc` устанавливают равным 1.

9.3.3.1.1.6 Процесс вывода `ctxIdxInc` элемента синтаксиса `ref_idx_10` и `ref_idx_11`

Вход в этот процесс – `mbPartIdx`.

Выход этого процесса – `ctxIdxInc`.

Интерпретация `ref_idx_1X` и `Pred_LX` в настоящем пункте описывается следующим образом.

- Если этот процесс активируют для вывода `ref_idx_10`, `ref_idx_1X` интерпретируется как `ref_idx_10`, а `Pred_LX` интерпретируется как `Pred_L0`.
- Иначе (этот процесс активируется для вывода `ref_idx_11`) `ref_idx_1X` интерпретируется как `ref_idx_11`, а `Pred_LX` интерпретируется как `Pred_L1`.

Положим, что `currSubMbType` равно `sub_mb_type[mbPartIdx]`.

Процесс нахождения смежных разделенных частей, определенных в п. 6.4.8.5, активируют с `mbPartIdx`, `currSubMbType` и `subMbPartIdx = 0` в качестве входа, а выходу присваивают значения `mbAddrA\mbPartIdxA` и `mbAddrB\mbPartIdxB`.

Со значением `ref_idx_IX[mbPartIdxN]` (при `N`, равным `A` или `B`), которая определяет элемент синтаксиса макроблока `mbAddrN`, предположим, что переменную `refIdxZeroFlagN` получили следующим образом.

- Если `MbaffFrameFlag` равно 1, текущий макроблок – макроблок кадра, а макроблок `mbAddrN` – макроблок поля:

$$\text{refIdxZeroFlagN} = ((\text{ref_idx_IX}[\text{mbPartIdxN}] > 1) ? 0 : 1). \quad (9-6)$$

- Иначе:

$$\text{refIdxZeroFlagN} = ((\text{ref_idx_IX}[\text{mbPartIdxN}] > 0) ? 0 : 1). \quad (9-7)$$

Предположим, что переменная `predModeEqualFlagN` определена следующим образом.

- Если макроблок `mbAddrN` имеет `mb_type`, равное `P_8x8` или `B_8x8`, то применяют следующее:
 - если `SubMbPredMode(sub_mb_type[mbPartIdxN])` не равно `Pred_LX` и не равно `BiPred`, то `predModeEqualFlagN` устанавливают равным 0, где `sub_mb_type` определяет элемент синтаксиса макроблока `mbAddrN`;
 - иначе `predModeEqualFlagN` устанавливают равным 1.
- Иначе применяют следующее:
 - если `MbPartPredMode(mb_type, mbPartIdxN)` не равно `Pred_LX` и не равно `BiPred`, то `predModeEqualFlagN` устанавливают равным 0, где `mb_type` определяет элемент синтаксиса макроблока `mbAddrN`;
 - иначе `predModeEqualFlagN` устанавливают равным 1.

Предположим, что переменную `condTermFlagN` (с `N` принимающим значения `A` или `B`) получили следующим образом.

- Если любое из перечисленных условий – истина, то `condTermFlagN` устанавливают равным 0:
 - `mbAddrN` недоступно;
 - макроблок `mbAddrN` имеет `mb_type`, равное `P_Skip` или `B_Skip`;
 - макроблок `mbAddrN` кодирован в режиме `Intra` предсказания;
 - `predModeEqualFlagN` равно 0;
 - `refIdxZeroFlagN` равно 1.
- Иначе `condTermFlagN` устанавливают равным 1.

Переменную `ctxIdxInc` находят как:

$$\text{ctxIdxInc} = \text{condTermFlagA} + 2 * \text{condTermFlagB}. \quad (9-8)$$

9.3.3.1.1.7 Процесс вывода `ctxIdxInc` элементов синтаксиса `mvd_10` и `mvd_11`

Входы в этот процесс – `mbPartIdx`, `subMbPartIdx` и `ctxIdxOffset`.

Выход этого процесса – `ctxIdxInc`.

Интерпретация `ref_idx_IX` и `Pred_LX` в настоящем пункте описывается следующим образом.

- Если этот процесс активируют для вывода `mvd_10`, `mvd_IX` интерпретируется как `mvd_10`, а `Pred_LX` интерпретируется как `Pred_L0`.
- Иначе (этот процесс активируют для вывода `mvd_11`) `mvd_IX` интерпретируется как `mvd_11`, а `Pred_LX` интерпретируется как `Pred_L1`.

Положим, что `currSubMbType` установлено равным `sub_mb_type[mbPartIdx]`.

Процесс нахождения смежных разделенных частей, определенных в п. 6.4.8.5, активируют с `mbPartIdx`, `currSubMbType` и `subMbPartIdx` в качестве входа, а выходу присваивают значения `mbAddrA\mbPartIdxA\subMbPartIdxA` и `mbAddrB\mbPartIdxB\subMbPartIdxB`.

Предположим, что переменную `compIdx` получили следующим образом:

- если `ctxIdxOffset` равно 40, то `compIdx` устанавливают равным 0;
- иначе (`ctxIdxOffset` равно 47) `compIdx` устанавливают равным 1.

Предположим, что переменная `predModeEqualFlagN` определена следующим образом.

- Если макроблок `mbAddrN` имеет `mb_type`, равное `P_8x8` или `B_8x8`, то применяют следующее:
 - если `SubMbPredMode(sub_mb_type[mbPartIdxN])` не равно `Pred_LX` и не равно `BiPred`, то `predModeEqualFlagN` устанавливают равным 0, где `sub_mb_type` определяет элемент синтаксиса макроблока `mbAddrN`;
 - иначе `predModeEqualFlagN` устанавливают равным 1.
- Иначе применяют следующее:
 - если `MbPartPredMode(mb_type, mbPartIdxN)` не равно `Pred_LX` и не равно `BiPred`, то `predModeEqualFlagN` устанавливают равным 0, где `mb_type` определяет элемент синтаксиса макроблока `mbAddrN`;
 - иначе `predModeEqualFlagN` устанавливают равным 1.

Предположим, что переменную `absMvdCompN` (с `N`, принимающим значения `A` или `B`) получили следующим образом.

- Если любое из следующих условий – истина, то `absMvdCompN` устанавливают равным 0:
 - `mbAddrN` недоступно;
 - макроблок `mbAddrN` имеет `mb_type`, равное `P_Skip` или `B_Skip`;
 - макроблок `mbAddrN` кодирован в режиме `Intra` предсказания;
 - `predModeEqualFlagN` равно 0.
- Иначе применяют следующее.
 - Если `compIdx` равно 1, `MbaffFrameFlag` равно 1, то текущий макроблок – макроблок кадра, а макроблок `mbAddrN` – макроблок поля:

$$\text{absMvdCompN} = \text{Abs}(\text{mvd_IX}[\text{mbPartIdxN}][\text{subMbPartIdxN}][\text{compIdx}]) * 2. \quad (9-9)$$

- Иначе, если `compIdx` равно 1, `MbaffFrameFlag` равно 1, то текущий макроблок – макроблок поля, а макроблок `mbAddrN` – макроблок кадра:

$$\text{absMvdCompN} = \text{Abs}(\text{mvd_IX}[\text{mbPartIdxN}][\text{subMbPartIdxN}][\text{compIdx}]) / 2. \quad (9-10)$$

- Иначе:

$$\text{absMvdCompN} = \text{Abs}(\text{mvd_IX}[\text{mbPartIdxN}][\text{subMbPartIdxN}][\text{compIdx}]). \quad (9-11)$$

Переменную `ctxIdxInc` находят следующим образом:

- если $(\text{absMvdCompA} + \text{absMvdCompB})$ меньше чем 3, `ctxIdxInc` устанавливают равным 0;
- иначе, если $(\text{absMvdCompA} + \text{absMvdCompB})$ больше чем 32, `ctxIdxInc` устанавливают равным 2;
- иначе $(\text{absMvdCompA} + \text{absMvdCompB})$ находится в диапазоне от 3 до 32 включительно) `ctxIdxInc` устанавливают равным 1.

9.3.3.1.1.8 Процесс вывода `ctxIdxInc` элемента синтаксиса `intra_chroma_pred_mode`

Выход этого процесса – `ctxIdxInc`.

Активируют процесс нахождения смежных макроблоков, определенных в п. 6.4.8.1, и присваивают выходу значения `mbAddrA` и `mbAddrB`.

Предположим, что переменную `condTermFlagN` (с заменой `N` на `A` или `B`) получили следующим образом.

- Если любое из перечисленных условий – истина, то `condTermFlagN` устанавливают равным 0:
 - `mbAddrN` недоступно;
 - макроблок `mbAddrN` кодирован в режим `Inter` предсказания;
 - `mb_type` макроблока `mbAddrN` равно `I_PCM`;
 - `intra_chroma_pred_mode` макроблока `mbAddrN` равно 0.
- Иначе `condTermFlagN` устанавливают равным 1.

Переменную `ctxIdxInc` находят как:

$$\text{ctxIdxInc} = \text{condTermFlagA} + \text{condTermFlagB}. \quad (9-12)$$

9.3.3.1.1.9 Процесс вывода `ctxIdxInc` элемента синтаксиса `coded_block_flag`

Вход в этот процесс – `ctxBlockCat`, а дополнительный вход описывают следующим образом:

- если `ctxBlockCat` равно 0, дополнительный вход отсутствует;
- иначе, если `ctxBlockCat` равно 1 или 2, то `luma4x4BlkIdx`;
- иначе, если `ctxBlockCat` равно 3, то индекс компонента цветности `iCbCr`;
- иначе (`ctxBlockCat` равно 4), то `chroma4x4BlkIdx`, а индекс компонента цветности `iCbCr`.

Выход этого процесса – `ctxIdxInc(ctxBlockCat)`.

Предположим, что переменную `transBlockN` (с `N`, принимающим значения `A` или `B`) получили следующим образом.

- Если `ctxBlockCat` равно 0, применяют следующее.
 - Активируют процесс нахождения смежных макроблоков, определенных в п. 6.4.8.1, и присваивают выходу значение `mbAddrN` (с `N`, принимающим значения `A` или `B`).
 - Переменную `transBlockN` находят следующим образом:
 - если `mbAddrN` доступно, а макроблок `mbAddrN` кодирован в режиме предсказания `Intra_16x16`, то блоку яркости DC макроблока `mbAddrN` присваивают значение `transBlockN`;
 - иначе `transBlockN` помечают как недоступный.
- Иначе, если `ctxBlockCat` равно 1 или 2, применяют следующее.
 - Активируют процесс нахождения смежных блоков яркости 4x4, определенных в п. 6.4.7.3, с `luma4x4BlkIdx` в качестве входа, и присваиваются выходу значение `mbAddrN`, `luma4x4BlkIdxN` (с `N`, принимающим значения `A` или `B`).
 - Переменную `transBlockN` находят следующим образом.
 - Если `mbAddrN` доступен, макроблок `mbAddrN` не содержит `mb_type`, равного `P_Skip`, `B_Skip` или `I_PCM`, $((\text{CodedBlockPatternLuma} \gg (\text{luma4x4BlkIdxN} \gg 2)) \& 1)$ не равно 0 для макроблока `mbAddrN`, а `transform_size_8x8_flag` равно 0 для макроблока `mbAddrN`, то блоку яркости 4x4 с индексом `luma4x4BlkIdxN` макроблока `mbAddrN` присваивают `transBlockN`.
 - Иначе, если `mbAddrN` доступен, макроблок `mbAddrN` не содержит `mb_type`, равного `P_Skip` или `B_Skip`, $((\text{CodedBlockPatternLuma} \gg (\text{luma4x4BlkIdxN} \gg 2)) \& 1)$ не равно 0 для макроблока `mbAddrN`, а `transform_size_8x8_flag` равно 1 для макроблока `mbAddrN`, то блоку яркости 8x8 с индексом $(\text{luma4x4BlkIdxN} \gg 2)$ макроблока `mbAddrN` присваивают `transBlockN`.
 - Иначе `transBlockN` помечают как недоступный.
- Иначе, если `ctxBlockCat` равно 3, применяют следующее.
 - Активируют процесс нахождения смежных макроблоков, определенных в п. 6.4.8.1, и присваивают выходу значение `mbAddrN` (с `N`, принимающим значения `A` или `B`).
 - Переменную `transBlockN` находят следующим образом:

- если mbAddrN доступно, макроблок mbAddrN не содержит mb_type, равного P_Skip, B_Skip или I_PCM и CodedBlockPatternChroma не равно 0 для макроблока mbAddrN, то компоненту цветности iCbCr DC блока цветности макроблока mbAddrN присваивают значение transBlockN;
- иначе transBlockN помечают как недоступный.
- Иначе (ctxBlockCat равно 4) применяют следующее.
 - Активируют процесс нахождения смежных блоков цветности 4x4, определенных в п. 6.4.8.4, с chroma4x4BlkIdx в качестве входа и присваивают выходу mbAddrN значение chroma4x4BlkIdxN (с N, принимающим значения A или B).
 - Переменную transBlockN находят следующим образом:
 - если mbAddrN доступно, макроблок mbAddrN не содержит mb_type, равного P_Skip, B_Skip или I_PCM и CodedBlockPatternChroma равно 2 макроблока mbAddrN, то компоненту цветности iCbCr блока цветности 4x4 с chroma4x4BlkIdxN макроблока mbAddrN присваивают значение transBlockN;
 - иначе transBlockN помечают как недоступный.

Предположим, что переменную condTermFlagN (с N, принимающим значения A или B) получили следующим образом.

- Если любое из перечисленных условий – истина, то condTermFlagN устанавливают равным 0:
 - mbAddrN недоступно, а текущий макроблок кодирован в режиме Inter предсказания;
 - mbAddrN доступно, а transBlockN недоступно и mb_type макроблока mbAddrN не равно I_PCM;
 - текущий макроблок кодирован в режиме Intra предсказания, constrained_intra_pred_flag равно 1, макроблок mbAddrN доступен и кодирован в режиме Inter предсказания, а также используют данные секции разделения на части (nal_unit_type находится в диапазоне от 2 до 4 включительно).
- Иначе, если следующие условия – истина, то condTermFlagN устанавливают равным 1:
 - mbAddrN недоступно, а текущий макроблок кодирован в режиме Intra предсказания;
 - mb_type макроблока mbAddrN равно I_PCM.
- Иначе condTermFlagN устанавливают равным значению coded_block_flag преобразования блока transBlockN, который был декодирован для макроблока mbAddrN.

Переменную ctxIdxInc(ctxBlockCat) находят как:

$$\text{ctxIdxInc}(\text{ctxBlockCat}) = \text{condTermFlagA} + 2 * \text{condTermFlagB}. \quad (9-13)$$

9.3.3.1.10 Процесс вывода ctxIdxInc элемента синтаксиса transform_size_8x8_flag

Выход этого процесса – ctxIdxInc.

Активируют процесс нахождения смежных макроблоков, определенных в п. 6.4.8.1, и присваивают выходу значения mbAddrA и mbAddrB.

Предположим, что переменную condTermFlagN (с N, принимающим значения A или B) получили следующим образом.

- Если любое из перечисленных условий – истина, то condTermFlagN устанавливают равным 0:
 - mbAddrN недоступно;
 - transform_size_8x8_flag макроблока mbAddrN равно 0
- Иначе condTermFlagN устанавливают равным 1.

Переменную ctxIdxInc находят как:

$$\text{ctxIdxInc} = \text{condTermFlagA} + \text{condTermFlagB}. \quad (9-14)$$

9.3.3.1.2 Процесс присвоения значения ctxIdxInc, которое используют перед декодированием значений бинов

Входы в этот процесс – ctxIdxOffset и binIdx.

Выход этого процесса – ctxIdxInc.

Таблица 9-32 содержит спецификацию `ctxIdxInc` для заданных значений `ctxIdxOffset` и `binIdx`.

Для каждого `ctxIdxOffset binIdx` значение `ctxIdxInc` находят, используя некоторые из ранее декодированных значений бинов ($b_0, b_1, b_2, \dots, b_k$), где значение индекса k меньше, чем значение `binIdx`.

Таблица 9-32 – Спецификация `ctxIdxInc` для особых значений `ctxIdxOffset` и `binIdx`

Значение (имя) <code>ctxIdxOffset</code>	<code>binIdx</code>	<code>ctxIdxInc</code>
3	4	$(b_3 \neq 0) ? 5 : 6$
	5	$(b_3 \neq 0) ? 6 : 7$
14	2	$(b_1 \neq 1) ? 2 : 3$
17	4	$(b_3 \neq 0) ? 2 : 3$
27	2	$(b_1 \neq 0) ? 4 : 5$
32	4	$(b_3 \neq 0) ? 2 : 3$
36	2	$(b_1 \neq 0) ? 2 : 3$

9.3.3.1.3 Процесс присвоения `ctxIdxInc` элементам синтаксиса `significant_coeff_flag`, `last_significant_coeff_flag` и `coeff_abs_level_minus1`

Входы в этот процесс – `ctxIdxOffset` и `binIdx`.

Выход этого процесса – `ctxIdxInc`.

Процесс присвоения `ctxIdxInc` элементам синтаксиса `significant_coeff_flag`, `last_significant_coeff_flag` и `coeff_abs_level_minus1`, а также `coded_block_flag` зависит от категорий различных блоков, отмеченных переменной `ctxBlockCat`. Спецификация этих категорий блоков показана в таблице 9-33.

Таблица 9-33 – Спецификация `ctxBlockCat` для различных блоков

Описание блока	<code>maxNumCoeff</code>	<code>ctxBlockCat</code>
блок DC уровней коэффициентов преобразования яркости (т. е. перечень <code>LumaLevel[i]</code> , как это описано в п. 7.4.5.3)	16	0
блок AC уровней коэффициентов преобразования яркости (т. е. перечень <code>LumaLevel[i]</code> , как это описано в п. 7.4.5.3)	15	1
блок 16 уровней коэффициентов преобразования яркости (т. е. перечень <code>LumaLevel[i]</code> , как это описано в п. 7.4.5.3)	$4 * \text{NumC8x8}$	2
блок DC уровней коэффициентов преобразования цветности	4	3
блок AC уровней коэффициентов преобразования цветности	15	4
Блок 64 уровней коэффициентов преобразования яркости (т. е. перечень <code>LumaLevel[i]</code> , как это описано в п. 7.4.5.3)	64	5

Положим, что переменная `levelListIdx` должна быть установлена равной индексу перечня уровней коэффициентов преобразования, как это описано в п. 7.4.5.3.

Для элементов синтаксиса `significant_coeff_flag` и `last_significant_coeff_flag` в блоках с `ctxBlockCat < 5` и `ctxBlockCat != 3` переменная `ctxIdxInc` определяется следующим образом:

$$\text{ctxIdxInc} = \text{levelListIdx}, \quad (9-15)$$

где `levelListIdx` находится в диапазоне от 0 до `maxNumCoeff – 2` включительно.

Для элементов синтаксиса `significant_coeff_flag` и `last_significant_coeff_flag` в блоках с `ctxBlockCat == 3` переменная `ctxIdxInc` определяется следующим образом:

$$\text{ctxIdxInc} = \text{Min}(\text{levelListIdx} / \text{NumC8x8}, 2), \quad (9-16)$$

где `levelListIdx` находится в диапазоне от 0 до $4 * \text{NumC8x8} - 2$ включительно.

Для элементов синтаксиса `significant_coeff_flag` и `last_significant_coeff_flag` в блоках яркости 8x8 с `ctxBlockCat == 5` в таблице 9-34 содержится спецификация `ctxIdxInc` для заданных значений `levelListIdx`, где `levelListIdx` находится в диапазоне от 0 до 62 включительно.

Таблица 9-34 – Отображение позиции сканирования в `ctxIdxInc` для `ctxBlockCat == 5`

<code>levelListIdx</code>	<code>ctxIdxInc</code> для <code>significant_coeff_flag</code> (макроблоки, кодированные кадром)	<code>ctxIdxInc</code> для <code>significant_coeff_flag</code> (макроблоки, кодированные полем)	<code>ctxIdxInc</code> для <code>last_significant_coeff_flag</code>	<code>levelListIdx</code>	<code>ctxIdxInc</code> для <code>significant_coeff_flag</code> (макроблоки, кодированные кадром)	<code>ctxIdxInc</code> для <code>significant_coeff_flag</code> (макроблоки, кодированные полем)	<code>ctxIdxInc</code> для <code>last_significant_coeff_flag</code>
0	0	0	0	32	7	9	3
1	1	1	1	33	6	9	3
2	2	1	1	34	11	10	3
3	3	2	1	35	12	10	3
4	4	2	1	36	13	8	3
5	5	3	1	37	11	11	3
6	5	3	1	38	6	12	3
7	4	4	1	39	7	11	3
8	4	5	1	40	8	9	4
9	3	6	1	41	9	9	4
10	3	7	1	42	14	10	4
11	4	7	1	43	10	10	4
12	4	7	1	44	9	8	4
13	4	8	1	45	8	13	4
14	5	4	1	46	6	13	4
15	5	5	1	47	11	9	4
16	4	6	2	48	12	9	5
17	4	9	2	49	13	10	5
18	4	10	2	50	11	10	5
19	4	10	2	51	6	8	5
20	3	8	2	52	9	13	6
21	3	11	2	53	14	13	6
22	6	12	2	54	10	9	6

levelListIdx	ctxIdxInc для significant_coeff_flag (макроблоки, кодированные кадром)	ctxIdxInc для significant_coeff_flag (макроблоки, кодированные полем)	ctxIdxInc для last_significant_coeff_flag	levelListIdx	ctxIdxInc для significant_coeff_flag (макроблоки, кодированные кадром)	ctxIdxInc для significant_coeff_flag (макроблоки, кодированные полем)	ctxIdxInc для last_significant_coeff_flag
23	7	11	2	55	9	9	6
24	7	9	2	56	11	10	7
25	7	9	2	57	12	10	7
26	8	10	2	58	13	14	7
27	9	10	2	59	11	14	7
28	10	8	2	60	14	14	8
29	9	11	2	61	10	14	8
30	8	12	2	62	12	14	8
31	7	11	2				

Положим, что numDec108odAbsLevelEq1 описывает общее число декодированных уровней коэффициентов преобразования с абсолютным значением, равным 1, и положим, что numDecodAbsLevelGt1 описывает общее число декодированных уровней коэффициентов преобразования с абсолютным значением, большим, чем 1. Оба набора чисел относятся к тем же коэффициентам преобразования блока, где происходит текущий процесс декодирования. Далее декодирование coeff_abs_level_minus1 и ctxIdxInc для coeff_abs_level_minus1 описывают в зависимости от binIdx следующим образом.

- Если binIdx равно 0, ctxIdxInc находят как:

$$\text{ctxIdxInc} = ((\text{numDecodAbsLevelGt1} \neq 0) ? 0 : \text{Min}(4, 1 + \text{numDecodAbsLevelEq1})). \quad (9-17)$$

- Иначе (binIdx больше, чем 0) ctxIdxInc находят как:

$$\text{ctxIdxInc} = 5 + \text{Min}(4 - (\text{ctxBlockCat} == 3), \text{numDecodAbsLevelGt1}). \quad (9-18)$$

9.3.3.2 Процесс арифметического декодирования

Входы в этот процесс – bypassFlag, ctxIdx, как указано в п. 9.3.3.1, и переменные состояний codIRange и codIOffset устройства арифметического декодирования.

Выход этого процесса – значение бина.

Рисунок 9-2 иллюстрирует весь процесс арифметического декодирования для одного бина. Для декодирования значения бина индекс контекста ctxIdx вводят в процесс арифметического декодирования DecodeBin(ctxIdx), который описывают следующим образом.

- Если bypassFlag равно 1, то активируют DecodeBypass(), как это определено в п. 9.3.3.2.3.
- Иначе, если bypassFlag равно 0, а ctxIdx равно 276, то активируют DecodeTerminate(), как это определено в п. 9.3.3.2.4.
- Иначе (bypassFlag равно 0, а ctxIdx не равно 276) применяется значение DecodeDecision(), как это определено в п. 9.3.3.2.1.

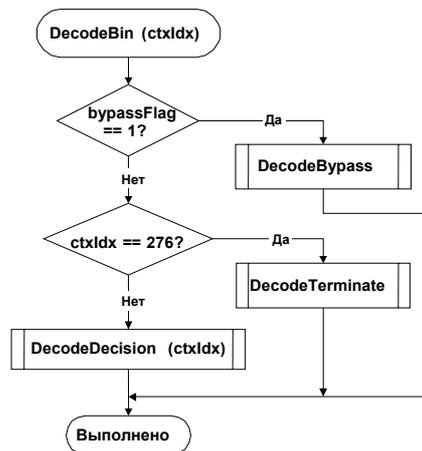


Рисунок 9-2 – Диаграмма последовательности процесса арифметического декодирования для одиночного бина (для информации)

ПРИМЕЧАНИЕ. – Арифметическое кодирование основано на принципе рекурсивного подразделения интервала. Заданная оценка вероятностей $p(0)$ и $p(1) = 1 - p(0)$ бинарного решения $(0, 1)$, первоначально заданная в подинтервале с диапазоном codIRange , должна быть разделена на два подинтервала с диапазонами $p(0) * \text{codIRange}$ и $\text{codIRange} - p(0) * \text{codIRange}$ соответственно. В зависимости от принятого решения должен быть выбран соответствующий подинтервал в качестве нового кодового интервала, а положение бинарной кодовой строки в этом интервале должно представлять последовательность рассмотренных бинарных решений. Полезно различать наиболее вероятный символ (MPS) и наименее вероятный символ (LPS) для того, чтобы это бинарное решение могло быть идентифицировано скорее как MPS или LPS, чем как 0 или 1. Придерживаясь такой терминологии, каждый контекст описывают вероятностью p_{LPS} для LPS и значением MPS (valMPS), которое может быть 0 или 1.

В настоящей Рекомендации | Международном стандарте основная часть арифметического устройства обладает тремя отличительными свойствами.

- Оценка вероятности производится конечным автоматом с табличным процессом перехода от 64 различных представлений состояний вероятности $\{p_{LPS}(pStateIdx) | 0 \leq pStateIdx < 64\}$ к вероятности LPS – p_{LPS} . Число состояний организуют таким образом, что вероятности состояний с индексом $pStateIdx = 0$ соответствует вероятность значения LPS в 0,5, с уменьшением вероятности LPS по мере возрастания индексов состояния.
- Перед вычислением нового диапазона интервала диапазон codIRange , представляющий состояние кодирующего устройства, квантуют до небольшого множества $\{Q_1, \dots, Q_4\}$ заранее квантованных значений. Хранимая таблица, которая содержит все 64×4 заранее вычисленные результирующие значения $Q_i * p_{LPS}(pStateIdx)$, разрешает свободный порядок аппроксимации результата в виде $\text{codIRange} * p_{LPS}(pStateIdx)$.
- Для элементов синтаксиса или их частей, для которых предполагают приблизительно равномерное распределение вероятности, используют упрощенный процесс обхода кодирования и декодирования.

9.3.3.2.1 Процесс арифметического декодирования для бинарного решения

Входы в этот процесс – ctxIdx , codIRange и codIOffset .

Выходы этого процесса – декодированные значения binVal и обновленные переменные codIRange и codIOffset .

На рисунке 9-3 показана диаграмма последовательности действий для декодирования одного решения (DecodeDecision).

1. Значение переменной codIRangeLPS находят следующим образом.
 - Для заданных текущих значений codIRange переменную qCodIRangeIdx находят как:

$$\text{qCodIRangeIdx} = (\text{codIRange} \gg 6) \& 0x03. \quad (9-19)$$

- Для заданных значений qCodIRangeIdx и $pStateIdx$, которые связаны с ctxIdx , переменной RangeTabLPS , как это определено в таблице 9-35, присваивают значение codIRangeLPS :

$$\text{codIRangeLPS} = \text{rangeTabLPS}[pStateIdx][\text{qCodIRangeIdx}]. \quad (9-20)$$

2. Переменную `codIRange` устанавливают равной `codIRange – codIRangeLPS` и применяют следующее.
 - Если `codIOffset` больше или равно `codIRange`, то переменную `binVal` устанавливают равной $1 - \text{valMPS}$, `codIOffset` уменьшают на `codIRange`, а `codIRange` устанавливают равной `codIRangeLPS`.
 - Иначе переменную `binVal` устанавливают равной `valMPS`.

При заданных значениях `binVal` состояние перехода выполняют, как это определено в п. 9.3.3.2.1.1. В зависимости от текущего значения `codIRange` выполняют изменение нормировки, как это определено в п. 9.3.3.2.2.

9.3.3.2.1.1 Процесс перехода состояния

Входы в этот процесс – текущее значение `pStateIdx`, декодированные значения `binVal` и `valMPS` контекстной переменной, связанной с `ctxIdx`.

Выходы этого процесса – обновленные значения `pStateIdx` и `valMPS` контекстной переменной, связанной с `ctxIdx`.

В зависимости от декодированных значений `binVal` обновление двух переменных `pStateIdx` и `valMPS`, связанных с `ctxIdx`, находят следующим образом:

```

if( binVal == valMPS )
    pStateIdx = transIdxMPS( pStateIdx )
else {
    if( pStateIdx == 0 )
        valMPS = 1 - valMPS
    pStateIdx = transIdxLPS( pStateIdx )
}.
  
```

(9-21)

Таблица 9-36 определяет правила перехода `transIdxMPS()` и `transIdxLPS()` после декодирования значений `valMPS` и $1 - \text{valMPS}$ соответственно.

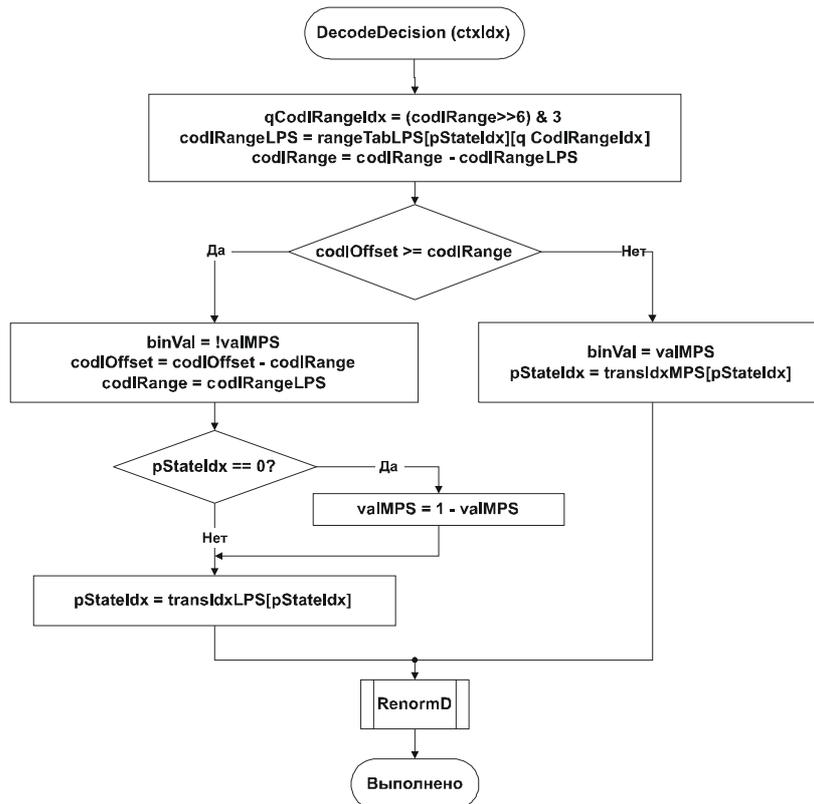


Рисунок 9-3 – Диаграмма последовательности для декодирования решения

Таблица 9-35 – Спецификация RangeTabLPS в зависимости от pStateIdx и qCodIRangeIdx

pStateIdx	qCodIRangeIdx				pStateIdx	qCodIRangeIdx			
	0	1	2	3		0	1	2	3
0	128	176	208	240	32	27	33	39	45
1	128	167	197	227	33	26	31	37	43
2	128	158	187	216	34	24	30	35	41
3	123	150	178	205	35	23	28	33	39
4	116	142	169	195	36	22	27	32	37
5	111	135	160	185	37	21	26	30	35
6	105	128	152	175	38	20	24	29	33
7	100	122	144	166	39	19	23	27	31
8	95	116	137	158	40	18	22	26	30
9	90	110	130	150	41	17	21	25	28
10	85	104	123	142	42	16	20	23	27
11	81	99	117	135	43	15	19	22	25
12	77	94	111	128	44	14	18	21	24
13	73	89	105	122	45	14	17	20	23
14	69	85	100	116	46	13	16	19	22
15	66	80	95	110	47	12	15	18	21
16	62	76	90	104	48	12	14	17	20
17	59	72	86	99	49	11	14	16	19
18	56	69	81	94	50	11	13	15	18
19	53	65	77	89	51	10	12	15	17
20	51	62	73	85	52	10	12	14	16
21	48	59	69	80	53	9	11	13	15
22	46	56	66	76	54	9	11	12	14
23	43	53	63	72	55	8	10	12	14
24	41	50	59	69	56	8	9	11	13
25	39	48	56	65	57	7	9	11	12
26	37	45	54	62	58	7	9	10	12
27	35	43	51	59	59	7	8	10	11
28	33	41	48	56	60	6	8	9	11
29	32	39	46	53	61	6	7	9	10
30	30	37	43	50	62	6	7	8	9
31	29	35	41	48	63	2	2	2	2

Таблица 9-36 – Таблица перехода состояний

pStateIdx	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
transIdxLPS	0	0	1	2	2	4	4	5	6	7	8	9	9	11	11	12
transIdxMPS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
pStateIdx	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
transIdxLPS	13	13	15	15	16	16	18	18	19	19	21	21	22	22	23	24
transIdxMPS	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
pStateIdx	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
transIdxLPS	24	25	26	26	27	27	28	29	29	30	30	30	31	32	32	33
transIdxMPS	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
pStateIdx	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
transIdxLPS	33	33	34	34	35	35	35	36	36	36	37	37	37	38	38	63
transIdxMPS	49	50	51	52	53	54	55	56	57	58	59	60	61	62	62	63

9.3.3.2.2 Процесс изменения нормировки в устройстве арифметического декодирования

Входы в этот процесс – биты из данных секции и переменные codIRange и codIOffset.

Выходы этого процесса – обновленные переменные codIRange и codIOffset.

Диаграмма последовательности изменения нормировки показана на рисунке 9-4. Текущее значение codIRange сначала сравнивают с 0x0100, а дальнейшие шаги следующие.

- Если codIRange больше или равно 0x0100, то изменения нормировки не требуется, и процесс RenormD заканчивается.
- Иначе (codIRange меньше, чем 0x0100) вводят цикл изменения нормировки. В этом цикле значение codIRange удваивают, т. е. сдвигают влево на 1, а единичный бит сдвигают на codIOffset, используя read_bits(1).

Поток битов не должен содержать данных, в результате которых значение codIOffset по завершении этого процесса окажется больше или равно codIRange.

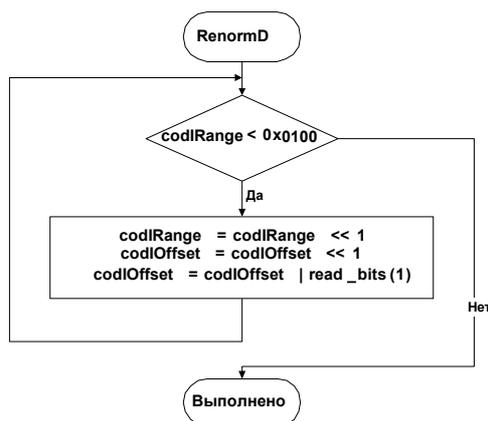


Рисунок 9-4 – Диаграмма последовательности изменения нормировки

9.3.3.2.3 Процесс обхода декодирования бинарных решений

Входы в этот процесс – биты из данных секции и переменные `codIRange` и `codIOffset`.

Выходы этого процесса – обновленная переменная `codIOffset` и декодированное значение `binVal`.

Процесс обхода декодирования активируют, если `bypassFlag` равно 1. На рисунке 9-5 показана диаграмма последовательности соответствующего процесса.

Сначала значение `codIOffset` удваивают, т. е. сдвигают влево на 1, а единичный бит сдвигают на `codIOffset`, используя `read_bits(1)`. Затем значение `codIOffset` сравнивают со значением `codIRange`, а дальнейшие шаги определены следующим образом:

- если `codIOffset` больше или равно `codIRange`, то переменную `binVal` устанавливают равной 1, а `codIOffset` уменьшают на `codIRange`;
- иначе (`codIOffset` меньше, чем `codIRange`) переменную `binVal` устанавливают на 0.

Поток битов не должен содержать данных, в результате которых значение `codIOffset` по завершении этого процесса окажется больше или равно `codIRange`.

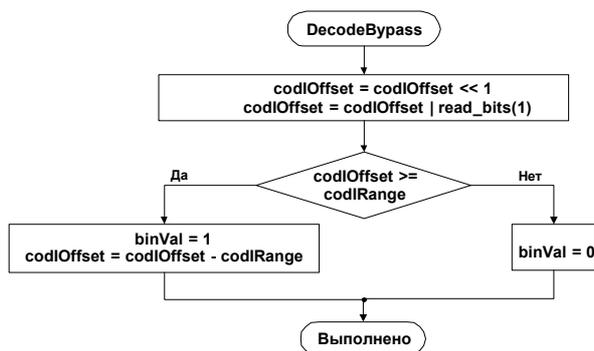


Рисунок 9-5 – Диаграмма последовательности процесса обхода декодирования

9.3.3.2.4 Процесс декодирования бинарного решения перед завершением

Входы в этот процесс – биты из данных секции и переменные `codIRange` и `codIOffset`.

Выходы этого процесса – обновленные переменные `codIRange` и `codIOffset` и декодированное значение `binVal`.

Эту особую рутинную процедуру декодирования применяют к декодированию `end_of_slice_flag` и к бину, который указывает соответствующий режим `I_PCM` значения `ctxIdx`, равного 276. На рисунке 9-6 показана диаграмма последовательности соответствующего процесса декодирования, который описывают следующим образом.

Сначала значение `codIRange` уменьшают на 2. Затем значение `codIOffset` сравнивают со значением `codIRange`, а дальнейшие шаги определены следующим образом.

- Если `codIOffset` больше или равно `codIRange`, то переменную `binVal` устанавливают равной 1, изменения нормировки не производят, а декодирование САВАС закончено. Последний введенный в регистр `codIOffset` бит равен 1. При декодировании `end_of_slice_flag` последний введенный в регистр `codIOffset` бит интерпретируют как `gbsp_stop_one_bit`.
- Иначе (`codIOffset` меньше `codIRange`) переменную `binVal` устанавливают на 0, процесс изменения нормировки выполняют, как это определено в п. 9.3.3.2.2.

ПРИМЕЧАНИЕ. – Эту процедуру можно также выполнить, используя `DecodeDecision(ctxIdx)` с `ctxIdx = 276`. В этом случае там, где декодированное значение равно 1, более семи битов могут быть считаны с помощью `DecodeDecision(ctxIdx)`, а процесс декодирования должен настраиваться указателем потока битов, должным образом декодированным согласно последующим элементам синтаксиса.

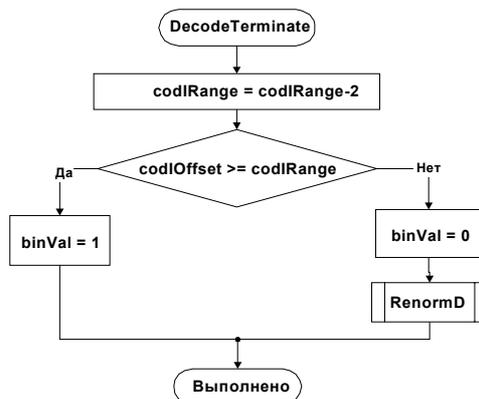


Рисунок 9-6 – Диаграмма последовательности декодирования решения перед завершением

9.3.4 Процесс арифметического кодирования (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Входы в этот процесс – решения, которые следует кодировать и записать.

Выходы этого процесса – биты, которые записаны в RBSP.

Этот информативный пункт описывает арифметическое кодирующее устройство, которое соответствует устройству арифметического декодирования, описанному в п. 9.3.3.2. Кодирующее устройство является по существу симметричным устройством декодирования, т. е. процедуры выполняют в том же самом порядке. В этом разделе описаны следующие процедуры: InitEncoder, EncodeDecision, EncodeBypass, EncodeTerminate, которым соответствуют InitDecoder, DecodeDecision, DecodeBypass и DecodeTerminate соответственно. Состояние устройства арифметического кодирования представлено значением переменной codILow, указывающей на нижний конец подинтервала, а значение переменной codIRange определяет соответствующий диапазон подинтервала.

9.3.4.1 Процесс инициализации устройства арифметического кодирования (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Процесс активируют перед кодированием первого макроблока секции и после кодирования любого pcm_alignment_zero_bit и всех данных pcm_sample_luma и pcm_sample_chroma макроблока типа I_PCM.

Выходы этого процесса – значения codILow, codIRange, firstBitFlag, bitsOutstanding и symCnt устройства арифметического кодирования.

Во время процедуры инициализации кодера codILow устанавливают равным 0, а codIRange устанавливают равным 0x01FE. Далее firstBitFlag устанавливают равным 1, а счетчики bitsOutstanding и symCnt устанавливают равными 0.

ПРИМЕЧАНИЕ. – Минимальная точность регистра, которую требуют для codILow, составляет 10 битов, а для CodIRange – 9 битов. Точность, которая требуется для счетчиков bitsOutstanding и symCnt, должна быть существенно выше, чтобы предотвратить переполнение соответствующих регистров. Если MaxBinCountInSlice указывает на максимальное общее число бинарных решений для кодирования в одной секции, то минимальная точность регистра, которая требуется для переменных bitsOutstanding и symCnt, задана значением $\text{Ceil}(\text{Log}_2(\text{MaxBinCountInSlice} + 1))$.

9.3.4.2 Процесс кодирования бинарного решения (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Входы в этот процесс – индекс контекста ctxIdx, значение binVal, которое следует кодировать, и переменные codIRange, codILow и symCnt.

Выходы этого процесса – переменные codIRange, codILow и symCnt.

На рисунке 9-7 показана диаграмма последовательности для кодирования одного решения. На первом этапе переменную codIRangeLPS находят следующим образом.

При заданном текущем значении codIRange codIRange отображают в индекс qCodIRangeIdx квантованного значения codIRange, используя уравнение 9-19. Значение qCodIRangeIdx и значение pStateIdx, объединенные с ctxIdx, используют для определения значения переменной rangeTabLPS (как это показано в таблице 9-33), которой присваивают значение codIRangeLPS. Значение codIRange – codIRangeLPS присваивают codIRange.

На втором этапе значение binVal сравнивают с valMPS, объединенным с ctxIdx. Если binVal отличается от valMPS, значение codIRange добавляют к codILow, а codIRange устанавливают равным значению codIRangeLPS. При заданном кодированном решении переход состояния выполняют, как это определено в п. 9.3.3.2.1.1. В зависимости от текущего значения codIRange производят изменение нормировки, как это определено в п. 9.3.4.3. Наконец, переменная symCnt увеличивается на 1.

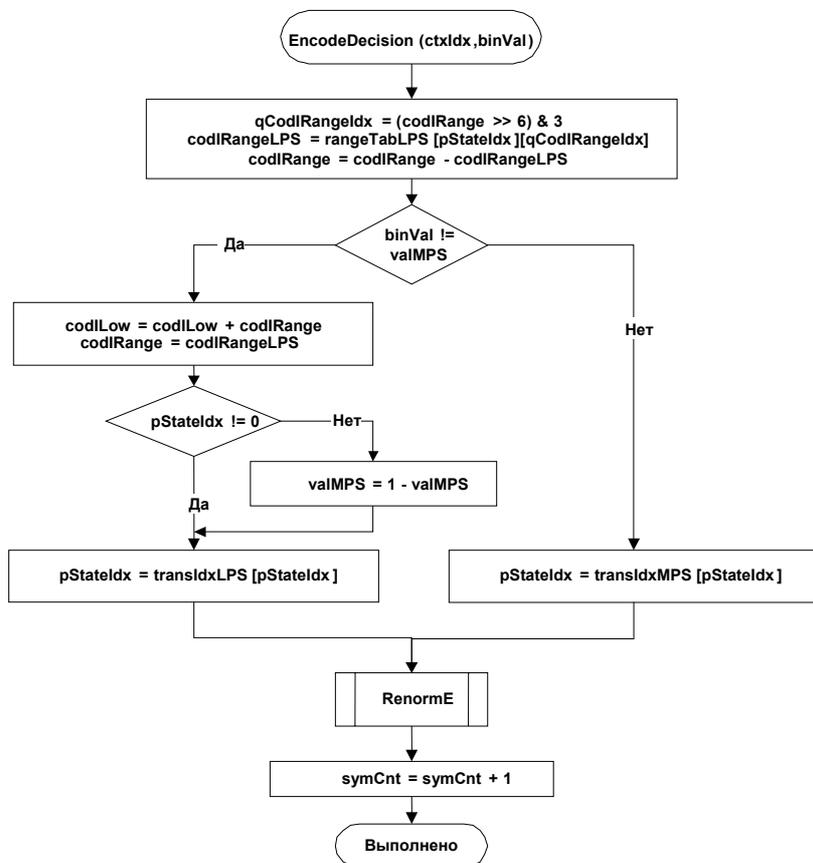


Рисунок 9-7 – Диаграмма последовательности кодирования решения

9.3.4.3 Процесс изменения нормировки в устройстве арифметического кодирования (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Входы в этот процесс – переменные codIRange, codILow, firstBitFlag и bitsOutstanding.

Выходы этого процесса – нуль или более битов, записанных в RBSP, и обновленные переменные codIRange, codILow, firstBitFlag и bitsOutstanding.

Изменение нормировки проиллюстрировано на рисунке 9-8.

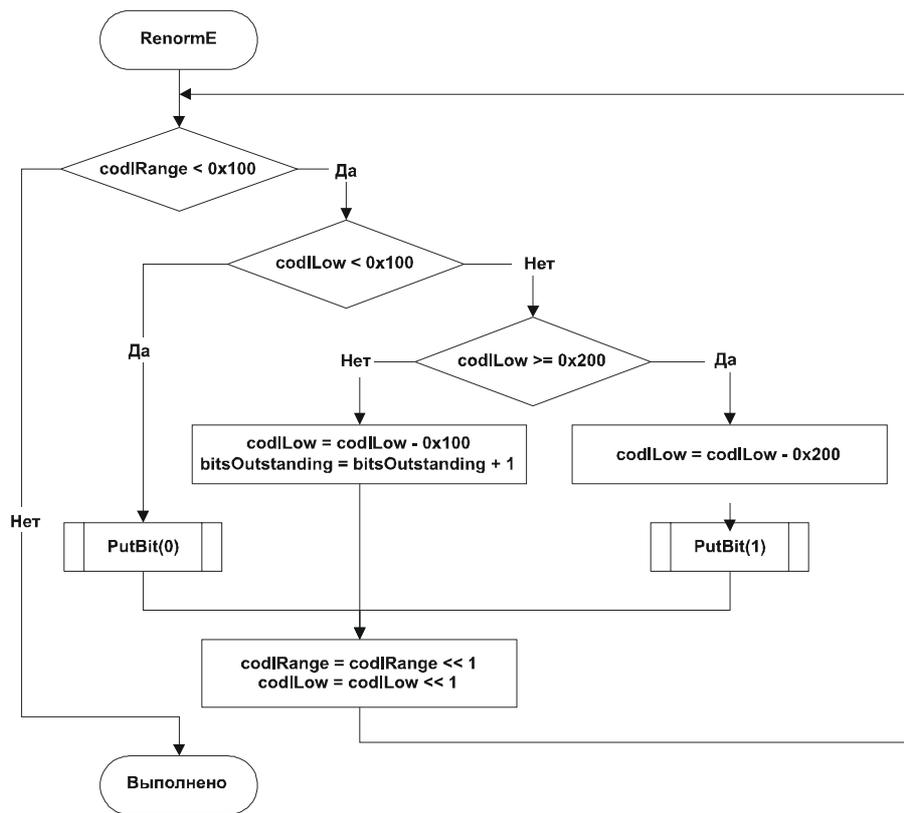


Рисунок 9-8 – Диаграмма последовательности изменения нормировки в кодере

Процедура PutBit(), показанная на рисунке 9-9, обеспечивает перенос управления. При этом используют функцию WriteBits(B, N), которая записывает N битов со значением (B) в поток битов и продвигает указатель потока битов вперед на N битов. Действие этой функции предполагает существование указателя потока битов с указанием позиции следующего бита, который следует записать в поток битов с помощью процесса кодирования.

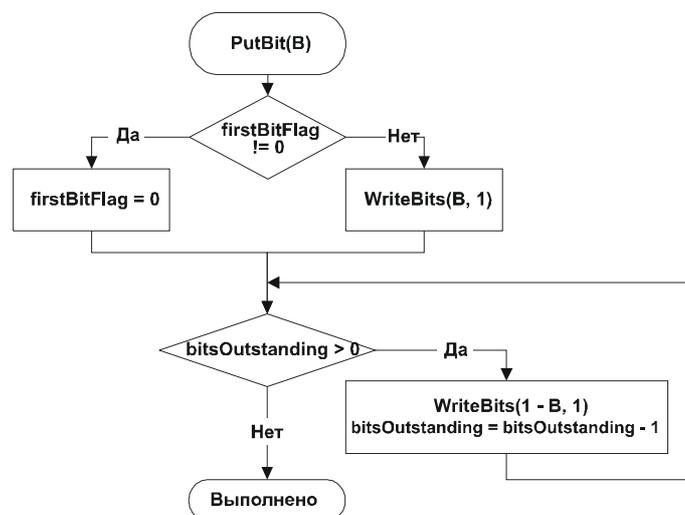


Рисунок 9-9 – Диаграмма последовательности PutBit(B)

9.3.4.4 Процесс обхода кодирования бинарных решений (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Входы в этот процесс – переменные binVal, codILow, codIRange, bitsOutstanding и symCnt.

Выход этого процесса – бит, записанный в RBSP, и обновленные переменные codILow, bitsOutstanding и symCnt.

Этот процесс кодирования применим ко всем бинарным решениям со значением bypassFlag, равным 1. Изменение нормировки включено в спецификацию этого процесса, как показано на рисунке 9-10.

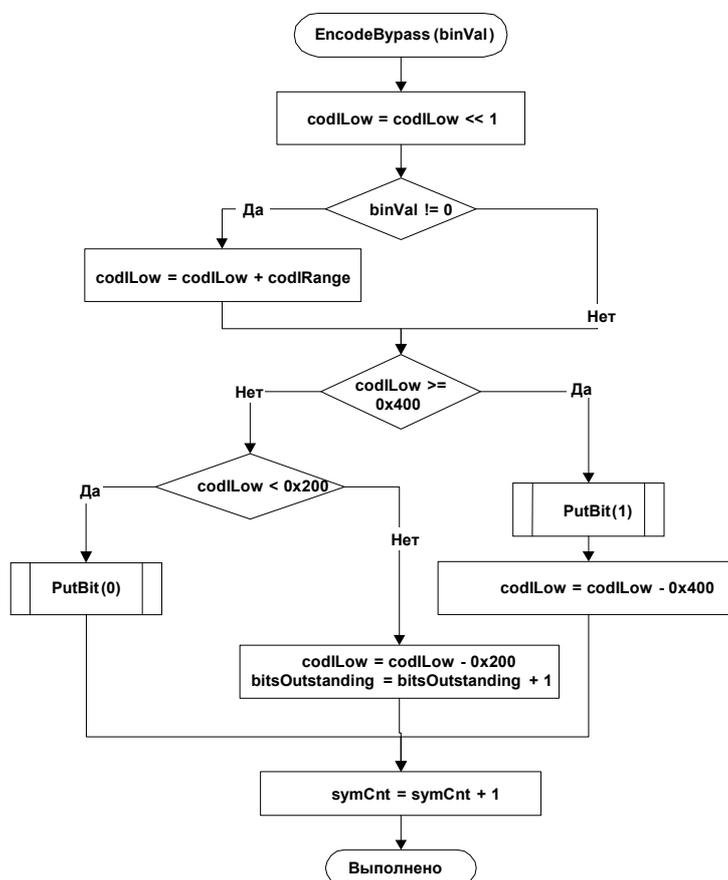


Рисунок 9-10 – Диаграмма последовательности обхода кодирования

9.3.4.5 Процесс кодирования бинарного решения перед завершением (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Входы в этот процесс – переменные binVal, codIRange, codILow, bitsOutstanding и symCnt.

Выходы этого процесса – нуль или более битов, записанных в RBSP, и обновленные переменные codILow, codIRange, bitsOutstanding и symCnt.

Общепринятый порядок кодирования, показанный на рисунке 9-11, применим к кодированию значения end_of_slice_flag и бина, указывающего тип I_PCM: mb_type. Оба значения связаны с ctxIdx, равным 276.

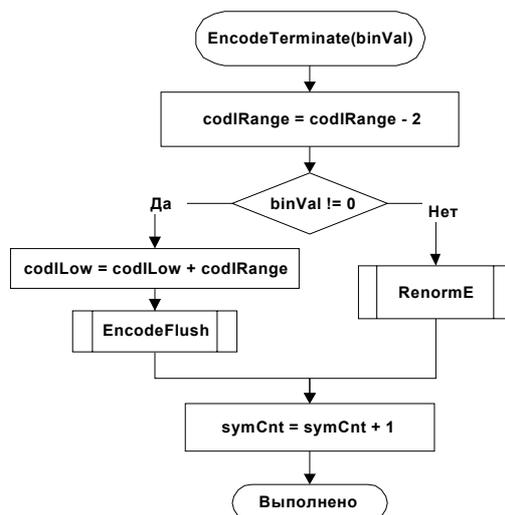


Рисунок 9-11 – Диаграмма последовательности кодирования решения перед завершением

Если предназначенное для кодирования значение binVal равно 1, то кодирование САВАС завершают и выполняют процедуру сдвига, показанную на рисунке 9-12. В процедуре сдвига последний бит, записанный с помощью WriteBits(B, N), равен 1. При кодировании end_of_slice_flag, последний бит интерпретируют как rbsp_stop_one_bit.

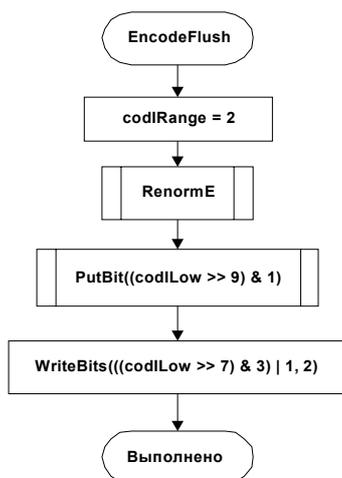


Рисунок 9-12 – Диаграмма последовательности сдвига при завершении

9.3.4.6 Процесс штаффинга байтов (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Процесс активируют после кодирования последнего макроблока последней секции изображения и после инкапсуляции.

Входы в этот процесс – число байтов NumBytesInVclNALunits из всех частей NAL VCL изображения, число макроблоков PicSizeInMbs в изображении и число бинарных символов BinCountsInNALunits, полученных в результате кодирования содержания всех частей NAL VCL изображения.

Выходы этого процесса – нуль или более байтов, присоединенных к блоку NAL.

Предположим, что переменная k установлена равной $\text{Ceil}(\text{Ceil}(3 * (32 * \text{BinCountsInNALunits} - \text{RawMbBits} * \text{PicSizeInMbs}) \div 1024) - \text{NumBytesInVclNALunits}) \div 3$). В зависимости от переменной k применяют следующее.

- Если k меньше или равно 0, к блоку NAL не присоединяют `cabac_zero_word`.
- Иначе (k больше 0) после инкапсуляции к блоку NAL k раз присоединяют 3-байтовую последовательность `0x000003`, где первые два байта `0x0000` представляют значение `cabac_zero_word`, а третий байт `0x03` представляет значение `emulation_prevention_three_byte`.

Приложение А

Профили и уровни

(Данное Приложение является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Профили и уровни определяют ограничения на потоки битов и поэтому ограничивают возможности, необходимые для декодирования потоков битов. Профили и уровни можно также использовать для указания совместимости применений отдельных декодеров.

ПРИМЕЧАНИЕ 1. – В настоящую Рекомендацию | Международный стандарт не включены отдельно выбранные "опции" декодера, так как это увеличило бы трудности совместимости.

Каждый профиль определяет подмножество алгоритмических особенностей и ограничений, которые должны поддерживать все декодеры, соответствующие этому профилю.

ПРИМЕЧАНИЕ 2. – Для того чтобы сделать возможным применение любых конкретных свойств, заложенных в профилях, кодеры не требуются.

Каждый уровень определяет набор ограничений на значения, которые могут накладываться элементами синтаксиса настоящей Рекомендации | Международного стандарта. Тот же набор определений уровней используют со всеми профилями, но частные применения могут поддерживать различный уровень для каждого поддерживаемого профиля. Уровни для любых заданных профилей, как правило, соответствуют обработке декодером нагрузки и возможностям его памяти.

А.1 Требования к возможностям видеodeкодера

Возможности видеodeкодеров, соответствующих настоящей Рекомендации | Международному стандарту, определены в понятиях способности декодировать видеопотоки, соответствующие ограничениям, накладываемым на профили и уровни, определенным в этом Приложении. Для каждого такого профиля также должны быть расширены уровни поддержки этого профиля.

Особые значения определены в этом Приложении для элементов синтаксиса `profile_idc` и `level_idc`. Все прочие значения `profile_idc` и `level_idc` зарезервированы для будущего использования МСЭ-Т | ИСО/МЭК.

ПРИМЕЧАНИЕ. – Декодеры не должны считать зарезервированные значения `profile_idc` или `level_idc`, которые могут находиться между значениями, определенными в настоящей Рекомендации | Международном стандарте, как некоторые промежуточные возможности между определенными профилями или уровнями. В действительности отсутствуют ограничения на метод, который был выбран МСЭ-Т | ИСО/МЭК для использования в будущем подобных зарезервированных значений.

А.2 Профили

А.2.1 Базовый профиль

На потоки битов, соответствующие базовому профилю, должны накладываться следующие ограничения.

- Могут быть представлены только типы секций I и P.
- Потоки блоков NAL не должны содержать значения `nal_unit_type` в диапазоне от 2 до 4 включительно.
- Наборы параметров последовательности должны иметь `frame_mbs_only_flag`, равное 1.
- В наборах параметров последовательности не должны присутствовать элементы синтаксиса `chroma_format_idc`, `bit_depth_luma_minus8`, `bit_depth_chroma_minus8`, `qpprime_y_zero_transform_bypass_flag` и `seq_scaling_matrix_present_flag`.
- Наборы параметров изображения должны иметь `weighted_pred_flag` и `weighted_bipred_idc`, оба равные 0.
- Наборы параметров изображения должны иметь `entropy_coding_mode_flag`, равное 0.
- Наборы параметров изображения должны иметь `num_slice_groups_minus1` в диапазоне от 0 до 7 включительно.
- В наборах параметров изображения не должны присутствовать элементы синтаксиса `transform_8x8_mode_flag`, `pic_scaling_matrix_present_flag` и `second_chroma_qp_index_offset`.
- Для базового профиля должен соблюдаться уровень ограничений, определенный в пункте А.3.

Соответствие потока битов базовому профилю определено значением `profile_idc`, равным 66.

Декодеры, соответствующие базовому профилю с определенным уровнем, должны быть в состоянии декодировать все потоки битов, в которых `profile_idc` равно 66 или `constraint_set0_flag` равно 1, а также в которых `level_idc` и `constraint_set3_flag` представляют уровень, меньший или равный определенному уровню.

А.2.2 Главный профиль

На потоки битов, соответствующие главному профилю, должны накладываться следующие ограничения.

- Могут быть представлены только типы секций I, P и B.
- Потоки блоков NAL не должны содержать значения `nal_unit_type` в диапазоне от 2 до 4 включительно.
- Произвольный порядок секций не допустим.
- В наборах параметров последовательности не должны присутствовать элементы синтаксиса `chroma_format_idc`, `bit_depth_luma_minus8`, `bit_depth_chroma_minus8`, `qpprime_y_zero_transform_bypass_flag` и `seq_scaling_matrix_present_flag`.
- Набор параметров изображения должен иметь значение `num_slice_groups_minus1`, равное только 0.
- Набор параметров изображения должен иметь значение `redundant_pic_cnt_present_flag`, равное только 0.
- В наборах параметров изображения не должны присутствовать элементы синтаксиса `transform_8x8_mode_flag`, `pic_scaling_matrix_present_flag` и `second_chroma_qp_index_offset`.
- Значение элемента синтаксиса `level_prefix` не должно превышать 15 (если элемент представлен).
- Для главного профиля должен соблюдаться уровень ограничений, определенный в пункте А.3.

Соответствие потока битов главному профилю определяется значением `profile_idc`, равным 77.

Декодеры, соответствующие главному профилю с определенным уровнем, должны быть в состоянии декодировать все потоки битов, в которых `profile_idc` равно 77 или `constraint_set1_flag` равно 1 и в которых `level_idc` и `constraint_set3_flag` представляют уровень, меньший или равный определенному уровню.

А.2.3 Расширенный профиль

На потоки битов, соответствующие расширенному профилю, должны накладываться следующие ограничения.

- Последовательность набора параметров должна иметь `direct_8x8_inference_flag`, равное 1.
- В наборах параметров последовательности не должны присутствовать элементы синтаксиса `chroma_format_idc`, `bit_depth_luma_minus8`, `bit_depth_chroma_minus8`, `qpprime_y_zero_transform_bypass_flag` и `seq_scaling_matrix_present_flag`.
- Набор параметров изображения должен иметь значение `entropy_coding_mode_flag`, равное 0.
- Набор параметров изображения должен иметь значение `num_slice_groups_minus1` в диапазоне от 0 до 7 включительно.
- В наборах параметров изображения не должны присутствовать элементы синтаксиса `transform_8x8_mode_flag`, `pic_scaling_matrix_present_flag` и `second_chroma_qp_index_offset`.
- Значение элемента синтаксиса `level_prefix` не должно превышать 15 (если элемент представлен).
- Для расширенного профиля должен соблюдаться уровень ограничений, определенный в пункте А.3.

Соответствие потока битов расширенному профилю определяется значением `profile_idc`, равным 88.

Декодеры, соответствующие расширенному профилю с определенным уровнем, должны быть в состоянии декодировать все потоки битов, в которых `profile_idc` равно 88 или `constraint_set2_flag` равно 1 и в которых `level_idc` представляет уровень, меньший или равный определенному уровню.

Декодеры, соответствующие расширенному профилю с определенным уровнем, должны быть также в состоянии декодировать все потоки битов, в которых `profile_idc` равно 66 или `constraint_set0_flag` равно 1 и в которых `level_idc` и `constraint_set3_flag` представляют уровень, меньший или равный определенному уровню.

А.2.4 Высокий профиль

На потоки битов, соответствующие высокому профилю, должны накладываться следующие ограничения.

- Могут быть представлены только типы секций I, P и B.
- Потоки блоков NAL не должны содержать значения `nal_unit_type` в диапазоне от 2 до 4 включительно.
- Произвольный порядок секций не допустим.
- Наборы параметров изображения должны иметь `num_slice_groups_minus1`, равное только 0.
- Наборы параметров изображения должны иметь `redundant_pic_cnt_present_flag`, равное только 0.
- Наборы параметров последовательности должны иметь `chroma_format_idc` в диапазоне от 0 до 1 включительно.
- Наборы параметров последовательности должны иметь `bit_depth_luma_minus8`, равное только 0.
- Наборы параметров последовательности должны иметь `bit_depth_chroma_minus8`, равное только 0.
- Наборы параметров последовательности должны иметь `qpprime_y_zero_transform_bypass_flag`, равное только 0.
- Для высокого профиля должен соблюдаться уровень ограничений, определенный в пункте А.3.

Соответствие потока битов высокому профилю определяется значением `profile_idc`, равным 100. Декодеры, соответствующие высокому профилю с определенным уровнем, должны быть в состоянии декодировать все потоки битов, в которых `level_idc` и `constraint_set3_flag` представляют уровень, меньший или равный определенному уровню, и в котором одно или оба следующих условий являются истиной:

- `profile_idc` равно 77 или 100, или
- `constraint_set1_flag` равно 1.

A.2.5 Высокий 10 профиль

На потоки битов, соответствующие высокому 10 профилю, должны накладываться следующие ограничения.

- Могут быть представлены только типы секций I, P и B.
- Потоки блоков NAL не должны содержать значения `nal_unit_type` в диапазоне от 2 до 4 включительно.
- Произвольный порядок секций не допустим.
- Наборы параметров изображения должны иметь `num_slice_groups_minus1`, равное только 0.
- Наборы параметров изображения должны иметь `redundant_pic_cnt_present_flag`, равное только 0.
- Наборы параметров последовательности должны иметь `chroma_format_idc` в диапазоне от 0 до 1 включительно.
- Наборы параметров последовательности должны иметь `bit_depth_luma_minus8` в диапазоне от 0 до 2 включительно.
- Наборы параметров последовательности должны иметь `bit_depth_chroma_minus8` в диапазоне от 0 до 2 включительно.
- Наборы параметров последовательности должны иметь `qpprime_y_zero_transform_bypass_flag`, равное только 0.
- Для высокого 10 профиля должен соблюдаться уровень ограничений, определенный в пункте A.3.

Соответствие потока битов высокому 10 профилю определяется значением `profile_idc`, равным 110. Декодеры, соответствующие высокому 10 профилю с определенным уровнем, должны быть в состоянии декодировать все потоки битов, в которых `level_idc` и `constraint_set3_flag` представляют уровень, меньший или равный определенному уровню, и в котором одно или оба следующих условий являются истиной:

- `profile_idc` равно 77, 100 или 110, или
- `constraint_set1_flag` равно 1.

A.2.6 Высокий 4:2:2 профиль

На потоки битов, соответствующие высокому 4:2:2 профилю, должны накладываться следующие ограничения.

- Могут быть представлены только типы секций I, P и B.
- Потоки блоков NAL не должны содержать значения `nal_unit_type` в диапазоне от 2 до 4 включительно.
- Произвольный порядок секций не допустим.
- Наборы параметров изображения должны иметь `num_slice_groups_minus1`, равное только 0.
- Наборы параметров изображения должны иметь `redundant_pic_cnt_present_flag`, равное только 0.
- Наборы параметров последовательности должны иметь `chroma_format_idc` в диапазоне от 0 до 2 включительно.
- Наборы параметров последовательности должны иметь `bit_depth_luma_minus8` в диапазоне от 0 до 2 включительно.
- Наборы параметров последовательности должны иметь `bit_depth_chroma_minus8` в диапазоне от 0 до 2 включительно.
- Наборы параметров последовательности должны иметь `qpprime_y_zero_transform_bypass_flag`, равное только 0.
- Для высокого 4:2:2 профиля должен соблюдаться уровень ограничений, определенный в пункте A.3.

Соответствие потока битов высокому 4:2:2 профилю определяется значением `profile_idc`, равным 122. Декодеры, соответствующие высокому 4:2:2 профилю с определенным уровнем, должны быть в состоянии декодировать все потоки битов, в которых `level_idc` и `constraint_set3_flag` представляют уровень, меньший или равный определенному уровню, и в котором одно или оба следующих условий являются истиной:

- `profile_idc` равно 77, 100, 110 или 122; или
- `constraint_set1_flag` равно 1.

А.2.7 Высокий 4:4:4 профиль

На потоки битов, соответствующие высокому 4:4:4 профилю, должны накладываться следующие ограничения.

- Могут быть представлены только типы секций I, P и B.
- Потоки блоков NAL не должны содержать значения `nal_unit_type` в диапазоне от 2 до 4 включительно.
- Произвольный порядок секций не допустим.
- Наборы параметров изображения должны иметь `num_slice_groups_minus1`, равное только 0.
- Наборы параметров изображения должны иметь `redundant_pic_cnt_present_flag`, равное только 0.
- Наборы параметров последовательности должны иметь `bit_depth_luma_minus8` в диапазоне от 0 до 4 включительно.
- Наборы параметров последовательности должны иметь `bit_depth_chroma_minus8` в диапазоне от 0 до 2 включительно.
- Для высокого 4:4:4 профиля должен соблюдаться уровень ограничений, определенный в пункте А.3.

Соответствие потока битов высокому 4:4:4 профилю определяется значением `profile_idc`, равным 144. Декодеры, соответствующие высокому 4:4:4 профилю с определенным уровнем, должны быть в состоянии декодировать все потоки битов, в которых `level_idc` и `constraint_set3_flag` представляют уровень, меньший или равный определенному уровню, и в котором одно или оба следующих условий являются истиной:

- `profile_idc` равно 77, 100, 110, 122 или 144, или
- `constraint_set1_flag` равно 1.

А.3 Уровни

В этом Приложении для выражения ограничений определено следующее.

- Положим, что блок доступа n – это n -й блок доступа в порядке декодирования, где первым блоком доступа является блок доступа 0.
- Положим, что изображение n – это первично кодированное изображение или соответствующее декодированное изображение блока доступа n .

А.3.1 Границы уровней, общие для базового, главного и расширенного профилей

Положим, что переменную `fR` находят следующим образом:

- если изображение n – это кадр, `fR` устанавливают равным $1 \div 172$;
- иначе (изображение n – это поле) `fR` устанавливают равным $1 \div (172 * 2)$.

На потоки битов, соответствующие базовому, главному или расширенному профилям на некотором определенном уровне, должны накладываться следующие ограничения.

- a) Номинальное время удаления блока доступа n ($n > 0$) из CPB, как это определено в п. С.1.2, удовлетворяет ограничению $t_{r,n}(n) - t_r(n-1)$ больше или равно $\text{Max}(\text{PicSizeInMbs} \div \text{MaxMBPS}, fR)$, где `MaxMBPS` – значение, определенное в таблице А-1, которое применяют к изображению $n-1$, а `PicSizeInMbs` – это число макроблоков в изображении $n-1$.
- b) Разность между последовательными временами выхода изображений из DPB, как это определено в п. С.2.2, удовлетворяет ограничению $\Delta t_{o,dpb}(n) \geq \text{Max}(\text{PicSizeInMbs} \div \text{MaxMBPS}, fR)$, где `MaxMBPS` – значение, указанное в таблице А-1 для изображения n , а `PicSizeInMbs` – число макроблоков изображения n , при условии, что изображение n – это изображение, которое является выходным и не последним изображением в потоке битов на выходе.
- c) Сумма переменных `NumBytesInNALunit` для блока доступа 0 меньше или равна $384 * (\text{PicSizeInMbs} + \text{MaxMBPS} * (t_r(0) - t_{r,n}(0))) \div \text{MinCR}$, где `MaxMBPS` и `MinCR` – значения, определенные в таблице А-1, которые применимы к изображению 0, а `PicSizeInMbs` – число макроблоков в изображении 0.
- d) Сумма переменных `NumBytesInNALunit` для блока доступа n ($n > 0$) меньше или равна $384 * \text{MaxMBPS} * (t_r(n) - t_r(n-1)) \div \text{MinCR}$, где `MaxMBPS` и `MinCR` – значения, определенные в таблице А-1, которые применимы к изображению n .
- e) $\text{PicWidthInMbs} * \text{FrameHeightInMbs} \leq \text{MaxFS}$, где `MaxFS` определено в таблице А-1.
- f) $\text{PicWidthInMbs} \leq \text{Sqrt}(\text{MaxFS} * 8)$.
- g) $\text{FrameHeightInMbs} \leq \text{Sqrt}(\text{MaxFS} * 8)$.
- h) $\text{max_dec_frame_buffering} \leq \text{MaxDpbSize}$, где `MaxDpbSize` равно $\text{Min}(1024 * \text{MaxDPB} / (\text{PicWidthInMbs} * \text{FrameHeightInMbs} * 384), 16)$, а `MaxDPB` задано в таблице А-1 в блоках по 1024 байтов.

- i) Если `vcl_hrd_parameters_present_flag` равно 1, то параметры VCL HRD $\text{BitRate}[\text{SchedSelIdx}] \leq 1000 * \text{MaxBR}$ и $\text{CpbSize}[\text{SchedSelIdx}] \leq 1000 * \text{MaxCPB}$, по крайней мере, для одного значения `SchedSelIdx`, где $\text{BitRate}[\text{SchedSelIdx}]$ задано уравнением E-38, а $\text{CpbSize}[\text{SchedSelIdx}]$ задано уравнением E-14. Значения `MaxBR` и `MaxCPB` определены в таблице A-1 в блоках 1000 битов/с и 1000 битов, соответственно. Поток битов должен удовлетворять этим условиям, по крайней мере, для одного значения `SchedSelIdx` в диапазоне от 0 до `spb_cnt_minus1` включительно.
- j) Если `nal_hrd_parameters_present_flag` равно 1, то параметры NAL HRD, $\text{BitRate}[\text{SchedSelIdx}] \leq 1200 * \text{MaxBR}$ и $\text{CpbSize}[\text{SchedSelIdx}] \leq 1200 * \text{MaxCPB}$, по крайней мере, для одного значения `SchedSelIdx`, где $\text{BitRate}[\text{SchedSelIdx}]$ задано уравнением E-38, а $\text{CpbSize}[\text{SchedSelIdx}]$ задано уравнением E-14. Значения `MaxBR` и `MaxCPB` определены в таблице A-1 в блоках 1200 бит/с и 1200 битов, соответственно. Поток битов должен удовлетворять этим условиям, по крайней мере, для одного значения `SchedSelIdx` в диапазоне от 0 до `spb_cnt_minus1`.
- k) Диапазон вертикальных компонентов вектора движения для векторов движения яркости, не превосходит `MaxVmvR` в блоках образцов кадров яркости, где `MaxVmvR` определено в таблице A-1.
- ПРИМЕЧАНИЕ 1. – Если `chroma_format_idc` равно 1 и текущим макроблоком является макроблок поля, диапазон компонентов вектора движения для векторов движения цветности может превышать `MaxVmvR` в блоках образцов кадра яркости, в силу метода вывода векторов движения цветности, определенного в п. 8.4.1.4.
- l) Диапазон вектора горизонтального движения не превосходит диапазон от –2048 до 2047,75 включительно в блоках образцов яркости.
- m) Число векторов движения на два последовательных макроблока в порядке декодирования (это также применимо к общему числу от последнего макроблока секции и первого макроблока следующей секции в порядке декодирования, и в частности применимо также к общему числу от последнего макроблока последней секции изображения и первого макроблока первой секции следующего изображения в порядке декодирования) не превышает `MaxMvsPer2Mb`, где `MaxMvsPer2Mb` определяется в таблице A-1. Число векторов движения для каждого макроблока – это значение переменной `MvCnt` после завершения процесса `intra` или `inter` предсказания макроблока.
- n) Число битов данных `macroblock_layer()` для любого макроблока не превышает 3200. В зависимости от `entropy_coding_mode_flag` биты данных `macroblock_layer()` подсчитываются следующим образом:
- если `entropy_coding_mode_flag` равно 0, число битов данных `macroblock_layer()` задано числом битов в структуре синтаксиса макроблока `macroblock_layer()`;
 - иначе (`entropy_coding_mode_flag` равно 1) число битов данных макроблока `macroblock_layer()` задано числом раз `read_bits(1)`, указанным в пп. 9.3.3.2.2 и 9.3.3.2.3, если синтаксический анализ `macroblock_layer()` связан с этим макроблоком.

В таблице A-1 указаны границы каждого уровня. Входы в таблице A-1, помеченные как "-", указывают на отсутствие соответствующей границы. Для целей сравнения возможностей уровня необходимо рассматривать уровень как низший (высший) относительно какого-либо другого уровня, если по сравнению с этим другим уровнем он находится ближе к верхней (нижней) строке таблицы A-1.

Уровень, которому соответствует поток битов, должен быть указан элементами синтаксиса `level_idc` и `constraint_set3_flag` следующим образом:

- если `level_idc` равно 11, а `constraint_set3_flag` равно 1, указанным уровнем является уровень 1b;
- иначе (`level_idc` не равно 11 или `constraint_set3_flag` не равно 1) значение `level_idc` должно устанавливаться равным значению десятикратного номера уровня, определенного в таблице A-1, а значение `constraint_set3_flag` должно устанавливаться равным 0.

Таблица А-1 – Границы уровней

Номер уровня	Макс. скорость обработки макроблока MaxMBPS (Мб/с)	Макс. размер кадра MaxFS (Мб)	Макс. размер буфера декодированного изображения MaxDPB (1024 байта для 4:2:0)	Макс. скорость видеосигнала MaxBR (1000 бит/с, сrbBrVclFactor бит/с или сrbBrNalFactor бит/с)	Макс. размер СРВ MaxCPB (1000 бит или 1200 бит, сrbBrVclFactor бит/с или сrbBrNalFactor бит/с)	Диапазон вертикального компонента MV MaxVmvR (образцы кадров яркости)	Мин. коэф. сжатия MinCR	Макс. число векторов движения на два последовательн. МВ MaxMvsPer2Mb
1	1 485	99	148,5	64	175	[-64,+63,75]	2	–
1b	1 485	99	148,5	128	350	[-64,+63,75]	2	–
1.1	3 000	396	337,5	192	500	[-128,+127,75]	2	–
1.2	6 000	396	891,0	384	1 000	[-128,+127,75]	2	–
1.3	11 880	396	891,0	768	2 000	[-128,+127,75]	2	–
2	11 880	396	891,0	2 000	2 000	[-128,+127,75]	2	–
2.1	19 800	792	1 782,0	4 000	4 000	[-256,+255,75]	2	–
2.2	20 250	1 620	3 037,5	4 000	4 000	[-256,+255,75]	2	–
3	40 500	1 620	3 037,5	10 000	10 000	[-256,+255,75]	2	32
3.1	108 000	3 600	6 750,0	14 000	14 000	[-512,+511,75]	4	16
3.2	216 000	5 120	7 680,0	20 000	20 000	[-512,+511,75]	4	16
4	245 760	8 192	12 288,0	20 000	25 000	[-512,+511,75]	4	16
4.1	245 760	8 192	12 288,0	50 000	62 500	[-512,+511,75]	2	16
4.2	522 240	8 704	13 056,0	50 000	62 500	[-512,+511,75]	2	16
5	589 824	22 080	41 400,0	135 000	135 000	[-512,+511,75]	2	16
5.1	983 040	36 864	69 120,0	240 000	240 000	[-512,+511,75]	2	16

Уровни с нецелыми номерами в таблице А-1 считают "промежуточными уровнями".

ПРИМЕЧАНИЕ 2. – Все уровни имеют одинаковый статус, но некоторые приложения могут выбирать для использования только целочисленные уровни.

Информативный п. А.3.4 показывает действие этих границ на частоту кадров для разных примеров форматов изображений.

А.3.2 Границы уровней, общие для высокого, высокого 10, высокого 4:2:2 и высокого 4:4:4 профилей

Положим, что переменную fR находят следующим образом:

- если изображение n – это кадр, fR устанавливают равным $1 \div 172$;
- иначе (изображение n – это поле) fR устанавливают равным $1 \div (172 * 2)$.

На потоки битов, соответствующие высокому, высокому 10, высокому 4:2:2 и высокому 4:4:4 профилям на некотором определенном уровне, должны накладываться следующие ограничения.

- а) Номинальное время удаления блока доступа n (с n > 0) из СРВ, как это определено в п. С.1.2, удовлетворяет ограничению в том, что $t_{r,n}(n) - t_r(n-1)$ больше или равно $\text{Max}(\text{PicSizeInMbs} \div \text{MaxMBPS}, fR)$, где MaxMBPS – значение, определенное в таблице А-1, которое применяют к изображению n-1, а PicSizeInMbs – это число макроблоков в изображении n-1.
- б) Разность между последовательными значениями времени выхода изображений из DPB, как это определено в п. С.2.2, удовлетворяет ограничению в том, что $\Delta t_{o,dpb}(n) \geq \text{Max}(\text{PicSizeInMbs} \div \text{MaxMBPS}, fR)$, где MaxMBPS – значение, указанное в таблице А-1 для изображения n, а PicSizeInMbs – число макроблоков изображения n, при условии, что изображение n – это изображение, которое является выходным и не последним изображением в потоке битов на выходе.
- в) $\text{PicWidthInMbs} * \text{FrameHeightInMbs} \leq \text{MaxFS}$, где MaxFS определяется в таблице А-1.

- d) $\text{PicWidthInMbs} \leq \sqrt{\text{MaxFS} * 8}$.
- e) $\text{FrameHeightInMbs} \leq \sqrt{\text{MaxFS} * 8}$.
- f) $\text{max_dec_frame_buffering} \leq \text{MaxDpbSize}$, где MaxDpbSize равно $\text{Min}(1024 * \text{MaxDPB} / (\text{PicWidthInMbs} * \text{FrameHeightInMbs} * 384), 16)$, а MaxDPB определяется в таблице А-1.
- g) Диапазон вертикальных компонентов вектора движения не превосходит MaxVmvR в блоках образцов кадров яркости, где MaxVmvR определено в таблице А-1.
- h) Диапазон горизонтальных компонентов вектора движения не выходит за пределы диапазона от -2048 до $2047,75$ включительно в блоках образцов яркости.
- i) Число векторов движения на два последовательных макроблока в порядке декодирования (это также применимо к общему числу от последнего макроблока секции и первого макроблока следующей секции в порядке декодирования) не превышает MaxMvsPer2Mb , где MaxMvsPer2Mb определяется в таблице А-1. Число векторов движения для каждого макроблока – это значение переменной MvCnt после завершения процесса *intra* или *inter* предсказания макроблока.
- j) Число битов данных $\text{macroblock_layer}()$ для любого макроблока не превышает $128 + \text{RawMbBits}$. В зависимости от $\text{entropy_coding_mode_flag}$ биты данных $\text{macroblock_layer}()$ подсчитываются следующим образом:
 - если $\text{entropy_coding_mode_flag}$ равно 0, число битов данных $\text{macroblock_layer}()$ задано числом битов в структуре синтаксиса макроблока $\text{macroblock_layer}()$;
 - иначе ($\text{entropy_coding_mode_flag}$ равно 1) число битов данных макроблока $\text{macroblock_layer}()$ задано числом раз $\text{read_bits}(1)$, указанным в пп. 9.3.3.2.2 и 9.3.3.2.3, если синтаксический анализ $\text{macroblock_layer}()$ связан с этим макроблоком.

В таблице А-1 указаны границы каждого уровня. Входы в таблице А-1, помеченные как "-", указывают на отсутствие соответствующей границы.

Уровень, которому соответствует поток битов, должен быть указан элементом синтаксиса level_idc следующим образом:

- если level_idc равно 9, указанным уровнем является уровень 1b;
- иначе (level_idc не равно 9) level_idc должно устанавливаться равным значению десятикратного номера уровня, определенного в таблице А-1.

А.3.3 Границы уровней, определяемые профилем

- a) В потоках битов, соответствующих главному, высокому, высокому 10, высокому 4:2:2 или высокому 4:4:4 профилю, время удаления блока доступа 0 должно удовлетворять требованию на число секций в изображении 0: меньше или равно $(\text{PicSizeInMbs} + \text{MaxMBPS} * (t_r(0) - t_{r,n}(0))) \div \text{SliceRate}$, где SliceRate – значение, определенное в таблице А-4, которое используют для изображения 0.
- b) В потоках битов, соответствующих главному, высокому, высокому 10, высокому 4:2:2 или высокому 4:4:4 профилю, разность между последовательными временами удаления блоков доступа n и $n - 1$ ($n > 0$) должна удовлетворять ограничению на число секций в изображении n : меньше или равно $\text{MaxMBPS} * (t_r(n) - t_r(n - 1)) \div \text{SliceRate}$, где SliceRate – значение, определенное в таблице А-4, которое используют для изображения n .
- c) В потоках битов, соответствующих главному, высокому, высокому 10, высокому 4:2:2 или высокому 4:4:4 профилю, наборы последовательность параметров должны иметь $\text{direct_8x8_inference_flag}$, равное 1, для всех уровней, определенных в таблице А-4.
 ПРИМЕЧАНИЕ 1. – Значение $\text{direct_8x8_inference_flag}$ не имеет отношения к базовому профилю, поскольку этот профиль не разрешает использовать секции типа В (определенные в п. А.2.1), а $\text{direct_8x8_inference_flag}$, равное 1, не разрешает это для всех уровней расширенного профиля (определенных в п. А.2.3).
- d) В потоках битов, соответствующих главному, высокому, высокому 10, высокому 4:2:2, высокому 4:4:4 или расширенному профилю, наборы параметров последовательности должны иметь $\text{frame_mbs_only_flag}$, равное 1, для уровней, определенных в таблице А-4 для главного, высокого, высокого 10, высокого 4:2:2 или высокого 4:4:4 профиля, а в таблице А-3 – для расширенного профиля.
 ПРИМЕЧАНИЕ 2. – Значение $\text{frame_mbs_only_flag}$ равно 1 для всех уровней базового профиля (определенных в п. А.2.1).
- e) В потоках битов, соответствующих главному, высокому, высокому 10, высокому 4:2:2 или высокому 4:4:4 профилю, значение sub_mb_type в макроблоках В не должно быть равно B_Bi_8x4 , B_Bi_4x8 , или B_Bi_4x4 для уровней, в которых MinLumaBiPredSize показано как 8×8 в таблице А-4 для главного, высокого, высокого 10, высокого 4:2:2 или высокого 4:4:4 профиля и в таблице А-5 для расширенного профиля.

- f) В потоках битов, соответствующих главному и расширенному профилям, значение $(xInt_{max} - xInt_{min} + 6) * (yInt_{max} - yInt_{min} + 6) \leq MaxSubMbRectSize$ в макроблоках, кодированных с mb_type , равным P_8x8 , $P_8x8ref0$ или V_8x8 для всех вызываемых процедур процесса, определенных в п. 8.4.2.2.1 и используемых, чтобы создать массив предсказанных образцов яркости для одного списка контрольных изображений (списка контрольных изображений 0 или списка контрольных изображений 1) для каждого субмакроблока $8x8$, где $NumSubMbPart(sub_mb_type) > 1$, а $MaxSubMbRectSize$ определено в таблице А-3 для базового профиля и в таблице А-5 для расширенного профиля, и
- $xInt_{min}$ – это минимальное значение $xInt_L$ среди всех предсказанных образцов яркости для субмакроблока;
 - $xInt_{max}$ – это максимальное значение $xInt_L$ среди всех предсказанных образцов яркости для субмакроблока;
 - $yInt_{min}$ – это минимальное значение $yInt_L$ среди всех предсказанных образцов яркости для субмакроблока;
 - $yInt_{max}$ – это максимальное значение $yInt_L$ среди всех предсказанных образцов яркости для субмакроблока.
- g) В потоках битов, соответствующих высокому 10, высокому 4:2:2 или высокому 4:4:4 профилю, для параметров VCL HRD: $BitRate[SchedSelIdx] \leq cpbBrVclFactor * MaxBR$ и $CpbSize[SchedSelIdx] \leq cpbBrVclFactor * MaxCPB$ для, по крайней мере, одного значения $SchedSelIdx$, где $cpbBrVclFactor$ определяется в таблице А-2, $BitRate[SchedSelIdx]$ определяется уравнением Е-37, а $CpbSize[SchedSelIdx]$ определяется уравнением Е-38, если $vcl_hrd_parameters_present_flag$ равно 1. $MaxBR$ и $MaxCPB$ определяются в таблице А-1 в единицах $cpbBrVclFactor$ бит/с и $cpbBrVclFactor$ битов, соответственно. Поток битов должен удовлетворять этим условиям для, по крайней мере, одного значения $SchedSelIdx$ в диапазоне от 0 до cpb_cnt_minus1 включительно.
- h) В потоках битов, соответствующих высокому 10, высокому 4:2:2 или высокому 4:4:4 профилю, для параметров NAL HRD: $BitRate[SchedSelIdx] \leq cpbBrNalFactor * MaxBR$ и $CpbSize[SchedSelIdx] \leq cpbBrNalFactor * MaxCPB$ для, по крайней мере, одного значения $SchedSelIdx$, где $cpbBrNalFactor$ определяется в таблице А-2, $BitRate[SchedSelIdx]$ определяется уравнением Е-37, а $CpbSize[SchedSelIdx]$ определяется уравнением Е-38, если $nal_hrd_parameters_present_flag$ равно 1. $MaxBR$ и $MaxCPB$ определяются в таблице А-1 в единицах $cpbBrNalFactor$ бит/с и $cpbBrNalFactor$ битов, соответственно. Поток битов должен удовлетворять этим условиям для, по крайней мере, одного значения $SchedSelIdx$ в диапазоне от 0 до cpb_cnt_minus1 .

Таблица А-2 – Спецификация $cpbBrVclFactor$ и $cpbBrNalFactor$

Профиль	$cpbBrVclFactor$	$cpbBrNalFactor$
Высокий	1 250	1 500
Высокий 10	3 000	3 600
Высокий 4:2:2	4 000	4 800
Высокий 4:4:4	4 000	4 800

А.3.3.1 Границы базового профиля

Таблица А-3 определяет для каждого уровня границы, которые являются конкретными для потоков битов, соответствующих базовому профилю. Входы в таблице А-3, помеченные как "-", указывают на отсутствие соответствующей границы.

Таблица А-3 – Границы уровней базового профиля

Номер уровня	MaxSubMbRectSize
1	576
1b	576
1.1	576
1.2	576
1.3	576
2	576
2.1	576
2.2	576
3	576
3.1	–
3.2	–
4	–
4.1	–
4.2	–
5	–
5.1	–

А.3.3.2 Границы главного, высокого, высокого 10, высокого 4:2:2 или высокого 4:4:4 профиля

Таблица А-4 определяет для каждого уровня границы, которые являются конкретными для потоков битов, соответствующих главному, высокому, высокому 10, высокому 4:2:2 или высокому 4:4:4 профилю. Входы в таблице А-4, помеченные как "–", указывают на отсутствие соответствующей границы.

Таблица А-4 – Границы уровней главного, высокого, высокого 10, высокого 4:2:2 или высокого 4:4:4 профиля

Номер уровня	SliceRate	MinLumaBiPredSize	direct_8x8_inference_flag	frame_mbs_only_flag
1	–	–	–	1
1b	–	–	–	1
1.1	–	–	–	1
1.2	–	–	–	1
1.3	–	–	–	1
2	–	–	–	1
2.1	–	–	–	–
2.2	–	–	–	–
3	22	–	1	–
3.1	60	8x8	1	–
3.2	60	8x8	1	–
4	60	8x8	1	–
4.1	24	8x8	1	–
4.2	24	8x8	1	1
5	24	8x8	1	1
5.1	24	8x8	1	1

А.3.3.3 Границы расширенного профиля

Таблица А-5 определяет для каждого уровня границы, которые являются конкретными для потоков битов, соответствующих расширенному профилю. Входы в таблице А-5, помеченные как "-", указывают на отсутствие соответствующей границы.

Таблица А-5 – Границы уровней расширенного профиля

Номер уровня	MaxSubMbRectSize	MinLumaBiPredSize	frame_mbs_only_flag
1	576	–	1
1b	576	–	1
1.1	576	–	1
1.2	576	–	1
1.3	576	–	1
2	576	–	1
2.1	576	–	–
2.2	576	–	–
3	576	–	–
3.1	–	8x8	–
3.2	–	8x8	–
4	–	8x8	–
4.1	–	8x8	–
4.2	–	8x8	1
5	–	8x8	1
5.1	–	8x8	1

А.3.4 Влияние границ уровня на частоту кадров (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Таблица А-6 – Максимальная частота кадров (кадров в секунду) для некоторых примеров размеров кадров

Уровень:					1	1b	1,1	1,2	1,3	2	2,1
Макс. размер кадра (макроблоков):					99	99	396	396	396	396	792
Макс. макроблоков/с:					1 485	1 485	3 000	6 000	11 880	11 880	19 800
Макс. размер кадра (образцов):					25 344	25 344	101 376	101 376	101 376	101 376	202 752
Макс. образцов/с:					380 160	380 160	768 000	1 536 000	3 041 280	3 041 280	5 068 800
Формат	Ширина яркости	Высота яркости	Всего МБ	Образцов яркости							
SQCIF	128	96	48	12 288	30,9	30,9	62,5	125,0	172,0	172,0	172,0
QCIF	176	144	99	25 344	15,0	15,0	30,3	60,6	120,0	120,0	172,0
QVGA	320	240	300	76 800	–	–	10,0	20,0	39,6	39,6	66,0
525 SIF	352	240	330	84 480	–	–	9,1	18,2	36,0	36,0	60,0
CIF	352	288	396	101 376	–	–	7,6	15,2	30,0	30,0	50,0
525 HHR	352	480	660	168 960	–	–	–	–	–	–	30,0
625 HHR	352	576	792	202 752	–	–	–	–	–	–	25,0
VGA	640	480	1 200	307 200	–	–	–	–	–	–	–
525 4SIF	704	480	1 320	337 920	–	–	–	–	–	–	–
525 SD	720	480	1 350	345 600	–	–	–	–	–	–	–
4CIF	704	576	1 584	405 504	–	–	–	–	–	–	–
625 SD	720	576	1 620	414 720	–	–	–	–	–	–	–
SVGA	800	600	1 900	486 400	–	–	–	–	–	–	–
XGA	1 024	768	3 072	786 432	–	–	–	–	–	–	–
720p HD	1 280	720	3 600	921 600	–	–	–	–	–	–	–
4VGA	1 280	960	4 800	1 228 800	–	–	–	–	–	–	–
SXGA	1 280	1 024	5 120	1 310 720	–	–	–	–	–	–	–
525 16SIF	1 408	960	5 280	1 351 680	–	–	–	–	–	–	–
16CIF	1 408	1 152	6 336	1 622 016	–	–	–	–	–	–	–
4SVGA	1 600	1 200	7 500	1 920 000	–	–	–	–	–	–	–
1080 HD	1 920	1 088	8 160	2 088 960	–	–	–	–	–	–	–
2Kx1K	2 048	1 024	8 192	2 097 152	–	–	–	–	–	–	–
2Kx1080	2 048	1 088	8 704	2 228 224	–	–	–	–	–	–	–
4XGA	2 048	1 536	12 288	3 145 728	–	–	–	–	–	–	–
16VGA	2 560	1 920	19 200	4 915 200	–	–	–	–	–	–	–
3616x1536 (2.35:1)	3 616	1 536	21 696	5 554 176	–	–	–	–	–	–	–
3672x1536 (2.39:1)	3 680	1 536	22 080	5 652 480	–	–	–	–	–	–	–
4Kx2K	4 096	2 048	32 768	8 388 608	–	–	–	–	–	–	–
4096x2304 (16:9)	4 096	2 304	36 864	9 437 184	–	–	–	–	–	–	–

**Таблица А-6 (продолжение) – Максимальная частота кадров (кадров в секунду)
для некоторых примеров размеров кадров**

Уровень:					2,2	3	3,1	3,2	4	4,1	4,2
Макс. размер кадра (макроблоков):					1 620	1 620	3 600	5 120	8 192	8 192	8 704
Макс. макроблоков/с:					20 250	40 500	108 000	216 000	245 760	245 760	522 240
Макс. размер кадра (образцов):					414 720	414 720	921 600	1 310 720	2 097 152	2 097 152	2 228 224
Макс. образцов/с:					5 184 000	10 368 000	27 648 000	55 296 000	62 914 560	62 914 560	133 693 440
Формат	Ширина яркости	Высота яркости	Всего МБ	Образцов яркости							
SQCIF	128	96	48	12 288	172,0	172,0	172,0	172,0	172,0	172,0	172,0
QCIF	176	144	99	25 344	172,0	172,0	172,0	172,0	172,0	172,0	172,0
QVGA	320	240	300	76 800	67,5	135,0	172,0	172,0	172,0	172,0	172,0
525 SIF	352	240	330	84 480	61,4	122,7	172,0	172,0	172,0	172,0	172,0
CIF	352	288	396	101 376	51,1	102,3	172,0	172,0	172,0	172,0	172,0
525 HHR	352	480	660	168 960	30,7	61,4	163,6	172,0	172,0	172,0	172,0
625 HHR	352	576	792	202 752	25,6	51,1	136,4	172,0	172,0	172,0	172,0
VGA	640	480	1 200	307 200	16,9	33,8	90,0	172,0	172,0	172,0	172,0
525 4SIF	704	480	1 320	337 920	15,3	30,7	81,8	163,6	172,0	172,0	172,0
525 SD	720	480	1 350	345 600	15,0	30,0	80,0	160,0	172,0	172,0	172,0
4CIF	704	576	1 584	405 504	12,8	25,6	68,2	136,4	155,2	155,2	172,0
625 SD	720	576	1 620	414 720	12,5	25,0	66,7	133,3	151,7	151,7	172,0
SVGA	800	600	1 900	486 400	–	–	56,8	113,7	129,3	129,3	172,0
XGA	1 024	768	3 072	786 432	–	–	35,2	70,3	80,0	80,0	172,0
720p HD	1 280	720	3 600	921 600	–	–	30,0	60,0	68,3	68,3	145,1
4VGA	1 280	960	4 800	1 228 800	–	–	–	45,0	51,2	51,2	108,8
SXGA	1 280	1 024	5 120	1 310 720	–	–	–	42,2	48,0	48,0	102,0
525 16SIF	1 408	960	5 280	1 351 680	–	–	–	–	46,5	46,5	98,9
16CIF	1 408	1 152	6 336	1 622 016	–	–	–	–	38,8	38,8	82,4
4SVGA	1 600	1 200	7 500	1 920 000	–	–	–	–	32,8	32,8	69,6
1080 HD	1 920	1 088	8 160	2 088 960	–	–	–	–	30,1	30,1	64,0
2Kx1K	2 048	1 024	8 192	2 097 152	–	–	–	–	30,0	30,0	63,8
2Kx1080	2 048	1 088	8 704	2 228 224	–	–	–	–	–	–	60,0
4XGA	2 048	1 536	12 288	3 145 728	–	–	–	–	–	–	–
16VGA	2 560	1 920	19 200	4 915 200	–	–	–	–	–	–	–
3616x1536 (2.35:1)	3 616	1 536	21 696	5 554 176	–	–	–	–	–	–	–
3672x1536 (2.39:1)	3 680	1 536	22 080	5 652 480	–	–	–	–	–	–	–
4Kx2K	4 096	2 048	32 768	8 388 608	–	–	–	–	–	–	–
4096x2304 (16:9)	4 096	2 304	36 864	9 437 184	–	–	–	–	–	–	–

**Таблица А-6 (окончание) – Максимальная частота кадров (кадров в секунду)
для некоторых примеров размеров кадров**

Уровень:					5	5.1
Макс. размер кадра (макроблоков):					22 080	36 864
Макс. макроблоков/с:					589 824	983 040
Макс. размер кадра (образцов):					5 652 480	9 437 184
Макс. образцов/с:					150 994 944	251 658 240
Формат	Ширина яркости	Высота яркости	Всего МБ	Образцов яркости		
SQCIF	128	96	48	12 288	172,0	172,0
QCIF	176	144	99	25 344	172,0	172,0
QVGA	320	240	300	76 800	172,0	172,0
525 SIF	352	240	330	84 480	172,0	172,0
CIF	352	288	396	101 376	172,0	172,0
525 HHR	352	480	660	168 960	172,0	172,0
625 HHR	352	576	792	202 752	172,0	172,0
VGA	640	480	1 200	307 200	172,0	172,0
525 4SIF	704	480	1 320	337 920	172,0	172,0
525 SD	720	480	1 350	345 600	172,0	172,0
4CIF	704	576	1 584	405 504	172,0	172,0
625 SD	720	576	1 620	414 720	172,0	172,0
SVGA	800	600	1 900	486 400	172,0	172,0
XGA	1 024	768	3 072	786 432	172,0	172,0
720p HD	1 280	720	3 600	921 600	163,8	172,0
4VGA	1 280	960	4 800	1 228 800	122,9	172,0
SXGA	1 280	1 024	5 120	1 310 720	115,2	172,0
525 16SIF	1 408	960	5 280	1 351 680	111,7	172,0
16CIF	1 408	1 152	6 336	1 622 016	93,1	155,2
4SVGA	1 600	1 200	7 500	1 920 000	78,6	131,1
1080 HD	1 920	1 088	8 160	2 088 960	72,3	120,5
2Kx1K	2 048	1 024	8 192	2 097 152	72,0	120,0
2Kx1080	2 048	1 088	8 704	2 228 224	67,8	112,9
4XGA	2 048	1 536	12 288	3 145 728	48,0	80,0
16VGA	2 560	1 920	19 200	4 915 200	30,7	51,2
3616x1536 (2.35:1)	3 616	1 536	21 696	5 554 176	27,2	45,3
3672x1536 (2.39:1)	3 680	1 536	22 080	5 652 480	26,7	44,5
4Kx2K	4 096	2 048	32 768	8 388 608	–	30,0
4096x2304 (16:9)	4 096	2 304	36 864	9 437 184	–	26,7

Необходимо отметить следующее.

- Настоящая Рекомендация | Международный стандарт представляет спецификацию с переменным размером кадров. Конкретный размер кадра в таблице А-6 является просто пояснительным примером.
- Указанное в таблице А-6 значение "525" относится к типовому использованию аналогового сканирования 525 линий (из которых примерно 480 линий содержат область видимого изображения), а "625" относится к использованию аналогового сканирования 625 линий (из которых примерно 576 линий содержат область видимого изображения).
- XGA также известно как (aka) XVGA, 4SVGA aka UXGA, 16XGA aka 4Kx3K, CIF aka 625 SIF, 625 HHR aka 2CIF aka half 625 D-1, aka half 625 ITU-R BT.601, 525 SD aka 525 D-1 aka 525 ITU-R BT.601, 625 SD aka 625 D-1 aka 625 ITU-R BT.601.
- Приведенные значения частоты кадров справедливы для режимов последовательного сканирования. Значения частоты кадров также справедливы для чередующегося кодирования видеосигнала для тех случаев, когда высота кадра кратна 32.

Приложение В

Формат потока байтов

(Данное Приложение является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

В этом Приложении определены синтаксис и семантика формата потока байтов, определенного для использования приложениями, которые доставляют некоторые или все потоки блоков NAL в виде упорядоченного потока байтов или битов, внутри которых расположены границы блоков NAL, необходимые для возможности различить шаблоны данных, как, например, в системах согласно Рекомендации МСЭ-Т Н.222.0 | ИСО/МЭК 13818-1 или Рекомендации МСЭ-Т Н.320. Для доставки с ориентацией на биты порядок битов в формате потока байтов определен при старте битом MSB первого байта, предшествующего LSB первого байта, за которым следует MSB второго байта и т. д.

Формат потока байтов состоит из последовательности потока байтов со структурой синтаксиса блоков NAL. Каждый поток байтов со структурой синтаксиса блоков NAL содержит один префикс кода старта, за которым следует одна структура синтаксиса `nal_unit(NumBytesInNALunit)`. Поток может также содержать (при некоторых обстоятельствах) дополнительный элемент синтаксиса `zero_byte`, а также один или более дополнительных элементов синтаксиса `trailing_zero_8bits`. Если это – первый поток байтов блоков NAL в потоке битов, он может также содержать один или более элементов синтаксиса `leading_zero_8bits`.

В.1 Синтаксис и семантика блоков NAL потока байтов

В.1.1 Синтаксис блоков NAL потока байтов

byte_stream_nal_unit(NumBytesInNALunit) {	С	Дескриптор
while(next_bits(24) != 0x000001 && next_bits(32) != 0x00000001)		
leading_zero_8bits /* равно 0x00 */		f(8)
if(next_bits(24) != 0x000001)		
zero_byte /* равно 0x00 */		f(8)
start_code_prefix_one_3bytes /* равно 0x000001 */		f(24)
nal_unit(NumBytesInNALunit)		
while(more_data_in_byte_stream() && next_bits(24) != 0x000001 && next_bits(32) != 0x00000001)		
trailing_zero_8bits /* равно 0x00 */		f(8)
}		

В.1.2 Семантика блоков NAL потока байтов

Порядок блоков NAL в потоке байтов должен следовать порядку декодирования блоков NAL, находящихся в потоке байтов (см. п. 7.4.1.2). Содержание каждого блока NAL потока байтов объединяют с тем же блоком доступа, что и блок NAL, который содержится среди блоков NAL потока байтов (см. п. 7.4.1.2.3).

leading_zero_8bits – байт, равный 0x00.

ПРИМЕЧАНИЕ. – Элемент синтаксиса `leading_zero_8bits` может быть представлен только в первом блоке NAL потока байтов из потока битов, так как (как показано на диаграмме синтаксиса в п. В.1.1) любые байты, равные 0x00, которые следуют за структурой синтаксиса блока NAL и предшествуют 4-байтовой последовательности 0x00000001 (что можно интерпретировать как `zero_byte` с последующим `start_code_prefix_one_3bytes`), должны рассматриваться как элементы синтаксиса `trailing_zero_8bits`, составляющие часть предшествующих блоков NAL потока байтов.

zero_byte – единичный байт, равный 0x00.

Если любое из следующих условий выполнено, то должен быть представлен элемент синтаксиса `zero_byte`:

- значение `nal_unit_type` в `nal_unit()` равно 7 (набор параметров последовательности) или 8 (набор параметров изображения);
- структура синтаксиса блока NAL в потоке байтов содержит первый блок NAL из блока доступа в порядке декодирования, как это определено в п. 7.4.1.2.3.

start_code_prefix_one_3bytes – последовательность из 3 байтов фиксированной величины, равной 0x000001. Этот элемент синтаксиса называют префиксом кода старта (запуска).

trailing_zero_8bits – байт, равный 0x00.

В.2 Процесс декодирования блоков NAL потока байтов

Вход в этот процесс состоит из упорядоченного потока байтов, содержащего последовательность структуры синтаксиса блоков NAL потока байтов.

Выход этого процесса состоит из последовательности структур синтаксиса блоков NAL.

В начале процесса декодирования декодер инициализирует свою текущую позицию в потоке байтов от начала этого потока. Далее декодер извлекает и отбрасывает каждый элемент синтаксиса `leading_zero_8bits` (если такой присутствует), продвигая текущую позицию в потоке байтов вперед каждый раз на один байт, пока текущая позиция в потоке байтов не станет такой, что следующие четыре байта в потоке битов сформируют 4-байтовую последовательность `0x00000001`.

Далее декодер повторно выполняет следующий пошаговый процесс, чтобы извлечь и декодировать каждую структуру синтаксиса блока NAL в потоке байтов, пока не будет достигнут конец (что устанавливается с помощью не определяемых здесь средств) и декодирован последний блок NAL в потоке байтов.

1. Если следующие 4 байта в потоке битов формируют 4-байтовую последовательность `0x00000001`, то следующий байт в потоке байтов (который является элементом синтаксиса `zero_byte`) извлекают и отбрасывают, а текущую позицию в потоке байтов устанавливают равной позиции байта, следующего за отброшенным.
2. Следующую 3-байтовую последовательность в потоке байтов (которой является `start_code_prefix_one_3bytes`) извлекают и отбрасывают, а текущую позицию в потоке байтов устанавливают равной позиции байта, следующего за 3-байтовой последовательностью.
3. Значение `NumBytesInNALunit` устанавливают равным числу байтов, которое начинается с байта на текущей позиции в потоке байтов и продолжается до (и включая) последний байт, который предшествует положению с любыми из следующих условий.
 - a. Последующая байт-ориентированная 3-байтовая последовательность равна `0x000000`; или
 - b. последующая байт-ориентированная 3-байтовая последовательность равна `0x000001`; или
 - c. конец потока байтов, что было установлено не определяемыми здесь средствами.
4. `NumBytesInNALunit` байтов удалены из потока битов, а текущая позиция в потоке байтов продвинута на `NumBytesInNALunit` байтов. Эту последовательность байтов `nal_unit(NumBytesInNALunit)` декодируют, используя процесс декодирования блока NAL.
5. Если текущая позиция в потоке байтов не является концом потока байтов (что устанавливают не определяемые здесь средства) и следующие байты в потоке байтов не начинаются с 3-байтовой последовательности, равной `0x000001`, а последующие байты в потоке байтов не начинаются с 4-байтовой последовательности, равной `0x00000001`, то декодер извлекает и отбрасывает каждый элемент синтаксиса `trailing_zero_8bits`, продвигающий текущую позицию в потоке байтов вперед каждый раз на один байт, до тех пор, пока текущая позиция в потоке байтов не станет такой, что следующие байты в потоке байтов не сформируют 4-байтовую последовательность `0x00000001`, или пока не появится конец потока байтов (что устанавливают не определяемые здесь средства).

В.3 Восстановление декодером выравнивания байтов (для информации)

Данный раздел не является неотъемлемой частью настоящей Рекомендации | Международного стандарта.

Многие приложения поставляют данные в декодер способом, который, по сути, является выровненным по байтам и поэтому не нуждается в описанной в этом пункте процедуре обнаружения ориентированного бита для выравнивания байтов.

Декодер называют выровненным по байтам с потоком битов, если декодер в состоянии определить, являются ли позиции данных в потоке битов выровненными по байтам. Если у декодера отсутствует выравнивание байтов относительно потока байтов кодера, то декодер может проверить входящий поток битов по бинарному шаблону `'00000000 00000000 00000000 00000001'` (31 последовательный бит, равный 0, за которым следует бит, равный 1). Бит, непосредственно следующий за этим шаблоном, – это первый бит выровненного байта, за которым следует префикс кода старта. После проверки этого шаблона декодер будет выровненным по байтам с кодером и будет занимать позицию в начале блока NAL в потоке байтов.

Как только произойдет выравнивание по байтам с кодером, декодер может проверить входящий поток байтов на следующие 3-байтовые последовательности: `0x000001` и `0x000003`.

Если 3-байтовая последовательность `0x000001` обнаружена, это и есть префикс кода старта.

Если обнаружена 3-байтовая последовательность `0x000003`, третий байт (`0x03`) – `emulation_prevention_three_byte` должен быть отброшен, как это определено в п. 7.4.1.

Если в потоке битов обнаруживается ошибка синтаксиса (например, ненулевое значение `forbidden_zero_bit` или одна из 3-байтовых или 4-байтовых последовательностей, запрещенных в п. 7.4.1), декодер может рассматривать обнаруженное условие как указание на вероятную потерю выравнивания байтов и может отбрасывать все данных потоков битов до обнаружения выравнивания байтов в более дальней позиции в потоке битов, как описано в данном пункте.

Приложение С

Гипотетический контрольный декодер

(Данное Приложение является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Это Приложение определяет гипотетический контрольный декодер (HRD) и его использование для контроля соответствия потока битов и декодера.

Под действие HRD попадают два типа потоков битов для проверки на соответствие настоящей Рекомендации | Международному стандарту. Первый тип потока битов, называемый потоком битов Типа I, это поток блоков NAL, содержащий только блоки NAL VCL и блоки NAL данных заполнения для всех блоков доступа в потоке битов. Второй тип, называемый потоком битов Типа II, содержит вдобавок к блокам NAL VCL и блокам NAL данных заполнения для всех блоков доступа в потоке битов, по крайней мере, одно из следующих:

- дополнительные не блоки NAL VCL и иные, чем блоки NAL данных заполнения;
- все элементы синтаксиса `leading_zero_8bits`, `zero_byte`, `start_code_prefix_one_3bytes` и `trailing_zero_8bits`, которые формируют поток блоков NAL из потока байтов (как определено в Приложении В).

На рисунке G-1 показаны точки соответствия типов потоков битов, которые контролирует HRD.

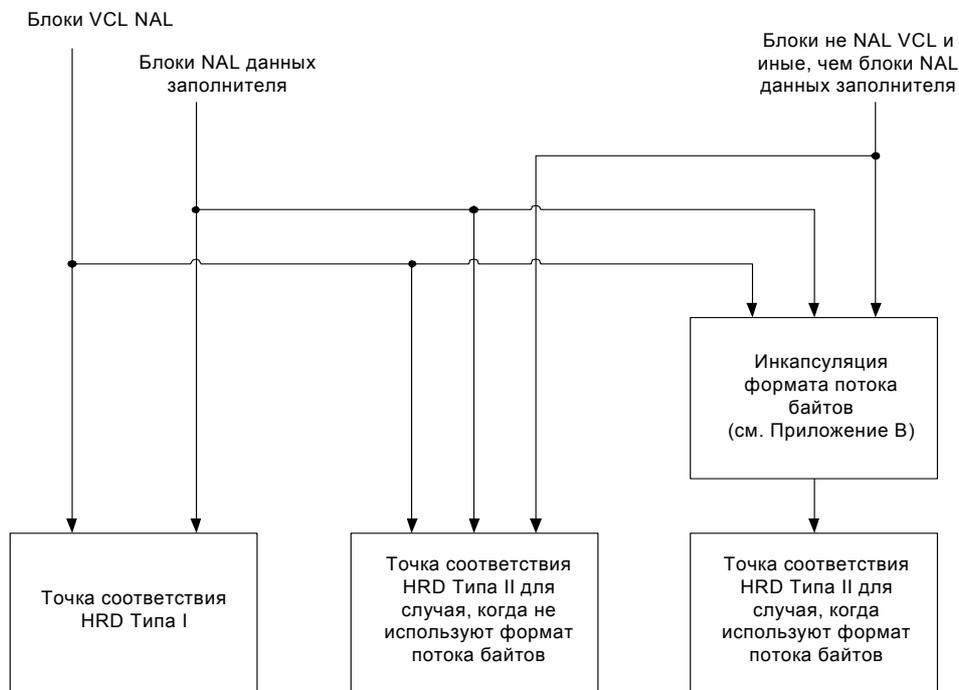


Рисунок С-1 – Структура потоков байтов и потоков блоков NAL для проверки соответствия с помощью HRD

Элементы синтаксиса не блоков NAL VCL (или их значения для некоторых элементов синтаксиса), которые требуются для HRD, рассмотрены в пунктах семантики в разделе 7 и в Приложениях D и E.

Используют два типа набора параметров HRD. Наборы параметров HRD сообщают с помощью информации об используемости изображения (как описано в пп. E.1 и E.2), к какой части структуры синтаксиса относится последовательность набора параметров.

Чтобы проверить соответствие потока битов с помощью HRD, все наборы параметров последовательности, а также наборы параметров изображения относят к блокам NAL VCL, а соответствующий период буферизации и сообщения SEI о синхронизации изображения должны быть переданы в HRD своевременно либо в потоке битов (не блоками NAL VCL), либо другими средствами, не описанными в настоящей Рекомендации | Международном стандарте.

В Приложениях С, D и E спецификация для "присутствия" не блоков NAL VCL также выполняется, если эти блоки NAL (или некоторые из них) поступают в декодеры (или в HRD) с помощью других средств, не описанных в настоящей Рекомендации | Международном стандарте. Поэтому при подсчете битов учитывают только те биты, которые действительно присутствуют в потоке битов.

ПРИМЕЧАНИЕ 1. – В качестве примера синхронизацию не блока NAL VCL (которую передают средствами иными, чем для блоков NAL, присутствующими в потоке битов) можно выполнить, указав на два пункта в потоке битов, между которыми не блок NAL VCL мог бы присутствовать, если бы кодер решил передавать его в потоке битов.

Если содержание не блока NAL VCL передают приложению некоторыми средствами иными, чем присутствием в потоке битов, то представление содержания не блока NAL VCL не требует использования того же синтаксиса, который описан в этом Приложении.

ПРИМЕЧАНИЕ 2. – Если информация HRD содержится в потоке битов, то возможно проверить соответствие потока битов требованиям этого пункта, основываясь исключительно на информации, которая имеется в потоке битов. Если информация HRD не содержится в потоке битов, как в случае для всех "автономных" потоков битов Типа I, соответствие можно проверить, если данные HRD поставлены какими-то другими средствами, не рассмотренными в настоящей Рекомендации | Международном стандарте.

HRD состоит из буфера кодированного изображения (CPB), процесса мгновенного декодирования, буфера декодированного изображения (DPB) и разделения на кадры или поля (кадрирование) на выходе, как показано на рисунке С-2.

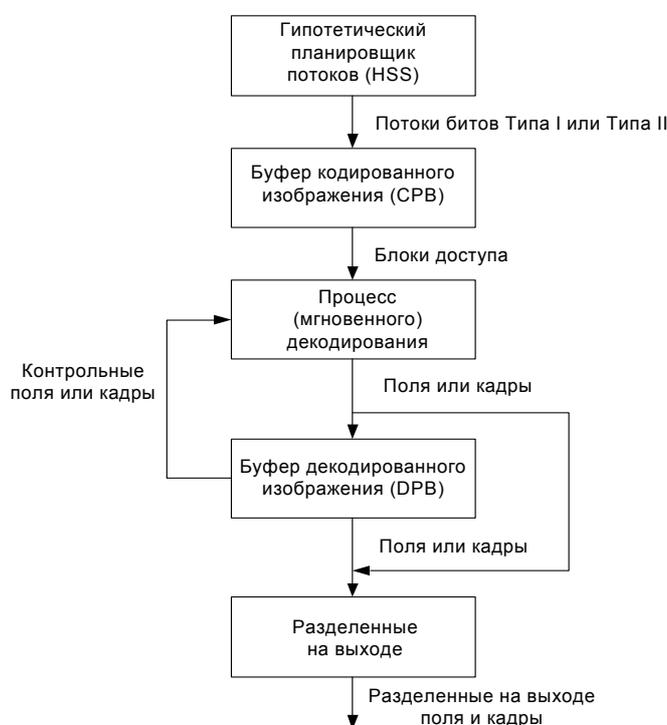


Рисунок С-2 – Модель буфера HRD

Размером CPB (числом битов) является $CpbSize[SchedSelIdx]$. Размером DPB (числом буферов кадров) является $Max(1, max_dec_frame_buffering)$.

Декодер HRD работает следующим образом. Данные, объединенные с блоками доступа, которые поступают в CPB согласно определенному расписанию, расставляют с помощью HSS. Данные, объединенные с каждым блоком доступа, удаляют и мгновенно декодируют процессом мгновенного декодирования в моменты удаления из CPB. Каждое декодированное изображение помещают в DPB в моменты удаления из CPB, если это не выход CPB в момент удаления и если это неконтрольное изображение. Если изображение размещено в DPB, его позднее удаляют из DPB во время выхода DPB или во время, при котором его помечают как "не используется для контроля".

Работа CPB описана в п. С.1. Работа мгновенного декодера описана в разделах 8 и 9. Работа DPB описана в п. С.2. Выход операции кадрирования описан в п. С.2.2.

Информация HSS и HRD, которая касается числа перечисленных расписаний доставки и их связи со скоростями битов и размерами буфера, описана в пп. E.1.1, E.1.2, E.2.1 и E.2.2. HRD инициализируют, как это определено сообщением SEI о периоде буферизации в пп. D.1.1 и D.2.1. Моменты удаления блоков доступа из CPB и моменты удаления на выходе DPB указаны в сообщении SEI о синхронизации изображения, как это определено в пп. D.1.2 и

D.2.2. Вся информация о распределении во времени, которая касается специального блока доступа, должна поступать до времени удаления из СРВ блока доступа.

HRD используют для контроля соответствия потоков битов и декодеров, как это определено в пп. С.3 и С.4, соответственно.

ПРИМЕЧАНИЕ 3. – Хотя соответствие гарантировано в предположении, что все значения частоты кадров и тактовых импульсов, использованных для генерации потока битов, точно соответствуют сигналам в потоке битов, в реальной системе каждое из них может отклоняться от этих сигналов или установленных значений.

Вся арифметика в этом Приложении выполнена над реальными значениями, с тем чтобы избежать размножения ошибок округления. Например, число битов в СРВ непосредственно перед и после удаления блока доступа не обязательно является целым числом.

Переменную t_c называют тактом системных часов (clock tick) и находят следующим образом:

$$t_c = \text{num_units_in_tick} \div \text{time_scale}. \quad (\text{C-1})$$

Следующие пункты определены для выражения ограничений в этом Приложении.

- Положим, что блок доступа n – это n -й блок доступа в порядке декодирования с первым блоком доступа, который является блоком доступа 0.
- Положим, что изображение n – первично кодированное изображение или первично декодированное изображение блока доступа n .

C.1 Работа буфера кодированного изображения (СРВ)

Спецификация в этом пункте применима независимо от каждого набора параметров СРВ и от точек соответствия для обоих типов, Типа I и Типа II, показанных на рисунке С-1.

C.1.1 Синхронизация поступления потока битов

HRD можно инициализировать в любой из периодов сообщения SEI о периоде буферизации. Перед инициализацией СРВ пусто.

ПРИМЕЧАНИЕ. – После инициализации HRD не инициализируют снова при последующем сообщении SEI о периоде буферизации.

Каждый блок доступа считают блоком доступа n , где число n идентифицирует конкретный блок доступа. Блок доступа, связанный с сообщением SEI о периоде буферизации, при котором происходит инициализация СРВ, считают блоком доступа 0. Это значение возрастает на 1 для каждого последующего блока доступа в порядке декодирования.

Время, когда первый бит блока доступа n начинает поступать в СРВ, считают начальным временем поступления $t_{ai}(n)$.

Начальное время поступления блоков доступа находят следующим образом.

- Если блок доступа – это блок доступа 0, $t_{ai}(0) = 0$.
- Иначе (блок доступа – это блок доступа n с $n > 0$) применяют следующее.
 - Если $\text{cbr_flag}[\text{SchedSelIdx}]$ равно 1, начальное время поступления блока доступа n равно конечному времени поступления (которое находят ниже) блока доступа $n - 1$:

$$t_{ai}(n) = t_{af}(n - 1). \quad (\text{C-2})$$

- Иначе ($\text{cbr_flag}[\text{SchedSelIdx}]$ равно 0) начальное время поступления блока доступа n находят как:

$$t_{ai}(n) = \text{Max}(t_{af}(n - 1), t_{ai,earliest}(n)), \quad (\text{C-3})$$

где $t_{ai,earliest}(n)$ находят следующим образом:

- если блок доступа n – не первый блок доступа последующего периода буферизации, то $t_{ai,earliest}(n)$ находят как:

$$t_{ai,earliest}(n) = t_{r,n}(n) - (\text{initial_cpb_removal_delay}[\text{SchedSelIdx}] + \text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]) \div 90000, \quad (\text{C-4})$$

где $t_{r,n}(n)$ – номинальное время удаления блока доступа n из СРВ, как это определено в п. С.1.2, а $\text{initial_cpb_removal_delay}[\text{SchedSelIdx}]$ и $\text{initial_cpb_removal_delay_offset}[\text{SchedSelIdx}]$ определены в предыдущем сообщении SEI о периоде буферизации;

- иначе (блок доступа n – не первый блок доступа в последующем периоде буферизации) $t_{ai,earliest}(n)$ находят как:

$$t_{ai,earliest}(n) = t_{r,n}(n) - (\text{initial_cpb_removal_delay}[\text{SchedSelIdx}] \div 90000), \quad (\text{C-5})$$

со значением $\text{initial_cpb_removal_delay}[\text{SchedSelIdx}]$, определенным в сообщении SEI о периоде буферизации, связанном с блоком доступа n .

Конечное время поступления блока доступа n находят как:

$$t_{af}(n) = t_{ai}(n) + b(n) \div \text{BitRate}[\text{SchedSelIdx}], \quad (\text{C-6})$$

где $b(n)$ – размер в битах блока доступа n , считая биты блоков NAL VCL и блоков NAL с данными заполнения для точек соответствия Типу I или все биты потока битов Типа II для точек соответствия Типу II, где точки соответствия Типу I и Типу II показаны на рисунке С-1.

Значения SchedSelIdx , $\text{BitRate}[\text{SchedSelIdx}]$ и $\text{CpbSize}[\text{SchedSelIdx}]$ ограничены следующим образом.

- Если блок доступа n и блок доступа $n - 1$ – части разных кодированных видеопоследовательностей, а содержание действующих наборов параметров последовательности двух кодированных видеопоследовательностей различаются, то HSS выбирает значение SchedSelIdx1 из SchedSelIdx среди значений SchedSelIdx , обеспечивая для кодированной видеопоследовательности, содержащей блок доступа n , эти результаты в значениях $\text{BitRate}[\text{SchedSelIdx1}]$ или $\text{CpbSize}[\text{SchedSelIdx1}]$ для второй из двух кодированных видеопоследовательностей (которая содержит блок доступа n). Значения $\text{BitRate}[\text{SchedSelIdx1}]$ или $\text{CpbSize}[\text{SchedSelIdx1}]$ могут отличаться от значений $\text{BitRate}[\text{SchedSelIdx0}]$ или $\text{CpbSize}[\text{SchedSelIdx0}]$ для значения SchedSelIdx0 из SchedSelIdx , которое было использовано в кодированной видеопоследовательности с блоком доступа $n - 1$.
- Иначе HSS продолжает работу с предыдущими значениями SchedSelIdx , $\text{BitRate}[\text{SchedSelIdx}]$ и $\text{CpbSize}[\text{SchedSelIdx}]$.

Если HSS выбирает значения $\text{BitRate}[\text{SchedSelIdx}]$ или $\text{CpbSize}[\text{SchedSelIdx}]$, которые отличаются от первоначальных значений блока доступа, применяют следующее.

- Переменная $\text{BitRate}[\text{SchedSelIdx}]$ вступает в действие в момент $t_{ai}(n)$.
- Переменная $\text{CpbSize}[\text{SchedSelIdx}]$ вступает в действие следующим образом:
 - если новое значение $\text{CpbSize}[\text{SchedSelIdx}]$ превосходит прежний размер CPB, оно вступает в действие в момент $t_{ai}(n)$;
 - иначе новое значение $\text{CpbSize}[\text{SchedSelIdx}]$ вступает в действие в момент $t_f(n)$.

С.1.2 Синхронизация удаления кодированного изображения

Для блока доступа 0 номинальное время удаления блока доступа из CPB определено как:

$$t_{r,n}(0) = \text{initial_cpb_removal_delay}[\text{SchedSelIdx}] \div 90000. \quad (\text{C-7})$$

Для первого блока доступа в период буферизации, при котором не происходит инициализации HRD, номинальное время удаления блока доступа из CPB определяют как:

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * \text{cpb_removal_delay}(n), \quad (\text{C-8})$$

где $t_{r,n}(n_b)$ – номинальное время удаления первого блока доступа предыдущего периода буферизации, а $\text{cpb_removal_delay}(n)$ – значение cpb_removal_delay , определенное в сообщении SEI о синхронизации изображения, связанного с блоком доступа n .

Если блок доступа n – первый блок доступа периода буферизации, то n_b устанавливают равным значению n во время удаления блока доступа n .

Номинальное время удаления $t_{r,n}(n)$ блока доступа n , который не является первым блоком доступа периода буферизации, задают как:

$$t_{r,n}(n) = t_{r,n}(n_b) + t_c * \text{cpb_removal_delay}(n), \quad (\text{C-9})$$

где $t_{r,n}(n_b)$ – номинальное время удаления первого блока доступа текущего периода буферизации, а $\text{cpb_removal_delay}(n)$ – значение cpb_removal_delay , определенное в сообщении SEI о синхронизации изображения, связанного с блоком доступа n .

Время удаления блока доступа n определено следующим образом.

- Если $low_delay_hrd_flag$ равно 0 или $t_{r,n}(n) \geq t_{af}(n)$, то время удаления блока доступа n определено как:

$$t_r(n) = t_{r,n}(n). \quad (C-10)$$

- Иначе ($low_delay_hrd_flag$ равно 1 и $t_{r,n}(n) < t_{af}(n)$) время удаления блока доступа n определено как:

$$t_r(n) = t_{r,n}(n) + t_c * Ceil((t_{af}(n) - t_{r,n}(n)) \div t_c). \quad (C-11)$$

ПРИМЕЧАНИЕ. – Последний случай указывает, что размер блока доступа n , $b(n)$ настолько велик, что его невозможно удалить за номинальное время удаления.

C.2 Работа буфера декодированного изображения (DPB)

Буфер декодированного изображения содержит буферы кадров. Каждый из буферов кадров может содержать декодированный кадр, декодированную пару дополнительных полей или единственное (непарное) декодированное поле, которые помечены как "используется для контроля" (контрольных изображений) или удерживаются для будущего выхода (измененного порядка или задержанного изображения). Перед инициализацией DPB пусто (заполнение DPB устанавливается на нуль). Все последующие разделы этого пункта имеют дело со временем $t_r(n)$ и с перечисленными последовательностями.

C.2.1 Декодирование промежутков в $frame_num$ и хранение "несуществующих" кадров

В случае применения промежутки в значении $frame_num$ обнаруживаются процессом декодирования, а генерированные кадры отмечают и вводят в DPB, как это определено ниже.

Промежутки в $frame_num$ обнаруживаются процессом декодирования, а генерированные кадры отмечают, как это определено в п. 8.2.5.2.

После отметки каждого сгенерированного кадра каждое изображение m , отмеченное в процессе отметки в "раздвижном окне" как "не используется для контроля", удаляют из DPB, если буфер также отмечен как "несуществующий" или если выходное время DPB меньше чем или равно времени удаления CPB текущего изображения n , т. е. $t_{o,dpb}(m) \leq t_r(n)$. Если кадр или последнее поле в кадре буфера удалено из DPB, то заполнение DPB определяют единицей. "Несуществующий" сгенерированный кадр вводят в DPB, а заполнение DPB увеличивается на единицу.

C.2.2 Декодирование и выход изображения

Изображение n декодируют, а время выхода DPB $t_{o,dpb}(n)$ находят как:

$$t_{o,dpb}(n) = t_r(n) + t_c * dpb_output_delay(n). \quad (C-12)$$

Выход текущего изображения определен следующим образом.

- Если $t_{o,dpb}(n) = t_r(n)$, то текущее изображение – это выход.

ПРИМЕЧАНИЕ. – Если текущее изображение – это контрольное изображение, оно будет сохранено в DPB.

- Иначе ($t_{o,dpb}(n) > t_r(n)$) текущее изображение – это выход с задержкой, и изображение будет сохранено в DPB (как это определено в п. C.2.4). Выход этого изображения происходит во время $t_{o,dpb}(n)$, если только не будет указано, что это – не выход с помощью декодирования или вмешательства команды $no_output_of_prior_pics_flag$, равной 1, за время, предшествующее $t_{o,dpb}(n)$.

Выход изображения должен быть разделен на кадры (или поля), используя прямоугольник кадрирования, определенный набором параметров для этой последовательности.

Если изображение n – это изображение, которое является выходом, но не последнее изображение в потоке битов, который представляет выход, то значение $\Delta t_{o,dpb}(n)$ определяют как:

$$\Delta t_{o,dpb}(n) = t_{o,dpb}(n_n) - t_{o,dpb}(n), \quad (C-13)$$

где n_n указывает на изображение, которое следует за изображением n в порядке выхода.

Декодированное изображение временно сохраняют (но не в DPB).

C.2.3 Удаление изображения из DPB перед возможным введением текущего изображения

Удаление изображения из DPB перед возможным введением текущего изображения происходит следующим образом.

- Если декодированное изображение – это IDR изображение, то применяют следующее.

- Все контрольные изображения в DPB помечают как "не используется для контроля", как это определено в п. 8.2.5.1.
- Если IDR изображение – это не первое декодированное IDR изображение, а значения PicWidthInMbs, или FrameHeightInMbs, или max_dec_frame_buffering, найденные из действующего набора параметров последовательности, отличаются от значений PicWidthInMbs, или FrameHeightInMbs, или max_dec_frame_buffering, найденных из набора параметров последовательности, которая была действующей соответственно для предыдущей последовательности, то в HRD значение no_output_of_prior_pics_flag принимают равным 1 независимо от действительного значения no_output_of_prior_pics_flag.

ПРИМЕЧАНИЕ. – При использовании декодера следует попытаться управлять изменениями размеров кадра или DPB, но более осторожно, чем в HRD по отношению к изменениям в PicWidthInMbs или FrameHeightInMbs.
- Если no_output_of_prior_pics_flag равно 1 или предполагают равным 1, то все буферы кадров в DPB опустошают без выдачи на выход изображений, которые там содержались, а заполнение DPB устанавливают на 0.
- Иначе (декодированное изображение – это не IDR изображение) применяют следующее.
 - Если заголовок секции текущего изображения включает memory_management_control_operation, равное 5, то все контрольные изображения в DPB помечают как "не используется для контроля".
 - Иначе (заголовок секции текущего изображения не включает memory_management_control_operation, равное 5) активируют процесс разметки декодированного контрольного изображения, определенный в п. 8.2.5.

Все изображения m в DPB, для которых все нижеследующие условия являются истиной, удаляют из DPB:

- изображение m помечено как "не используется для контроля" или изображение m – это неконтрольное изображение. Если изображение – это контрольный кадр, то считают, что он помечен как "не используется для контроля", только если оба его поля уже помечены как "не используется для контроля";
- изображение m помечено как "несуществующее" или время выхода из DPB меньше чем или равно времени удаления из CPB текущего изображения n , т. е. $t_{o,dpb}(m) \leq t(n)$.

Если кадр или последнее поле в буфере кадров удалены из DPB, то заполнение DPB уменьшают на единицу.

С.2.4 Разметка и хранение текущего декодированного изображения

С.2.4.1 Разметка и хранение контрольного декодированного изображения в DPB

Если текущее изображение – это контрольное изображение, то его хранят в DPB следующим образом.

- Если текущее декодированное изображение является вторым полем (в порядке декодирования) дополнительной пары контрольных полей, а первое поле пары все еще находится в DPB, то текущее декодированное изображение хранят в том же буфере кадров, что и первое поле пары.
- Иначе текущее декодированное изображение хранят в пустом буфере кадров, а заполнение DPB возрастает на единицу.

С.2.4.2 Хранение неконтрольного изображения в DPB

Если текущее изображение – это неконтрольное изображение и для текущего изображения n имеет место $t_{o,dpb}(n) > t(n)$, то его хранят в DPB следующим образом.

- Если текущее декодированное изображение – это второе поле (в порядке декодирования) дополнительной пары неконтрольных полей, а первое поле пары все еще находится в DPB, текущее декодированное изображение хранят в том же буфере кадров, что и первое поле пары.
- Иначе текущее декодированное изображение хранят в пустом буфере кадров, а заполнение DPB возрастает на единицу.

С.3 Соответствие потока битов

Поток битов кодированных данных, соответствующих настоящей Рекомендации | Международному стандарту, должен выполнять следующие требования.

Поток битов создают вне рамок этого Приложения, согласно синтаксису, семантике и ограничениям, определенным в настоящей Рекомендации | Международном стандарте.

Поток битов проверяют с помощью HRD, как это определено ниже.

Для потоков битов Типа I число проверок выполняют $spb_cnt_minus1 + 1$ раз, где spb_cnt_minus1 – это либо элемент синтаксиса $hrd_parameters()$, следующий за $vcl_hrd_parameters_present_flag$, либо определенный другими средствами приложения, которые не рассмотрены в настоящей Рекомендации | Международном стандарте. Для каждой комбинации скорости битов и размера CPB, определенных значением $hrd_parameters()$, следующим за

vcl_hrd_parameters_present_flag, выполняют одну проверку. Каждая из этих проверок руководствуется точкой соответствия для Типа I, показанной на рисунке С-1.

Для потоков битов Типа II существует два набора проверок. Число проверок первого набора равно $cpb_cnt_minus1 + 1$, где cpb_cnt_minus1 – это либо элемент синтаксиса `hrd_parameters()`, следующий за `vcl_hrd_parameters_present_flag`, либо определенный другими средствами приложения, которые не рассмотрены в настоящей Рекомендации | Международном стандарте. Для каждой комбинации скорости битов и размера CPB выполняют одну проверку. Каждая из этих проверок руководствуется точкой соответствия для Типа I, показанной на рисунке С-1. Для этих проверок в качестве входной скорости битов и памяти CPB учитывают только блоки VCL и NAL с данными заполнения.

Число проверок второго набора для потока битов Типа II равно $cpb_cnt_minus1 + 1$, где cpb_cnt_minus1 – это либо элемент синтаксиса `hrd_parameters()`, следующий за `nal_hrd_parameters_present_flag`, либо определенный другими средствами приложения, которые не рассмотрены в настоящей Рекомендации | Международном стандарте. Для каждой комбинации скорости битов и размера CPB, определенной значением `hrd_parameters()`, следующим за `nal_hrd_parameters_present_flag`, выполняют одну проверку. Каждая из этих проверок руководствуется точкой соответствия для Типа II, показанной на рисунке С-1. Для этих проверок учитывают все блоки NAL (поток блоков NAL Типа II) или все байты (из потока байтов) в качестве входной скорости битов и памяти CPB.

ПРИМЕЧАНИЕ 1. – Параметры NAL HRD, установленные значением `SchedSelIdx` точки соответствия Типа II, показанной на рисунке С-1, достаточны также, чтобы установить соответствие VCL HRD для точки соответствия Типа I, показанной на рисунке С-1, для тех же значений `initial_cpb_removal_delay[SchedSelIdx]`, `BitRate[SchedSelIdx]` и `CpbSize[SchedSelIdx]` для случая VBR (`cbr_flag[SchedSelIdx]` равно 0). Так происходит, потому что поток данных в точке соответствия Типа I является подмножеством потока данных в точке соответствия Типа II, а также потому, что в случае VBR буферу CPB разрешено стать пустым и оставаться пустым до тех пор, пока по расписанию не начнет поступать следующее изображение. Например, если параметры NAL HRD предусмотрены для точки соответствия Типа II таким образом, что не только находятся в границах набора параметров для соответствия профиля NAL HRD подпункту j п. А.3.1 или подпункта i п. А.3.3 (в зависимости от используемого профиля), но также находятся в границах набора параметров для соответствия профиля VCL HRD подпункту i п. А.3.1 или подпункту h п. А.3.3 (в зависимости от используемого профиля), то соответствие VCL HRD для точки соответствия Типа I также уверенно попадет в границы подпункта i пункта А.3.1.

Для соответствующих потоков битов все следующие условия должны выполняться для каждой из проверок.

- Для каждого блока доступа n с $n > 0$, связанного с сообщением SEI о периоде буферизации, и с $\Delta t_{g,90}(n)$, определенным как:

$$\Delta t_{g,90}(n) = 90\,000 * (t_{r,n}(n) - t_{af}(n-1)). \quad (C-14)$$

Значение `initial_cpb_removal_delay[SchedSelIdx]` должно быть ограничено следующим образом.

- Если `cbr_flag[SchedSelIdx]` равно 0:

$$\text{initial_cpb_removal_delay[SchedSelIdx]} \leq \text{Ceil}(\Delta t_{g,90}(n)). \quad (C-15)$$

- Иначе (`cbr_flag[SchedSelIdx]` равно 1):

$$\text{Floor}(\Delta t_{g,90}(n)) \leq \text{initial_cpb_removal_delay[SchedSelIdx]} \leq \text{Ceil}(\Delta t_{g,90}(n)). \quad (C-16)$$

ПРИМЕЧАНИЕ 2. – Точное число битов в CPB во время удаления каждого изображения может зависеть от того, какой период буферизации в сообщении SEI был выбран для инициализации HRD. Кодеры должны принимать это во внимание, чтобы обеспечить соблюдение всех определенных ограничений, независимо от того, какой период буферизации в сообщении SEI был выбран для инициализации HRD, поскольку HRD можно инициализировать в один из периодов буферизации сообщений SEI.

- Переполнение CPB определено как условие, при котором общее число битов в CPB больше, чем размер CPB. Буфер CPB никогда не должен переполняться.
- Незаполнение CPB определено как условие, при котором $t_{r,n}(n)$ меньше $t_{af}(n)$. Если `low_delay_hrd_flag` равно 0, буфер CPB никогда не должен переполняться.
- Номинальное время удалений изображений из CPB (начиная со второго изображения в порядке декодирования) должно удовлетворять ограничениям на $t_{r,n}(n)$ и $t_r(n)$, выраженным в пп. А.3.1–А.3.3 для профилей и уровней, определенных в потоке битов.
- Сразу же после добавления любого декодированного изображения к DPB заполнение DPB должно быть меньше или равно размеру DPB, как это ограничено Приложениями А, D и E для профилей и уровней, определенных в потоке битов.
- Если это необходимо для предсказания, то все контрольные изображения должны быть представлены в DPB. Каждое изображение должно быть представлено в DPB во время поступления на выход DPB, если только оно вообще не хранилось в DPB или не было удалено из DPB до времени поступления на выход одним из процессов, определенных в п. С.2.

- Значение $\Delta t_{o,dpb}(n)$, данное уравнением C-13, которое выражает разность между временем выхода изображения и временем изображения, следующим за ним сразу в порядке выхода, должно удовлетворять ограничению, выраженному в п. A.3.1 для профилей и уровней, определенных в потоке битов.

C.4 Соответствие декодера

Декодер, соответствующий настоящей Рекомендации | Международному стандарту, должен удовлетворять следующим требованиям.

Декодер, заявленный как соответствующий конкретным профилю и уровню, должен быть в состоянии успешно декодировать все соответствующие потоки битов (спецификация соответствия приведена в п. C.3), при условии, что все наборы параметров последовательностей и изображений относятся к блокам NAL VCL, а соответствующие сообщения SEI о периоде буферизации и о синхронизации изображения поступают в декодер своевременно либо в потоке битов (не блоками NAL VCL), либо от внешних устройств, не определенных этой Рекомендацией | Международным стандартом.

Существуют два типа соответствия, которые должны быть заявлены для декодера: соответствие времени выхода и соответствие порядка выхода.

Чтобы проверить соответствие декодера, тестовые потоки битов, соответствующие заявленным профилям и уровням, как это описано в п. C.3, вводятся планировщиком гипотетического потока (HSS) как в HRD, так и в испытуемый декодер (DUT). Все изображения на выходе HRD должны быть также выходом DUT. Для каждого изображения на выходе HRD значения всех образцов, которые появляются на выходе DUT для соответствующего изображения, должны быть равны значениям образцов на выходе HRD.

Для соответствия синхронизации выхода декодера HSS работает, как описано выше, с вводом таблиц, выбранных только из подмножества значений SchedSelIdx, для которых скорость битов и размер CPB ограничены, как это указано в Приложении A, для определенных профилей и уровней, или с вводом "интерполированных" таблиц, как это определено ниже, для которых скорость битов и размер CPB ограничены, как указано в Приложении A. Такой же ввод таблиц используют для HRD и DUT.

Если параметры HRD и сообщение SEI о периоде буферизации представлены со значением `cpb_cnt_minus1`, большим 0, декодер должен быть в состоянии декодировать поток битов как поступающий от HSS, действуя с использованием "интерполированных" таблиц, определенных как имеющих пиковую скорость битов r , размер CPB $c(r)$ и начальную задержку удаления CPB $(f(r) \div r)$, следующим образом:

$$\alpha = (r - \text{BitRate}[\text{SchedSelIdx} - 1]) \div (\text{BitRate}[\text{SchedSelIdx}] - \text{BitRate}[\text{SchedSelIdx} - 1]), \quad (\text{C-17})$$

$$c(r) = \alpha * \text{CpbSize}[\text{SchedSelIdx}] + (1 - \alpha) * \text{CpbSize}[\text{SchedSelIdx} - 1], \quad (\text{C-18})$$

$$f(r) = \alpha * \text{initial_cpb_removal_delay}[\text{SchedSelIdx}] * \text{BitRate}[\text{SchedSelIdx}] + (1 - \alpha) * \text{initial_cpb_removal_delay}[\text{SchedSelIdx} - 1] * \text{BitRate}[\text{SchedSelIdx} - 1], \quad (\text{C-19})$$

для любого $\text{SchedSelIdx} > 0$ и r такого, что $\text{BitRate}[\text{SchedSelIdx} - 1] \leq r \leq \text{BitRate}[\text{SchedSelIdx}]$, с тем, чтобы r и $c(r)$ находились в границах, как указано в Приложении A, максимальной скорости битов и размера буфера для определенных профилей и уровней.

ПРИМЕЧАНИЕ 1. – Значение `initial_cpb_removal_delay[SchedSelIdx]` может отличаться от такого же значения периода буферизации и его следует вычислить еще раз.

Для соответствия декодера синхронизации выхода в HRD, как это было описано выше, используют синхронизацию (относительно ввода первого бита) выхода изображения одинаково как для HRD, так и для DUT, вплоть до фиксированной задержки.

Для соответствия декодера порядку выхода HSS вводит поток битов в DUT "по требованию" от DUT, подразумевая, что HSS вводит биты (в порядке декодирования) только в том случае, если DUT требует больше битов для продолжения обработки.

ПРИМЕЧАНИЕ 2. – Это означает, что для этого теста минимальный буфер кодированного изображения DUT может быть таким же, как размер наибольшего блока доступа.

Как описано ниже, для этого используется модифицированный буфер HRD, а HSS вводит поток битов в HRD с помощью одной из таблиц, определенных в потоке битов, так, что скорость битов и размер CPB ограничиваются, как это определено в Приложении A. Порядок выхода изображения должен быть одинаковым для HRD и для DUT.

Для соответствия декодера порядку выхода размер CPB HRD равен `CpbSize[SchedSelIdx]` для выбранной таблицы, а размер DPB равен `MaxDpbSize`. Время удаления из CPB для HRD равно времени поступления заключительного бита, а декодирование происходит безотлагательно. Работа буфера DPB этого HRD описана ниже.

C.4.1 Работа DPB по организации очередности выхода

Буфер декодированного изображения содержит буферы кадров. Каждый из буферов кадров может содержать декодированный кадр, декодированную пару дополнительных полей или единственное (непарное) декодированное поле, которое помечено как "используется для контроля" (контрольного изображения), или каждый из буферов можно содержать для будущего выхода (изображения с измененным порядком). При инициализации HRD заполнение DPB, измеряемое в кадрах, устанавливается на 0. Все следующие шаги происходят одновременно, если блок доступа удален из CPB в перечисленном порядке.

C.4.2 Декодирование промежутков в frame_num и хранение "несуществующих" изображений

В случае применения промежутки в значении frame_num обнаруживают процессом декодирования, а необходимое число "несуществующих" кадров учитывают в порядке, определенном генерацией значения UnusedShortTermFrameNum в уравнении 7-21, и отмечают, как это определено в п. 8.2.5.2. Буферы кадров, содержащие кадр или пару дополнительных полей или непарное поле, которые помечены как "не требуется для выхода" и "не используется для контроля", опустошаются (не имеют выхода), а заполнение DPB уменьшается на число пустых буферов кадров. Каждый "несуществующий" кадр хранят в DPB следующим образом.

- Если не имеется пустого буфера кадров (т.е. заполнение DPB равно размеру DPB), процесс "пульсации", определенный в п. C.4.5.3, активируют повторно до тех пор, пока не образуется пустой буфер кадров, в котором можно хранить "несуществующий" кадр.
- "Несуществующий" кадр хранят в пустом буфере кадров и помечают как "не требуется для выхода", а заполнение DPB увеличивают на единицу.

C.4.3 Декодирование изображения

Первично кодированное изображение является декодированным, и его временно хранят (но не в DPB).

C.4.4 Удаление изображения из DPB перед возможным введением текущего изображения

Удаление изображения из DPB перед возможным введением текущего изображения происходит следующим образом.

- Если декодированное изображение – это IDR изображение, то применяют следующее.
 - Все контрольные изображения в DPB помечают как "не используется для контроля", как это определено в п. 8.2.5.
 - Если IDR изображение – это не первое декодированное IDR изображение, а значения PicWidthInMbs, или FrameHeightInMbs, или max_dec_frame_buffering, найденные из действующего набора параметров последовательности, отличаются от значений PicWidthInMbs или FrameHeightInMbs или max_dec_frame_buffering, найденных из набора параметров последовательности, который был действующим соответственно для предыдущей последовательности, то в HRD значение no_output_of_prior_pics_flag принимают равным 1 независимо от действительного значения no_output_of_prior_pics_flag.
 - ПРИМЕЧАНИЕ. – Реальные декодеры должны пытаться обрабатывать изменения значения PicWidthInMbs или FrameHeightInMbs или max_dec_frame_buffering более мягко, чем HRD.
 - Если no_output_of_prior_pics_flag равно 1 или предполагают равным 1, то все буферы кадров в DPB опустошают без выдачи на выход изображений, которые там содержались, а заполнение DPB устанавливают на 0.
- Иначе (декодированное изображение – это не IDR изображение) активируют процесс разметки декодированного контрольного изображения, определенный в п. 8.2.5. Буферы кадров, содержащие кадр, или пару дополнительных полей или непарное поле, которые помечают как "не требуется для выхода" и "не используется для контроля", опустошаются (без выхода), а заполнение DPB уменьшают на число опустошенных буферов кадров.

Если текущее изображение имеет memory_management_control_operation, равное 5, или является IDR изображением, для которого no_output_of_prior_pics_flag не равно 1 и не принимается равным 1, выполняются следующие два шага.

1. Буферы кадров, содержащие кадр или пару дополнительных полей или непарное поле, помеченные как "не требуется для выхода" и "не используется для контроля", опустошаются (не имеют выхода), а заполнение DPB уменьшают на число опустошенных буферов кадров.
2. Все не пустые буферы кадров в DPB повторно опустошают, запуская процесс "пульсации", определенный в п. C.4.5.3, а заполнение DPB устанавливают на 0.

C.4.5 Разметка и хранение текущего декодированного изображения

C.4.5.1 Разметка и хранение контрольного декодированного изображения в DPB

Если текущее изображение – это контрольное изображение, то его хранят в DPB следующим образом.

- Если текущее декодированное изображение является вторым полем (в порядке декодирования) пары дополнительных контрольных полей, а первое поле пары все еще находится в DPB, то текущее

декодированное изображение сохраняется в том же буфере кадра, что и первое поле пары, и помечается как "требуется для выхода".

- Иначе выполняют следующие операции.
 - Если отсутствует пустой буфер кадров (т. е. заполнение DPB равно размеру DPB), процесс "пульсации", определенный в п. С.4.5.3, повторно активируют до тех пор, пока не появится пустой буфер кадров, в котором можно хранить текущее декодированное изображение.
 - Текущее декодированное изображение хранят в пустом буфере кадров и помечают как "требуется для выхода", а заполнение DPB увеличивают на единицу.

С.4.5.2 Разметка и хранение неконтрольного декодированного изображения в DPB

Если текущее изображение – это неконтрольное изображение, то выполняют следующие действия.

- Если текущее декодированное изображение – это второе поле (в порядке декодирования) дополнительной неконтрольной пары полей, а первое поле пары еще находится в DPB, то текущее изображение сохраняется в том же буфере кадров, что и первое поле пары, и помечается как "требуется для выхода".
- Иначе повторно выполняют следующие действия до тех пор, пока текущее декодированное изображение не будет разделено на кадры (или поля) на выходе или сохранено в DPB.
 - Если отсутствует пустой буфер кадров (т. е. заполнение DPB равно размеру DPB), применяют следующее:
 - если текущее изображение не содержит более низкого значения `PicOrderCnt()`, чем все изображения в DPB, которые помечены как "необходимые для выхода", то выполняют процесс "пульсации", определенный в п. С.4.5.3;
 - иначе (текущее изображение содержит более низкие значения `PicOrderCnt()`, чем все изображения в DPB, которые помечены как "необходимые для выхода") текущее изображение кадрируют (разделяют на кадры или поля), используя прямоугольник кадрирования, определенный в наборе параметров последовательности для данной последовательности, а разделенное на кадры изображение является выходом.
 - Иначе (существует пустой буфер кадров, т. е. заполнение DPB меньше, чем размер DPB) текущее декодированное изображение хранят в пустом буфере кадров и помечают как "необходимое для выхода", а заполнение DPB увеличивается на единицу.

С.4.5.3 Процесс "пульсации"

Процесс "пульсации" активируют в следующих случаях.

- Отсутствует пустой буфер кадров (т. е. заполнение DPB равно размеру DPB), а пустой буфер кадров требуется для хранения предполагаемого "несуществующего" кадра, как это определено в п. С.4.2.
- Текущее изображение – это IDR изображение, а значение `no_output_of_prior_pics_flag` не равно 1, и не предполагают, что оно будет равно 1, как это определено в п. С.4.4.
- Текущее изображение имеет `memory_management_control_operation`, равное 5, как это определено в п. С.4.4.
- Отсутствует пустой буфер кадров (т. е. заполнение DPB равно размеру DPB), а пустой буфер кадров требуется для хранения декодированного (не IDR) контрольного изображения, как это определено в п. С.4.5.1.
- Отсутствует пустой буфер кадров (т. е. заполнение DPB равно размеру DPB). Текущее изображение не является неконтрольным изображением, которое не является вторым полем дополнительной пары неконтрольных полей, а в DPB существуют изображения, помеченные как "необходимые для выхода", которые предшествуют текущему неконтрольному изображению в порядке выхода, как это определено в п. С.4.5.2. Поэтому для хранения текущего изображения необходим пустой буфер.

Процесс "пульсации" заключается в следующем.

- Следующим образом выбирают изображение или пару дополнительных контрольных полей, которые являются первыми на выходе.
 - Выбирают такой буфер кадров, который содержит изображение, имеющее наименьшее значение `PicOrderCnt()` из всех изображений в DPB, помеченных как "необходимые для выхода".
 - Если этот буфер кадров содержит пару дополнительных неконтрольных полей с обоими полями, помеченными как "необходимые для выхода", и оба поля имеют то же значение `PicOrderCnt()`, то первое из этих полей в порядке декодирования рассматривают первым в качестве выхода.
 - Иначе, если этот буфер кадров содержит пару дополнительных контрольных полей с обоими полями, помеченными как "необходимые для выхода", и оба поля имеют то же значение `PicOrderCnt()`, то собственно пару дополнительных контрольных полей рассматривают в качестве выхода.
 - Иначе изображение в этом буфере кадров, которое имеет наименьшее значение `PicOrderCnt()`, рассматривают первым в качестве выхода.
- Если единственное изображение рассматривают первым в качестве выхода, то это изображение кадрируют, используя прямоугольник кадрирования, определенный в наборе параметров последовательности, а кадрированное изображение считают выходом и помечают как "не требуется для выхода".

- Иначе (пару дополнительных контрольных полей первую рассматривают в качестве выхода) кадрируют оба поля из этой дополнительной пары контрольных полей, используя прямоугольник кадрирования, определенный в наборе параметров последовательности, оба поля дополнительной пары контрольных полей вместе составляют выход, и оба поля дополнительной пары контрольных полей помечают как "не требуется для выхода".
- Буфер кадров, который включает изображение или пару дополнительных контрольных полей, которые были разделены на кадры, а также выход проверяют, и если любое из следующих условий удовлетворено, то буфер кадров опустошают, а заполнение DPB уменьшают на единицу.
 - Буфер кадров содержит неконтрольное непарное поле.
 - Буфер кадров содержит неконтрольный кадр.
 - Буфер кадров содержит пару дополнительных неконтрольных полей, а оба поля помечены как "не требуется для выхода".
 - Буфер кадров содержит непарное контрольное поле, помеченное как "не используется для контроля".
 - Буфер кадров содержит контрольный кадр с двумя полями, помеченными как "не используется для контроля".
 - Буфер кадров содержит пару дополнительных контрольных полей, и оба поля помечены как "не используется для контроля" и "не требуется для выхода".

Приложение D

Дополнительная расширенная информация

(Данное Приложение является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Это Приложение определяет синтаксис и семантику полезной нагрузки сообщения SEI.

Сообщения SEI помогают в процессах, связанных с декодированием, отображением и для других целей. Однако сообщения SEI не требуются для построения образцов яркости или цветности процессом декодирования. Соответствие декодеров не требуется, чтобы обработать эту информацию для согласования порядка выхода с этой Рекомендацией | Международным стандартом (см. Приложение C для спецификации соответствия). Некоторая информация сообщения SEI требуется для проверки соответствия потока битов, а также соответствия декодера для синхронизации выхода.

В Приложении D спецификация присутствия сообщений SEI также выполняется, если эти сообщения (или некоторое их подмножество) передают в декодеры (или в HRD) другими средствами, не определенными в настоящей Рекомендации | Международном стандарте. Если сообщения SEI присутствуют в потоке битов, они должны подчиняться синтаксису и семантике, которые определены в пп. 7.3.2.3 и 7.4.2.3, также в этом Приложении. Если содержание сообщения SEI передают приложению какими-нибудь средствами иными, чем те, которые имеются в потоке битов, то для представления содержания сообщения SEI не требуется использование того же синтаксиса, который определен в этом Приложении. Для подсчета битов учитывают только соответствующие биты, которые действительно находятся в потоке битов.

D.1 Синтаксис полезной нагрузки SEI

sei_payload(payloadType, payloadSize) {	С	Дескриптор
if(payloadType == 0)		
buffering_period(payloadSize)	5	
else if(payloadType == 1)		
pic_timing(payloadSize)	5	
else if(payloadType == 2)		
pan_scan_rect(payloadSize)	5	
else if(payloadType == 3)		
filler_payload(payloadSize)	5	
else if(payloadType == 4)		
user_data_registered_itu_t_t35(payloadSize)	5	
else if(payloadType == 5)		
user_data_unregistered(payloadSize)	5	
else if(payloadType == 6)		
recovery_point(payloadSize)	5	
else if(payloadType == 7)		
dec_ref_pic_marking_repetition(payloadSize)	5	
else if(payloadType == 8)		
spare_pic(payloadSize)	5	
else if(payloadType == 9)		
scene_info(payloadSize)	5	
else if(payloadType == 10)		
sub_seq_info(payloadSize)	5	
else if(payloadType == 11)		
sub_seq_layer_characteristics(payloadSize)	5	
else if(payloadType == 12)		
sub_seq_characteristics(payloadSize)	5	
else if(payloadType == 13)		
full_frame_freeze(payloadSize)	5	
else if(payloadType == 14)		
full_frame_freeze_release(payloadSize)	5	
else if(payloadType == 15)		
full_frame_snapshot(payloadSize)	5	
else if(payloadType == 16)		
progressive_refinement_segment_start(payloadSize)	5	
else if(payloadType == 17)		
progressive_refinement_segment_end(payloadSize)	5	
else if(payloadType == 18)		
motion_constrained_slice_group_set(payloadSize)	5	
else if(payloadType == 19)		
film_grain_characteristics(payloadSize)	5	
else if(payloadType == 20)		
deblocking_filter_display_preference(payloadSize)	5	
else if(payloadType == 21)		
stereo_video_info(payloadSize)	5	
else		
reserved_sei_message(payloadSize)	5	

if(!byte_aligned()) {		
bit_equal_to_one /* равно 1 */	5	f(1)
while(!byte_aligned())		
bit_equal_to_zero /* равно 0 */	5	f(1)
}		
}		

D.1.1 Синтаксис сообщения SEI о периоде буферизации

	С	Дескриптор
buffering_period(payloadSize) {		
seq_parameter_set_id	5	ue(v)
if(NalHrdBpPresentFlag) {		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
initial_cpb_removal_delay [SchedSelIdx]	5	u(v)
initial_cpb_removal_delay_offset [SchedSelIdx]	5	u(v)
}		
}		
if(VclHrdBpPresentFlag) {		
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
initial_cpb_removal_delay [SchedSelIdx]	5	u(v)
initial_cpb_removal_delay_offset [SchedSelIdx]	5	u(v)
}		
}		
}		

D.1.2 Синтаксис сообщения SEI о синхронизации изображения

	С	Дескриптор
pic_timing(payloadSize) {		
if(CpbDpbDelaysPresentFlag) {		
cpb_removal_delay	5	u(v)
dpb_output_delay	5	u(v)
}		
if(pic_struct_present_flag) {		
pic_struct	5	u(4)
for(i = 0; i < NumClockTS ; i++) {		
clock_timestamp_flag [i]	5	u(1)
if(clock_timestamp_flag[i]) {		
ct_type	5	u(2)
nuit_field_based_flag	5	u(1)
counting_type	5	u(5)
full_timestamp_flag	5	u(1)
discontinuity_flag	5	u(1)
cnt_droppeded_flag	5	u(1)
n_frames	5	u(8)
if(full_timestamp_flag) {		
seconds_value /* 0..59 */	5	u(6)
}		
}		

minutes_value /* 0..59 */	5	u(6)
hours_value /* 0..23 */	5	u(5)
} else {		
seconds_flag	5	u(1)
if(seconds_flag) {		
seconds_value /* диапазон 0..59 */	5	u(6)
minutes_flag	5	u(1)
if(minutes_flag) {		
minutes_value /* 0..59 */	5	u(6)
hours_flag	5	u(1)
if(hours_flag)		
hours_value /* 0..23 */	5	u(5)
}		
}		
}		
if(time_offset_length > 0)		
time_offset	5	i(v)
}		
}		
}		
}		

D.1.3 Синтаксис сообщения SEI о прямоугольнике сканирования

pan_scan_rect(payloadSize) {	C	Дескриптор
pan_scan_rect_id	5	ue(v)
pan_scan_rect_cancel_flag	5	u(1)
if(!pan_scan_rect_cancel_flag) {		
pan_scan_cnt_minus1	5	ue(v)
for(i = 0; i <= pan_scan_cnt_minus1; i++) {		
pan_scan_rect_left_offset[i]	5	se(v)
pan_scan_rect_right_offset[i]	5	se(v)
pan_scan_rect_top_offset[i]	5	se(v)
pan_scan_rect_bottom_offset[i]	5	se(v)
}		
pan_scan_rect_repetition_period	5	ue(v)
}		
}		

D.1.4 Синтаксис сообщения SEI о полезной нагрузке заполнителя

filler_payload(payloadSize) {	C	Дескриптор
for(k = 0; k < payloadSize; k++)		
ff_byte /* равно 0xFF */	5	f(8)
}		

D.1.5 Синтаксис сообщения SEI о данных пользователя, зарегистрированных по Рекомендации МСЭ-Т Т.35

	С	Дескриптор
user_data_registered_itu_t_t35(payloadSize) {		
itu_t_t35_country_code	5	b(8)
if(itu_t_t35_country_code != 0xFF)		
i = 1		
else {		
itu_t_t35_country_code_extension_byte	5	b(8)
i = 2		
}		
do {		
itu_t_t35_payload_byte	5	b(8)
i++		
} while(i < payloadSize)		
}		

D.1.6 Синтаксис сообщения SEI о незарегистрированных данных пользователя

	С	Дескриптор
user_data_unregistered(payloadSize) {		
uuid_iso_iec_11578	5	u(128)
for(i = 16; i < payloadSize; i++)		
user_data_payload_byte	5	b(8)
}		

D.1.7 Синтаксис сообщения SEI о пункте восстановления

	С	Дескриптор
recovery_point(payloadSize) {		
recovery_frame_cnt	5	ue(v)
exact_match_flag	5	u(1)
broken_link_flag	5	u(1)
changing_slice_group_idc	5	u(2)
}		

D.1.8 Синтаксис сообщения SEI о повторной разметке декодированного контрольного изображения

	С	Дескриптор
dec_ref_pic_marking_repetition(payloadSize) {		
original_idr_flag	5	u(1)
original_frame_num	5	ue(v)
if(!frame_mbs_only_flag) {		
original_field_pic_flag	5	u(1)
if(original_field_pic_flag)		
original_bottom_field_flag	5	u(1)
}		
dec_ref_pic_marking()	5	
}		

D.1.9 Синтаксис сообщения SEI о резервном изображении

	С	Дескриптор
spare_pic(payloadSize) {		
target_frame_num	5	ue(v)
spare_field_flag	5	u(1)
if(spare_field_flag)		
target_bottom_field_flag	5	u(1)
num_spare_pics_minus1	5	ue(v)
for(i = 0; i < num_spare_pics_minus1 + 1; i++) {		
delta_spare_frame_num[i]	5	ue(v)
if(spare_field_flag)		
spare_bottom_field_flag[i]	5	u(1)
spare_area_idc[i]	5	ue(v)
if(spare_area_idc[i] == 1)		
for(j = 0; j < PicSizeInMapUnits; j++)		
spare_unit_flag[i][j]	5	u(1)
else if(spare_area_idc[i] == 2) {		
mapUnitCnt = 0		
for(j=0; mapUnitCnt < PicSizeInMapUnits; j++) {		
zero_run_length[i][j]	5	ue(v)
mapUnitCnt += zero_run_length[i][j] + 1		
}		
}		
}		
}		

D.1.10 Синтаксис сообщения SEI о сценической информации

	С	Дескриптор
scene_info(payloadSize) {		
scene_info_present_flag	5	u(1)
if(scene_info_present_flag) {		
scene_id	5	ue(v)
scene_transition_type	5	ue(v)
if(scene_transition_type > 3)		
second_scene_id	5	ue(v)
}		
}		

D.1.11 Синтаксис сообщения SEI об информации субпоследовательности

	С	Дескриптор
sub_seq_info(payloadSize) {		
sub_seq_layer_num	5	ue(v)
sub_seq_id	5	ue(v)
first_ref_pic_flag	5	u(1)
leading_non_ref_pic_flag	5	u(1)
last_pic_flag	5	u(1)
sub_seq_frame_num_flag	5	u(1)
if(sub_seq_frame_num_flag)		
sub_seq_frame_num	5	ue(v)
}		

D.1.12 Синтаксис сообщения SEI о характеристиках слоя субпоследовательности

	С	Дескриптор
sub_seq_layer_characteristics(payloadSize) {		
num_sub_seq_layers_minus1	5	ue(v)
for(layer = 0; layer <= num_sub_seq_layers_minus1; layer++) {		
accurate_statistics_flag	5	u(1)
average_bit_rate	5	u(16)
average_frame_rate	5	u(16)
}		
}		

D.1.13 Синтаксис сообщения SEI о характеристиках субпоследовательности

	С	Дескриптор
sub_seq_characteristics(payloadSize) {		
sub_seq_layer_num	5	ue(v)
sub_seq_id	5	ue(v)
duration_flag	5	u(1)
if(duration_flag)		
sub_seq_duration	5	u(32)
average_rate_flag	5	u(1)
if(average_rate_flag) {		
accurate_statistics_flag	5	u(1)
average_bit_rate	5	u(16)
average_frame_rate	5	u(16)
}		
num_referenced_subseqs	5	ue(v)
for(n = 0; n < num_referenced_subseqs; n++) {		
ref_sub_seq_layer_num	5	ue(v)
ref_sub_seq_id	5	ue(v)
ref_sub_seq_direction	5	u(1)
}		
}		

D.1.14 Синтаксис сообщения SEI о полностью фиксированном кадре

full_frame_freeze(payloadSize) {	C	Дескриптор
full_frame_freeze_repetition_period	5	ue(v)
}		

D.1.15 Синтаксис сообщения SEI об отмене режима полностью фиксированного кадра

full_frame_freeze_release(payloadSize) {	C	Дескриптор
}		

D.1.16 Синтаксис сообщения SEI о стоп-кадре

full_frame_snapshot(payloadSize) {	C	Дескриптор
snapshot_id	5	ue(v)
}		

D.1.17 Синтаксис сообщения SEI о запуске сегмента последовательного улучшения

progressive_refinement_segment_start(payloadSize) {	C	Дескриптор
progressive_refinement_id	5	ue(v)
num_refinement_steps_minus1	5	ue(v)
}		

D.1.18 Синтаксис сообщения SEI об окончании сегмента последовательного улучшения

progressive_refinement_segment_end(payloadSize) {	C	Дескриптор
progressive_refinement_id	5	ue(v)
}		

D.1.19 Синтаксис сообщения SEI о наборе группы секций ограниченного движения

motion_constrained_slice_group_set(payloadSize) {	C	Дескриптор
num_slice_groups_in_set_minus1	5	ue(v)
for(i = 0; i <= num_slice_groups_in_set_minus1; i++)		
slice_group_id[i]	5	u(v)
exact_sample_value_match_flag	5	u(1)
pan_scan_rect_flag	5	u(1)
if(pan_scan_rect_flag)		
pan_scan_rect_id	5	ue(v)
}		

D.1.20 Синтаксис сообщения SEI о характеристиках зернистости пленки

	С	Дескриптор
film_grain_characteristics(payloadSize) {		
film_grain_characteristics_cancel_flag	5	u(1)
if(!film_grain_characteristics_cancel_flag) {		
model_id	5	u(2)
separate_colour_description_present_flag	5	u(1)
if(separate_colour_description_present_flag) {		
film_grain_bit_depth_luma_minus8	5	u(3)
film_grain_bit_depth_chroma_minus8	5	u(3)
film_grain_full_range_flag	5	u(1)
film_grain_colour_primaries	5	u(8)
film_grain_transfer_characteristics	5	u(8)
film_grain_matrix_coefficients	5	u(8)
}		
blending_mode_id	5	u(2)
log2_scale_factor	5	u(4)
for(c = 0; c < 3; c++)		
comp_model_present_flag[c]	5	u(1)
for(c = 0; c < 3; c++)		
if(comp_model_present_flag[c]) {		
num_intensity_intervals_minus1[c]	5	u(8)
num_model_values_minus1[c]	5	u(3)
for(i = 0; i <= num_intensity_intervals_minus1[c]; i++) {		
intensity_interval_lower_bound[c][i]	5	u(8)
intensity_interval_upper_bound[c][i]	5	u(8)
for(j = 0; j <= num_model_values_minus1[c]; j++)		
comp_model_value[c][i][j]	5	se(v)
}		
}		
}		
film_grain_characteristics_repetition_period	5	ue(v)
}		
}		

D.1.21 Семантика сообщения SEI о предпочтениях отображения с учетом фильтра устранения блочности

	С	Дескриптор
deblocking_filter_display_preference(payloadSize) {		
deblocking_display_preference_cancel_flag	5	u(1)
if(!deblocking_display_preference_cancel_flag) {		
display_prior_to_deblocking_preferred_flag	5	u(1)
dec_frame_buffering_constraint_flag	5	u(1)
deblocking_display_preference_repetition_period	5	ue(v)
}		
}		

D.1.22 Синтаксис сообщения SEI об информации, касающейся стереоизображения

stereo_video_info(payloadSize) {	C	Дескриптор
field_views_flag	5	u(1)
if(field_views_flag)		
top_field_is_left_view_flag	5	u(1)
else {		
current_frame_is_left_view_flag	5	u(1)
next_frame_is_second_view_flag	5	u(1)
}		
left_view_self_contained_flag	5	u(1)
right_view_self_contained_flag	5	u(1)
}		

D.1.23 Синтаксис зарезервированных сообщений SEI

reserved_sei_message(payloadSize) {	C	Дескриптор
for(i = 0; i < payloadSize; i++)		
reserved_sei_message_payload_byte	5	b(8)
}		

D.2 Семантика полезной нагрузки SEI

D.2.1 Семантика сообщения SEI о периоде буферизации

Если NalHrdVpPresentFlag или VclHrdVpPresentFlag равно 1, то сообщение SEI о периоде буферизации может быть связано с любым блоком доступа в потоке битов. Сообщение SEI о периоде буферизации должно быть связано с каждым IDR блоком доступа, а каждый блок доступа связан с сообщением SEI о пункте восстановления.

ПРИМЕЧАНИЕ. – Для некоторых приложений частое появление сообщения SEI о периоде буферизации может быть нежелательным.

Период буферизации определен как набор блоков доступа между двумя событиями – сообщениями SEI о периоде буферизации – в порядке декодирования.

seq_parameter_set_id определяет набор параметров последовательности, который содержит последовательность атрибутов HRD. Значение seq_parameter_set_id должно быть равно значению seq_parameter_set_id в установке параметров изображения, на которую ссылается начальное кодированное изображение, связанное с сообщением SEI о периоде буферизации. Значение seq_parameter_set_id должно быть в диапазоне от 0 до 31 включительно.

initial_cpb_removal_delay[SchedSelIdx] определяет задержку для SchedSelIdx-ого буфера CPB между временем поступления в CPB первого бита кодированных данных, связанных с блоком доступа, который, в свою очередь, связан с сообщением SEI о периоде буферизации, и временем удаления из CPB кодированных данных, связанных с тем же блоком доступа за первый период буферизации после инициализации HRD. Элемент синтаксиса имеет длину в битах, заданную начальным значением initial_cpb_removal_delay_length_minus1 + 1. Он расположен в блоках с тактовой частотой 90 кГц. Значение initial_cpb_removal_delay[SchedSelIdx] не должно быть равно 0 и не должно превышать $90\,000 * (\text{CpbSize[SchedSelIdx]} \div \text{BitRate[SchedSelIdx]})$, что является временным эквивалентом размера CPB для синхронизирующих импульсов 90 кГц.

initial_cpb_removal_delay_offset[SchedSelIdx] используют для SchedSelIdx-ого буфера CPB в комбинации с cpb_removal_delay для определения начального времени доставки кодированных блоков доступа в CPB. Значение initial_cpb_removal_delay_offset[SchedSelIdx] записано в единицах тактов 90 кГц. Элемент синтаксиса initial_cpb_removal_delay_offset[SchedSelIdx] – это фиксированная длина кода, у которого длина в битах задана initial_cpb_removal_delay_length_minus1 + 1. Этот элемент синтаксиса не используют декодеры, и он необходим только для доставки планировщика (HSS), определенного в Приложении C.

Вдобавок к собственно кодированной видеопоследовательности сумма initial_cpb_removal_delay[SchedSelIdx] и initial_cpb_removal_delay_offset[SchedSelIdx] должна быть константой для каждого значения SchedSelIdx.

D.2.2 Семантика сообщения SEI о синхронизации изображения

Наличие сообщения SEI о синхронизации изображения определяется следующим образом.

- Если `CpbDpbDelaysPresentFlag` равно 1 или `pic_struct_present_flag` равно 1, в каждом блоке доступа кодированной видеопоследовательности должно быть представлено одно сообщение SEI о синхронизации изображения.
- Иначе (`CpbDpbDelaysPresentFlag` равно 0 и `pic_struct_present_flag` равно 0) в любом блоке доступа кодированной видеопоследовательности не должно быть представлено ни одного сообщения SEI о синхронизации изображения.

Если `CpbDpbDelaysPresentFlag` равно 1, сообщение SEI о синхронизации изображения должно быть связано с каждым блоком доступа в потоке битов.

cpb_removal_delay определяет, сколько тактов системных часов (см. п. E.2.1) требуется ожидать после удаления из CPB блока доступа, связанного с самым последним сообщением SEI о периоде буферизации перед удалением из буфера блока доступа данных, связанных с сообщением SEI о синхронизации изображения. Это значение также используют, чтобы вычислить самое раннее возможное время поступления данных блока доступа в CPB для HSS, как это определено в Приложении C. Элемент синтаксиса является кодом с фиксированной длиной, чья длина в битах задана `cpb_removal_delay_length_minus1 + 1`. Значение `cpb_removal_delay` – это остаток $2^{(\text{cpb_removal_delay_length_minus1} + 1)}$ счетчика.

Значение `cpb_removal_delay` для первого изображения в потоке битов должно быть равно 0.

dpb_output_delay используют для вычисления времени выхода из DPB изображения. Оно определяет, сколько тактов системных часов требуется ожидать после удаления из CPB блока доступа перед тем, как декодированное изображение может появиться на выходе DPB (см. п. C.2).

ПРИМЕЧАНИЕ 1. – Изображение не удаляют из DPB в назначенное время, если оно все еще помечено как "используется для краткосрочного контроля" или "используется для долгосрочного контроля".

ПРИМЕЧАНИЕ 2. – Для декодированного изображения определено только одно значение `dpb_output_delay`.

Размер элемента синтаксиса `dpb_output_delay` задан в битах `dpb_output_delay_length_minus1 + 1`. Если `max_dec_frame_buffering` равно 0, то `dpb_output_delay` должно быть равно 0.

Время выхода, полученное из `dpb_output_delay` для любого изображения, которое является выходом от выходного сигнала синхронизации согласованного декодера, как это определено в п. C.2, должно предшествовать времени выхода, полученного из `dpb_output_delay` для всех изображений в любой кодированной видеопоследовательности в порядке декодирования.

Время выхода, полученное из `dpb_output_delay` второго поля в порядке декодирования пары дополнительных контрольных полей, не должно превышать времени выхода, полученного из `dpb_output_delay` первого поля той же пары дополнительных контрольных полей.

Порядок изображения на выходе, установленный значениями этого элемента синтаксиса, должен быть таким же, как порядок, установленный значениями `PicOrderCnt()`, как это определено в пп. C.4.1–C.4.5, за исключением тех случаев, когда два поля пары дополнительных контрольных полей имеют то же значение `PicOrderCnt()` и различные времена выхода.

Для изображений, которые не появляются на выходе с помощью процесса "пульсаций" из п. C.4.5, поскольку они предшествуют в порядке декодирования IDR изображению с `no_output_of_prior_pics_flag`, равным 1 (или который считают равным 1), времена выхода, полученные из `dpb_output_delay`, должны возрастать с ростом значения `PicOrderCnt()` относительно всех изображений в той же самой кодированной видеопоследовательности, следующей за любым изображением с `memory_management_control_operation`, равным 5.

pic_struct указывает, действительно ли изображение может быть выведено на дисплей в виде кадра или одного или более полей, согласно таблице D-1. Дублирование кадров (`pic_struct` равно 7) указывает, что кадр может быть выведен на дисплей последовательно два раза, а утроение кадров (`pic_struct` равно 8) указывает, что кадр может быть выведен на дисплей последовательно три раза.

ПРИМЕЧАНИЕ 3. – Дублирование кадров можно ослабить для отображения, например, 25 строк видео на 50-строчный экран и 29,97 строк видео на 59,94-строчный экран. Использование дублирования и утроения кадров в комбинации с каждым другим кадром может облегчить отображение 23,98 строк видео на 59,94-строчный экран.

Таблица D-1 – Интерпретация pic_struct

Значение	Указанный вывод изображения	Ограничения	NumClockTS
0	кадр	field_pic_flag должно быть 0	1
1	верхнее поле	field_pic_flag должно быть 1, bottom_field_flag должно быть 0	1
2	нижнее поле	field_pic_flag должно быть 1, bottom_field_flag должно быть 1	1
3	верхнее поле, нижнее поле – в таком порядке	field_pic_flag должно быть 0	2
4	нижнее поле, верхнее поле – в таком порядке	field_pic_flag должно быть 0	2
5	верхнее поле, нижнее поле, верхнее поле повторно – в таком порядке	field_pic_flag должно быть 0	3
6	нижнее поле, верхнее поле, нижнее поле повторно – в таком порядке	field_pic_flag должно быть 0	3
7	дублирование кадра	field_pic_flag должно быть 0 fixed_frame_rate_flag должно быть 1	2
8	утроение кадра	field_pic_flag должно быть 0 fixed_frame_rate_flag должно быть 1	3
9..15	зарезервировано		

NumClockTS определено с помощью pic_struct, как это показано в таблице D-1. Имеется до NumClockTS разных наборов информационных отметок времени для изображения, как это определено значением clock_timestamp_flag[i], в каждом наборе. Наборы информационных отметок времени применяют к полю (полям) или к кадру (кадрам), связанным с изображением с помощью pic_struct.

Содержание элементов синтаксиса отметок времени указывает на время возникновения, захвата или альтернативного, идеального отображения. Указанное время вычисляют как:

$$\text{clockTimestamp} = ((\text{hN} * 60 + \text{mM}) * 60 + \text{sS}) * \text{time_scale} + \text{nFrames} * (\text{num_units_in_tick} * (1 + \text{nuit_field_based_flag})) + \text{tOffset}, \quad (\text{D-1})$$

в единицах тактовых импульсов синхронизатора с задающей частотой, равной time_scale Гц, относительно некоторой неопределенной точки, для которой clockTimestamp равно 0. Значение clockTimestamp не влияет на порядок выхода и синхронизации выхода DPB. Если два или более кадров с pic_struct, равным 0, следуют друг за другом в порядке выхода и имеют равные значения clockTimestamp, то это означает, что кадры представляют то же содержание и что последний такой кадр предпочтительней для представления в порядке выхода.

ПРИМЕЧАНИЕ 4. – Индикация времени clockTimestamp может сделать более простым вывод на устройства с обновленными скоростями, чем индикация, подстроенная под временные отметки на выходе DPB.

clock_timestamp_flag[i], равное 1, указывает, что элементы синтаксиса числа отметок времени присутствуют и следуют сразу за элементами синтаксиса. Значение clock_timestamp_flag[i], равное 0, указывает, что связанные с отметками времени элементы синтаксиса отсутствуют. Если NumClockTS больше 1, а clock_timestamp_flag[i] равно 1 более чем для одного значения i, то значение clockTimestamp должно быть неубывающим при возрастании значений i.

ct_type указывает на тип сканирования (чередующийся или последовательный) исходного материала, как определено в таблице D-2.

Два поля кодированного кадра могут иметь разные значения ct_type.

Если clockTimestamp равно для двух полей противоположной четности, которые последовательны в порядке выхода, и оба поля с ct_type, равным 0 (последовательно), или ct_type, равным 2 (неизвестно), то это указывает, что оба поля поступили от одного и того же последовательного кадра. Два последовательных поля в порядке выхода должны иметь различные значения clockTimestamp, если значение ct_type для каждого поля равно 1 (чередование).

**Таблица D-2 – Отображение `st_type` на сканирование
исходного изображения**

Значение	Сканирование исходного изображения
0	последовательное
1	чередующееся
2	неизвестное
3	зарезервировано

`nuit_field_based_flag`: используют для вычисления `clockTimestamp`, как это определено в уравнении D-1.

`counting_type`: определяет метод разделения на кадры значений `n_frames`, как это определено в таблице D-3.

Таблица D-3 – Определение значений `counting_type`

Значение	Интерпретация
0	нет разделения на кадры вычисленных значений <code>n_frames</code> , и не используют <code>time_offset</code>
1	нет разделения на кадры вычисленных значений <code>n_frames</code>
2	разделение на кадры отдельных нулевых вычисленных значений <code>n_frames</code>
3	разделение на кадры отдельных вычисленных значений <code>n_frames</code> MaxFPS-1
4	разделение на кадры двух наименьших (значение 0 и 1) вычисленных значений <code>n_frames</code> , если <code>seconds_value</code> равно 0, а <code>minutes_value</code> – не целое кратное 10
5	разделение на кадры неопределенных отдельных значений <code>n_frames</code>
6	разделение на кадры неопределенного числа неопределенных значений <code>n_frames</code>
7..31	зарезервировано

`full_timestamp_flag`, равное 1, определяет, что элемент синтаксиса `n_frames` следует за `seconds_value`, `minutes_value` и `hours_value`. Значение `full_timestamp_flag`, равное 0, определяет, что элемент синтаксиса `n_frames` следует за `seconds_flag`.

`discontinuity_flag`, равное 0, указывает, что разность между значением `clockTimestamp` и значением `clockTimestamp`, вычисленную из прежних отметок времени в порядке выхода, можно интерпретировать как разность временем возникновения или захвата объединенных кадров или полей. Значение `discontinuity_flag`, равное 1, указывает, что разность между текущим значением `clockTimestamp` и значением `clockTimestamp`, вычисленную из прежних отметок времени в порядке выхода, нельзя интерпретировать как разность времени возникновения или захвата объединенных кадров или полей. Если `discontinuity_flag` равно 0, то значение `clockTimestamp` должно быть больше чем или равно всем значениям `clockTimestamp`, присутствующим в предшествующем изображении в порядке выхода из DPB.

`cnt_dropped_flag` определяет пропуск одного или более значений `n_frames`, используя метод вычислений, определенный `counting_type`.

`n_frames` определяет значение `nFrames`, использованное для вычисления `clockTimestamp`. `n_frames` должно быть меньше следующего значения:

$$\text{MaxFPS} = \text{Ceil}(\text{time_scale} \div \text{num_units_in_tick}). \quad (\text{D-2})$$

ПРИМЕЧАНИЕ 5. – `n_frames` – это счетчик на основе кадров. Для специального обозначения синхронизации поля используют `time_offset` для указания отличия `clockTimestamp` для каждого поля.

Если `counting_type` равно 2, а `cnt_dropped_flag` равно 1, то `n_frames` должно быть равно 1, а значение `n_frames` для прежнего изображения в порядке выхода не должно быть равно 0 до тех пор, пока `discontinuity_flag` не будет равен 1.

ПРИМЕЧАНИЕ 6. – Если `counting_type` равно 2, необходимых значительных увеличений величин `tOffset` в уравнении D-1, когда используют постоянные не целочисленные значения частоты кадров (например, 12,5 кадров в секунду с `time_scale`, равным 25, и `num_units_in_tick`, равным 2, а `nuit_field_based_flag`, равным 0), можно избежать, пропуская иногда при подсчетах значение `n_frames`, равное 0 (например, подсчитывая `n_frames` от 0 до 12, а затем увеличивая `seconds_value` и считая `n_frames` от 1 до 12, а затем увеличивая `seconds_value` и `counting_n_frames` от 0 до 12, и т. д.).

Если `counting_type` равно 3, а `cnt_dropped_flag` равно 1, то `n_frames` должно быть равно 0, а значение `n_frames` для предыдущего изображения в порядке выхода не должно быть равно `MaxFPS` – 1 до тех пор, пока `discontinuity_flag` не будет равен 1.

ПРИМЕЧАНИЕ 7. – Если `counting_type` равно 3, необходимых значительных увеличений величин `tOffset` в уравнении D-1, когда используют постоянные не целочисленные значения частоты кадров (например, 12,5 кадров в секунду с `time_scale`, равным 25, и `num_units_in_tick`, равным 2, а `nuit_field_based_flag`, равным 0), можно избежать, пропуская иногда при подсчетах значение `n_frames`, равное 0 (например, подсчитывая `n_frames` от 0 до 12, а затем увеличивая `seconds_value` и считая `n_frames` от 1 до 12, а затем увеличивая `seconds_value` и подсчитывая `n_frames` от 0 до 12, и т. д.).

Если `counting_type` равно 4, а `cnt_dropped_flag` равно 1, то `n_frames` должно быть равно 2, а особое значение `sS` должно быть нулем, особое значение `mM` не должно быть целым кратным 10, `n_frames` для предыдущего изображения в порядке выхода не должно быть равно 0 или 1 до тех пор, пока `discontinuity_flag` не будет равен 1.

ПРИМЕЧАНИЕ 8. – Если `counting_type` равно 4, необходимые значительные увеличения величин `tOffset` в уравнении D-1, когда используют постоянные не целочисленные значения частоты кадров (например, 30000÷1001 кадров в секунду с `time_scale`, равным 60000, `num_units_in_tick`, равным 1001, а `nuit_field_based_flag`, равным 1) можно снизить, пропуская иногда при подсчетах значение `n_frames` равное `MaxFPS` (например, подсчитывая `n_frames` от 0 до 29, а затем увеличивая `seconds_value` и считая `n_frames` от 0 до 29, и т. д., до тех пор пока `seconds_value` не будет нулем, а `minutes_value` не будет целым кратным 10, а затем считая `n_frames` от 2 до 29, затем увеличивая `seconds_value` и считая `n_frames` от 0 до 29, и т. д.). Этот метод подсчета хорошо известен в промышленности, и на него часто ссылаются как на метод "NTSC drop-frame".

Если `counting_type` равно 5 или 6, а `cnt_dropped_flag` равно 1, то `n_frames` не должно быть равно 1 плюс значение `n_frames` для предыдущего изображения в порядке выхода по модулю `MaxFPS` до тех пор, пока `discontinuity_flag` не будет равен 1.

ПРИМЕЧАНИЕ 9. – Если `counting_type` равно 5 или 6, необходимых значительных увеличений величин `tOffset` в уравнении D-1, когда используют постоянные не целочисленные скорости кадров, можно избежать, пропуская иногда при подсчетах значение `n_frames`. Особые значения `n_frames`, которые пропускают, не являются определенными, если `counting_type` равно 5 или 6.

seconds_flag, равное 1, указывает, что `seconds_value` и `minutes_flag` присутствуют, если `full_timestamp_flag` равно 0. Значение `seconds_flag`, равное 0, указывает, что `seconds_value` и `minutes_flag` отсутствуют.

seconds_value указывает, что значение `sS` используют для вычисления `clockTimestamp`. Значение `seconds_value` должно быть в диапазоне от 0 до 59 включительно. Если `seconds_value` не присутствует, то для вычисления `clockTimestamp` должно быть использовано предыдущее значение `seconds_value` в порядке декодирования вместо `sS`.

minutes_flag, равное 1, указывает, что `minutes_value` и `hours_flag` присутствуют, если `full_timestamp_flag` равно 0, а `seconds_flag` равно 1. Значение `minutes_flag`, равное 0, указывает, что `minutes_value` и `hours_flag` отсутствуют.

minutes_value указывает, что значение `mM` используют для вычисления `clockTimestamp`. Значение `minutes_value` должно быть в диапазоне от 0 до 59 включительно. Если `minutes_value` не присутствует, то для вычисления `clockTimestamp` должно быть использовано предыдущее значение `minutes_value` в порядке декодирования вместо `mM`.

hours_flag, равное 1, указывает, что `hours_value` присутствует, если `full_timestamp_flag` равно 0 и `seconds_flag` равно 1, а `minutes_flag` равно 1.

hours_value указывает, что для вычисления `clockTimestamp` используют значение `hH`. Значение `hours_value` должно быть в диапазоне от 0 до 23 включительно. Если `hours_value` отсутствует, то для вычисления `clockTimestamp` должно быть использовано предыдущее значение `hours_value` в порядке декодирования в виде `hH`.

time_offset указывает, что значение `tOffset` используют для вычисления `clockTimestamp`. Число битов, используемое для представления `time_offset`, должно быть равно `time_offset_length`. Если `time_offset` отсутствует, то для вычисления `clockTimestamp` должно быть использовано значение 0 вместо `tOffset`.

D.2.3 Семантика сообщения SEI о прямоугольнике сканирования

Элементы синтаксиса сообщения SEI о прямоугольнике сканирования определяют координаты прямоугольника относительно разделения (на кадры) прямоугольника при установке параметров последовательности. Каждая координата этого прямоугольника определена в единицах одной шестнадцатой образца относительно опорной сетки яркости.

pan_scan_rect_id содержит идентифицирующий номер, который может быть использован для идентификации цели прямоугольника сканирования (например, для идентификации прямоугольника как площади, которая должна быть видна на конкретном экране, или как площади, на которой присутствует на сцене конкретный актер). Значение `pan_scan_rect_id` должно быть в диапазоне от 0 до $2^{32} - 1$ включительно.

Значения `pan_scan_rect_id` от 0 до 255 и от 512 до $2^{31} - 1$ может быть использовано, как это определено приложением. Значения `pan_scan_rect_id` от 256 до 511 и от 2^{31} до $2^{32} - 1$ зарезервированы для будущего использования МСЭ-Т | ИСО/МЭК. Декодеры, встречающие значение `pan_scan_rect_id` в диапазоне от 256 до 511 или в диапазоне от 2^{31} до $2^{32} - 1$, должны его игнорировать (удалять из потока битов и отбрасывать).

`pan_scan_rect_cancel_flag`, равное 1, указывает, что сообщение SEI аннулирует последствие любого предыдущего сообщения SEI о прямоугольнике сканирования в порядке выхода. Значение `pan_scan_rect_cancel_flag`, равное 0, указывает, что далее следует информация о прямоугольнике сканирования.

`pan_scan_cnt_minus1` определяет число прямоугольников сканирования, которые присутствуют в сообщении SEI. Значение `pan_scan_cnt_minus1` должно быть в диапазоне от 0 до 2 включительно. Значение `pan_scan_cnt_minus1`, равное 0, указывает, что присутствует единственный прямоугольник сканирования, который использует все поля декодированного изображения. Значение `pan_scan_cnt_minus1` должно быть равно 0, если текущее изображение – это поле. Значение `pan_scan_cnt_minus1`, равное 1, указывает, что присутствуют два прямоугольника сканирования, первый из которых используют для первого поля изображения в порядке выхода, а второй – для второго поля изображения в порядке выхода. Значение `pan_scan_cnt_minus1`, равное 2, указывает, что присутствуют три прямоугольника сканирования. Первый используют для первого поля изображения в порядке выхода, второй – для второго поля изображения в порядке выхода, а третий используют для повторения первого поля как третьего в порядке выхода.

`pan_scan_rect_left_offset[i]`, `pan_scan_rect_right_offset[i]`, `pan_scan_rect_top_offset[i]` и `pan_scan_rect_bottom_offset[i]` определяют местоположение прямоугольника сканирования, как целые величины со знаками в единицах одной шестнадцатой образца относительно опорной сетки яркости. Значения каждого из этих четырех элементов синтаксиса должно быть в диапазоне от -2^{31} до $2^{31} - 1$ включительно.

Прямоугольник сканирования определен в единицах одной шестнадцатой образца относительно опорной сетки яркости как область с горизонтальными координатами кадра яркости от $16 * \text{CropUnitX} * \text{frame_crop_left_offset} + \text{pan_scan_rect_left_offset}[i]$ до $16 * (16 * \text{PicWidthInMbs} - \text{CropUnitX} * \text{frame_crop_right_offset}) + \text{pan_scan_rect_right_offset}[i] - 1$ и с вертикальными координатами от $16 * \text{CropUnitY} * \text{frame_crop_top_offset} + \text{pan_scan_rect_top_offset}[i]$ до $16 * (16 * \text{PicHeightInMbs} - \text{CropUnitY} * \text{frame_crop_bottom_offset}) + \text{pan_scan_rect_bottom_offset}[i] - 1$ включительно. Значение $16 * \text{CropUnitX} * \text{frame_crop_left_offset} + \text{pan_scan_rect_left_offset}[i]$ должно быть меньше или равно $16 * (16 * \text{PicWidthInMbs} - \text{CropUnitX} * \text{frame_crop_right_offset}) + \text{pan_scan_rect_right_offset}[i] - 1$; а значение $16 * \text{CropUnitY} * \text{frame_crop_top_offset} + \text{pan_scan_rect_top_offset}[i]$ должно быть меньше или равно $16 * (16 * \text{PicHeightInMbs} - \text{CropUnitY} * \text{frame_crop_bottom_offset}) + \text{pan_scan_rect_bottom_offset}[i] - 1$.

Если площадь прямоугольника сканирования включает образцы вне разделенного (на кадры) прямоугольника, то область вне этого прямоугольника может быть заполнена синтезированным контентом (например, черным или нейтральным серым видеоконтентом) для вывода на экран.

`pan_scan_rect_repetition_period` определяет длительность хранения сообщения SEI о прямоугольнике сканирования и может определять интервал подсчета порядка изображений, в течение которого другое сообщение SEI о прямоугольнике сканирования с таким же значением `pan_scan_rect_id` или с концом кодированной видеопоследовательности должно присутствовать в потоке битов. Значение `pan_scan_rect_repetition_period` должно быть в диапазоне от 0 до 16 384 включительно. Если `pan_scan_cnt_minus1` больше 0, то `pan_scan_rect_repetition_period` не должно превышать 1.

`pan_scan_rect_repetition_period`, равное 0, определяет, что информация о прямоугольнике сканирования применима только к текущему декодированному изображению.

`pan_scan_rect_repetition_period`, равное 1, определяет, что информацию о прямоугольнике сканирования сохраняется в порядке выхода, пока любое из следующих условий не станет истиной.

- Начинается новая кодированная видеопоследовательность.
- Изображение в блоке доступа, которое содержит сообщение SEI о прямоугольнике сканирования с тем же значением `pan_scan_rect_id`, является выходом, имеющим `PicOrderCnt()`, значение которого больше, чем `PicOrderCnt(CurrPic)`.

`pan_scan_rect_repetition_period`, равное 0 или 1, указывает, что другое сообщение SEI о прямоугольнике сканирования с тем же значением `pan_scan_rect_id` может присутствовать, а может и не присутствовать.

`pan_scan_rect_repetition_period` больше 1 определяет, что информацию о прямоугольнике сканирования сохраняется в порядке выхода, пока любое из следующих условий не станет истиной.

- Начинается новая кодированная видеопоследовательность.
- Изображение в блоке доступа, которое содержит сообщение SEI о прямоугольнике сканирования с тем же значением `pan_scan_rect_id`, является выходом, имеющим `PicOrderCnt()`, значение которого больше `PicOrderCnt(CurrPic)` и меньше или равно `PicOrderCnt(CurrPic) + pan_scan_rect_repetition_period`.

`pan_scan_rect_repetition_period` больше 1 указывает, что другое сообщение SEI о прямоугольнике сканирования с тем же значением `pan_scan_rect_id` должно присутствовать в блоке доступа для изображения, которое служит выходом

c PicOrderCnt(), значение которого больше PicOrderCnt(CurrPic) и меньше или равно PicOrderCnt(CurrPic) + pan_scan_rect_repetition_period, если не заканчивается поток битов или не начинается новая кодированная видеопоследовательность без такого изображения на выходе.

D.2.4 Семантика сообщения SEI о полезной нагрузке заполнителя

Это сообщение содержит ряд байтов payloadSize значения 0xFF, которое может отбрасываться.

ff_byte должно быть байтом со значением 0xFF.

D.2.5 Семантика сообщения SEI о данных пользователя, зарегистрированных по Рекомендации МСЭ-Т Т.35

Это сообщение содержит данные пользователя, зарегистрированные, как это определено в Рекомендации МСЭ-Т Т.35. Содержание этих данных не определено настоящей Рекомендацией | Международным стандартом.

itu_t_t35_country_code должно быть байтом, имеющим значение определенного кода страны, согласно Приложению А к Рекомендации МСЭ-Т Т.35.

itu_t_t35_country_code_extension_byte должно быть байтом, имеющим значение определенного кода страны, согласно Приложению В к Рекомендации МСЭ-Т Т.35.

itu_t_t35_payload_byte должно быть байтом, который содержит данные, зарегистрированные, как это определено Рекомендацией МСЭ-Т Т.35.

Код провайдера терминала в соответствии с Рекомендацией МСЭ-Т Т.35 и ориентированный код провайдера терминала должны содержаться в первом или более байтах itu_t_t35_payload_byte в формате, определенном администрацией, которая публикует код провайдера терминала. Любые остающиеся данные itu_t_t35_payload_byte должны быть данными с синтаксисом и семантикой, которые определены идентифицированными по МСЭ-Т Т.35 кодом страны и кодом провайдера терминала.

D.2.6 Семантика сообщения SEI о незарегистрированных данных пользователя

Это сообщение содержит идентифицированные UUID незарегистрированные данные пользователя, содержание которых не определено этой Рекомендацией | Международным стандартом.

uuid_iso_iec_11578 должно иметь значение, определенное как UUID, согласно процедуре Приложения А ИСО/МЭК 11578:1996.

user_data_payload_byte должно быть байтом, который содержит данные с синтаксисом и семантикой, как это определено производителем UUID.

D.2.7 Семантика сообщения SEI о пункте восстановления

Сообщение SEI о пункте восстановления помогает декодеру определить, когда процесс декодирования обеспечит вывод изображений на экран после инициирования произвольного доступа или после того, как кодер укажет на разомкнутое звено связи в последовательности. Если процесс декодирования начинают в порядке декодирования с блока доступа, связанного с сообщением SEI о пункте восстановления, то обо всех декодированных изображениях в этом или последующем пункте в порядке выхода, определенном в этом сообщении SEI, будет указано как о правильных или приблизительно правильных по содержанию. Декодированные изображения, полученные произвольным доступом к изображению или ранее и связанные с сообщением SEI о пункте восстановления, не требуют исправления содержания до тех пор, пока не поступит указания от пункта восстановления. Действие процесса декодирования, которое начинается с изображения, связанного с сообщением SEI о пункте восстановления, может содержать ссылки на изображения, недоступные буферу декодированного изображения.

Кроме того, используя broken_link_flag, сообщение SEI о пункте восстановления может указать декодеру местоположение некоторых изображений в потоке битов, которые возникают в результате серьезных визуальных дефектов изображения при выводе на экран, если даже процесс декодирования начался в месте расположения предыдущего IDR блока доступа в порядке декодирования.

ПРИМЕЧАНИЕ 1. – Значение broken_link_flag может быть использовано кодерами, чтобы указать место расположения пункта, после которого процесс декодирования при декодировании некоторых изображений может вызвать ссылки на изображения, которые будучи доступными в процессе декодирования не являются изображениями, которые можно было бы использовать как контрольные, если бы поток битов был бы кодирован изначально (т. е. благодаря операции секционирования, которую выполнили при генерации потока битов).

Пункт восстановления устанавливает, в единицах приращений frame_num, следующего за frame_num текущего блока доступа в месте сообщения SEI.

ПРИМЕЧАНИЕ 2. – Если в потоке битов присутствует информация HRD, то сообщение SEI о периоде буферизации должно быть объединено с блоком доступа, связанного с сообщением SEI о пункте восстановления, чтобы после произвольного доступа выполнить инициализацию модели буфера HRD.

recovery_frame_cnt определяет пункт восстановления выходных изображений в порядке выхода. Все декодированные изображения в порядке выхода указаны как правильные или приблизительно правильные по содержанию, начиная с положения в порядке выхода контрольного изображения, имеющего frame_num, равным frame_num блоков NAL текущего блока доступа, с приращением recovery_frame_cnt по арифметике модуля MaxFrameNum. Значение recovery_frame_cnt должно быть в диапазоне от 0 до MaxFrameNum – 1 включительно.

exact_match_flag указывает, действительно ли декодированные изображения – в указанном пункте восстановления и далее в порядке выхода, – которые находят, запуская процесс декодирования в блоке доступа, связанного с сообщением SEI о пункте восстановления, должны в точности совпадать с изображениями, которые могли бы быть созданы запуском

процесса декодирования в месте расположения предыдущего IDR блока доступа в потоке блоков NAL. Значение 0 указывает, что не требуется точного совпадения, а значение 1 указывает, что совпадение должно быть точным.

Если декодирование начинают с места сообщения SEI о пункте восстановления, все ссылки на недоступность контрольных изображений должны восприниматься как ссылки на изображения, которые содержат только макроблоки, использующие режимы Intra предсказания, и образцы, заданные значениями Y, равным 128, Cb, равным 128, и Cr, равным 128 (средний уровень серого), с целью определения соответствия значения exact_match_flag.

ПРИМЕЧАНИЕ 3. – При выполнении произвольного доступа декодеры должны учитывать все ссылки на недоступные контрольные изображения как ссылки на изображения, которые содержат только intra макроблоки и образцы, заданные значениями Y, равным 128, Cb, равным 128, и Cr, равным 128 (средний уровень серого), независимо от значения exact_match_flag.

Если exact_match_flag равно 0, качество аппроксимации в пункте восстановления выбирают процессом кодирования, не определенным этой Рекомендацией | Международным стандартом.

broken_link_flag указывает на присутствие или отсутствие разомкнутого звена связи в потоке блоков NAL в месте сообщения SEI о пункте восстановления. Этому значению присвоена следующая семантика.

- Если broken_link_flag равно 1, то изображения, созданные запуском процесса декодирования в месте предыдущего IDR блока доступа, могут содержать визуальные дефекты, нежелательные до такой степени, что декодированные изображения в этом и последующих блоках доступа, связанных с сообщением SEI о пункте восстановления в порядке декодирования, не смогут быть выведены на экран в порядке выхода до определенного пункта восстановления.
- Иначе (broken_link_flag равно 0) не дано никаких указаний на любое потенциальное присутствие нежелательных визуальных дефектов.

Независимо от значения broken_link_flag следующие до определенного пункта восстановления в порядке выхода изображения определены таким образом, чтобы правильно или приблизительно правильно передать содержание.

ПРИМЕЧАНИЕ 4. – Если субпоследовательность информации сообщения SEI присутствует вместе с сообщением SEI о пункте восстановления, в котором broken_link_flag равно 1, и если sub_seq_layer_num равно 0, то sub_seq_id должно отличаться от самого последнего значения sub_seq_id для sub_seq_layer_num, равного 0, которое было декодировано до места сообщения SEI о пункте восстановления. Если broken_link_flag равно 0, то значение sub_seq_id в субпоследовательности слоя 0 следует оставить неизменным.

changing_slice_group_idc, равное 0, указывает, что декодированные изображения правильны или приблизительно правильны по содержанию в этом и следующем пункте восстановления в порядке выхода, если все макроблоки первичных кодированных изображений декодированы в период изменения группы секции, т. е. в промежутке между блоком доступа, связанного с сообщением SEI о пункте восстановления (включительно), и указанным пунктом восстановления (включительно) в порядке декодирования. Значение changing_slice_group_idc должно быть равно 0, если num_slice_groups_minus1 равно 0 в любом первичном кодированном изображении в период изменения группы секции.

Если changing_slice_group_idc равно 1 или 2, то num_slice_groups_minus1 должно быть равно 1, а тип отображения макроблока в группу секции 3, 4 или 5 должен быть применен к каждому первичному кодированному изображению в период изменения группы секции.

changing_slice_group_idc, равное 1, указывает, что в период изменения группы секции никакие значения образцов вне декодированных макроблоков в группе секции 0 не используют для inter предсказания любого макроблока в группе секции 0. Кроме того, changing_slice_group_idc, равное 1, указывает, что если все макроблоки в группе секции 0 в период изменения группы секции декодированы, то декодированные изображения будут правильными или приблизительно правильными по содержанию в этом и в следующем указанном пункте восстановления в порядке выхода, независимо от того, декодирован ли любой макроблок в группе секции 1 в период изменения группы секции.

changing_slice_group_idc, равное 2, указывает, что в период изменения группы секции никакие значения образцов вне декодированных макроблоков в группе секции 1 не используют для inter предсказания любого макроблока в группе секции 1. Кроме того, changing_slice_group_idc, равное 2, указывает, что если все макроблоки в группе секции 1 в период изменения группы секции декодированы, то декодированные изображения будут правильными или приблизительно правильными по содержанию в этом и в следующем указанном пункте восстановления в порядке выхода, независимо от того, декодирован ли любой макроблок в группе секции 1 в период изменения группы секции.

changing_slice_group_idc должно быть в диапазоне от 0 до 2 включительно.

D.2.8 Семантика сообщения SEI о повторной разметке декодированного контрольного изображения

Сообщение SEI о повторной разметке декодированного контрольного изображения используют, чтобы повторить структуру синтаксиса разметки декодированного контрольного изображения, которая находилась в заголовке секции более раннего изображения в последовательности порядка декодирования.

original_idr_flag должно быть равно 1, если структура синтаксиса разметки декодированного контрольного изображения появилась ранее в IDR изображении. Значение original_idr_flag должно быть равно 0, если структура синтаксиса разметки повторного декодированного контрольного изображения не появилась ранее в IDR изображении.

original_frame_num должно быть равно значению **frame_num** изображения, в котором ранее появилась структура синтаксиса разметки повторного декодированного контрольного изображения. Изображение, отмеченное как **original_frame_num**, – это предыдущее кодированное изображение с определенным значением **frame_num**. Значение **original_frame_num**, использованное для ссылки на изображение с **memory_management_control_operation**, равным 5, должно быть нулем.

original_field_pic_flag должно быть равно **field_pic_flag** изображения, в котором ранее появилась структура синтаксиса разметки повторного декодированного контрольного изображения.

original_bottom_field_flag должно быть равно **bottom_field_flag** изображения, в котором ранее появилась структура синтаксиса разметки повторного декодированного контрольного изображения.

dec_ref_pic_marking() должна содержать копию структуры синтаксиса разметки декодированного контрольного изображения, у которого значение **frame_num** было **original_frame_num**. Значение **nal_unit_type**, используемое для соответствия повторного значения **dec_ref_pic_marking()** структуры синтаксиса, должно быть **nal_unit_type** в заголовке секции(ий) изображения, у которого значение **frame_num** было **original_frame_num** (т.е. значение **nal_unit_type**, использованное в п. 7.3.3.3, должно считаться равным 5, если **original_idr_flag** равно 1, и не должно считаться равным 5, если **original_idr_flag** равно 0).

D.2.9 Семантика сообщения SEI о резервном изображении

Это сообщение SEI указывает, что отдельные блоки отображения группы секции, которые называют резервными блоками отображения группы секции, в одном или более декодированных контрольных изображениях имеют сходство с так называемыми блоками отображения группы секции в определенном декодированном изображении, которое называют целевым изображением. Резервный блок отображения группы секции может быть использован для замены близко расположенного неправильно декодированного блока отображения группы секции в целевом изображении. Декодированное изображение, которое содержит резервные блоки отображения группы секции, называют резервным изображением.

Для всех резервных изображений, идентифицированных в сообщении SEI о резервном изображении, значение **frame_mbs_only_flag** должно быть равно значению **frame_mbs_only_flag** целевого изображения в том же сообщении SEI. Резервные изображения в сообщении SEI ограничены следующим образом.

- Если целевое изображение – это декодированное поле, то все резервные изображения, идентифицированные в том же сообщении SEI, должно быть декодированными полями.
- Иначе (целевое изображение – это декодированный кадр) все резервные изображения, идентифицированные в том же сообщении SEI, должно быть декодированными кадрами.

Для всех резервных изображений, идентифицированных в сообщении SEI о резервном изображении, значения **pic_width_in_mbs_minus1** и **pic_height_in_map_units_minus1** должны быть равны значениям **pic_width_in_mbs_minus1** и **pic_height_in_map_units_minus1**, соответственно, целевого изображения в том же сообщении SEI. Изображение, связанное (как это определено в п. 7.4.1.2.3) с этим сообщением, должно появляться в порядке декодирования после целевого изображения.

target_frame_num указывает значение **frame_num** целевого изображения.

spare_field_flag, равное 0, указывает, что целевое изображение и резервные изображения – это декодированные кадры. Значение **spare_field_flag**, равное 1, указывает, что целевое изображение и резервные изображения – это декодированные поля.

target_bottom_field_flag, равно 0, указывает, что целевое изображение – это верхнее поле. **target_bottom_field_flag**, равно 1, указывает, что целевое изображение – это нижнее поле.

Целевое изображение – это декодированное контрольное изображение, чье соответствие первичному кодированному изображению предшествует в порядке декодирования текущему изображению и в котором значения **frame_num**, **field_pic_flag** (если присутствует) и **bottom_field_flag** (если присутствует) равны **target_frame_num**, **spare_field_flag** и **target_bottom_field_flag**, соответственно.

num_spare_pics_minus1 указывает число резервных изображений для определенного целевого изображения. Число резервных изображений равно **num_spare_pics_minus1 + 1**. Значение **num_spare_pics_minus1** должно быть в диапазоне от 0 до 15 включительно.

delta_spare_frame_num[i] используют для идентификации резервного изображения, которое содержит *i*-й набор резервных блоков отображения группы секции, здесь и далее называемых *i*-м резервным изображением, как определено ниже. Значение **delta_spare_frame_num[i]** должно быть в диапазоне от 0 до **MaxFrameNum – 1 – !spare_field_flag** включительно.

Значение **frame_num** *i*-ого резервного изображения, т.е. **spareFrameNum[i]**, находят следующим образом для всех значений *i* от 0 до **num_spare_pics_minus1** включительно:

```
candidateSpareFrameNum = target_frame_num – !spare_field_flag
for ( i = 0; i <= num_spare_pics_minus1; i++ ) {
    if( candidateSpareFrameNum < 0 )
        candidateSpareFrameNum = MaxFrameNum – 1
    spareFrameNum[ i ] = candidateSpareFrameNum – delta_spare_frame_num[ i ]
}
```

 (D-3)

```

    if( spareFrameNum[ i ] < 0 )
        spareFrameNum[ i ] = MaxFrameNum + spareFrameNum[ i ]
        candidateSpareFrameNum = spareFrameNum[ i ] - !spare_field_flag
    }.

```

spare_bottom_field_flag[i], равное 0, указывает, что i-е резервное изображение – это верхнее поле. **spare_bottom_field_flag[i]**, равное 1, указывает, что i-ое резервное изображение – это нижнее поле.

Нулевое (0-е) резервное изображение – это декодированное контрольное изображение, чье соответствие первичному кодированному изображению предшествует в порядке декодирования целевому изображению, и в котором значения **frame_num**, **field_pic_flag** (если присутствует) и **bottom_field_flag** (если присутствует) равны **spareFrameNum[0]**, **spare_field_flag** и **spare_bottom_field_flag[0]**, соответственно. i-е резервное изображение – это декодированное контрольное изображение, чье соответствие первичному кодированному изображению в порядке декодирования предшествует (i – 1)-е резервное изображение и в котором значения **frame_num**, **field_pic_flag** (если присутствует) и **bottom_field_flag** (если присутствует) равны **spareFrameNum[i]**, **spare_field_flag** и **spare_bottom_field_flag[i]**, соответственно.

spare_area_idc[i] указывает на метод, использованный для идентификации резервных блоков отображения группы секции в i-м резервном изображении. Значение **spare_area_idc[i]** должно быть в диапазоне от 0 до 2 включительно. Значение **spare_area_idc[i]**, равное 0, указывает, что все блоки отображения группы секции в i-м резервном изображении – это резервные блоки. Значение **spare_area_idc[i]**, равное 1, указывает, что значение элемента синтаксиса **spare_unit_flag[i][j]** используют для идентификации резервных блоков отображения группы секции. Значение **spare_area_idc[i]**, равное 2, указывает, что элемент синтаксиса **zero_run_length[i][j]** используют для вывода значения **spareUnitFlagInBoxOutOrder[i][j]**, как описано ниже.

spare_unit_flag[i][j], равное 0, указывает, что j-й блок отображения группы секции при растровом порядке сканирования в i-м резервном изображении – это резервный блок. Значение **spare_unit_flag[i][j]** равное 1, указывает, что j-й блок отображения группы секции при растровом порядке сканирования в i-м резервном изображении – это не резервный блок.

zero_run_length[i][j] используют для вывода значения **spareUnitFlagInBoxOutOrder[i][j]**, если **spare_area_idc[i]** равно 2. В этом случае резервные блоки отображения группы секции, идентифицированные в **spareUnitFlagInBoxOutOrder[i][j]**, появляются в счетчике по часовой стрелке, как это определено в п. 8.2.2.4, для каждого резервного изображения. Значение **spareUnitFlagInBoxOutOrder[i][j]**, равное 0, указывает, что j-й блок отображения группы секции в счетчике по часовой стрелке в i-м резервном изображении – это резервный блок. Значение **spareUnitFlagInBoxOutOrder[i][j]**, равное 1, указывает, что j-й блок отображения группы секции в счетчике по часовой стрелке в i-м резервном изображении – это не резервный блок.

Если **spare_area_idc[0]**, равно 2, то **spareUnitFlagInBoxOutOrder[0][j]** находят следующим образом:

```

for( j = 0, loop = 0; j < PicSizeInMapUnits; loop++ ) {
    for( k = 0; k < zero_run_length[ 0 ][ loop ]; k++ )
        spareUnitFlagInBoxOutOrder[ 0 ][ j++ ] = 0
        spareUnitFlagInBoxOutOrder[ 0 ][ j++ ] = 1
    }.

```

(D-4)

Если **spare_area_idc[i]** равно 2 и значение i больше 0, то **spareUnitFlagInBoxOutOrder[i][j]** находят следующим образом:

```

for( j = 0, loop = 0; j < PicSizeInMapUnits; loop++ ) {
    for( k = 0; k < zero_run_length[ i ][ loop ]; k++ )
        spareUnitFlagInBoxOutOrder[ i ][ j ] = spareUnitFlagInBoxOutOrder[ i – 1 ][ j++ ]
        spareUnitFlagInBoxOutOrder[ i ][ j ] = !spareUnitFlagInBoxOutOrder[ i – 1 ][ j++ ]
    }.

```

(D-5)

D.2.10 Семантика сообщения SEI о сценической информации

Сцену и сценический переход здесь и далее определяют как набор последовательных изображений в порядке выхода.

ПРИМЕЧАНИЕ 1. – Декодированные изображения в одной сцене, как правило, имеют одно и то же содержание. Сообщение SEI о сценической информации используют для отметки изображений в идентификаторах сцены и для указаний об изменениях сцены. Это сообщение определяет, как для отмеченных изображений были созданы источники изображений. Декодер может использовать эту информацию, чтобы выбрать подходящий алгоритм для подавления ошибок при передаче. Например, может быть использован специальный алгоритм для подавления ошибок передачи, которые встречаются в изображениях при постепенных изменениях сцены. Более того, сообщение SEI о сценической информации может быть использовано как способ, определенный приложением, например, для индексации сцен в кодированной последовательности.

Сообщение SEI о сценической информации помечает все изображения в порядке декодирования от первично кодированного изображения до того (включительно), с которым связано данное сообщение SEI, как это определено в п. 7.4.1.2.3, а также до того (исключая) первично кодированного изображения, с которым связано следующее сообщение SEI о сценической информации (если такое присутствует) в порядке декодирования, или (иначе) до последнего блока доступа в потоке битов (включительно). Эти изображения здесь и далее считаются целевыми изображениями.

scene_info_present_flag, равное 0, указывает, что не определены сцена или сценический переход, к которым относятся целевые изображения. Значение **scene_info_present_flag**, равное 1, указывает, что данные целевые изображения принадлежат к этой же сцене или сценическому переходу.

scene_id идентифицирует сцену, к которой принадлежат целевые изображения. Если значение **scene_transition_type** целевого изображения меньше 4 и предыдущее изображение в порядке выхода отмечено значением **scene_transition_type**, меньшим 4, а значение **scene_id** такое же, как значение **scene_id** предыдущего изображения в порядке выхода, то это указывает, что исходная сцена для целевых изображений и исходная сцена для предыдущего изображения (в порядке выхода) рассматриваются кодером как одна и та же сцена. Если значение **scene_transition_type** целевого изображения больше 3 и предыдущее изображение в порядке выхода отмечено значением **scene_transition_type**, меньшим 4, а значение **scene_id** такое же, как значение **scene_id** предыдущего изображения в порядке выхода, то это указывает, что исходная сцена для целевых изображений и исходная сцена для предыдущего изображения (в порядке выхода) рассматриваются кодером как одна и та же сцена. Если значение **scene_id** не равно значению **scene_id** предыдущего изображения в порядке выхода, это означает, что целевые изображения и предыдущее изображение (в порядке выхода) рассматриваются кодером как разные сцены.

Значение **scene_id** должно быть в диапазоне от 0 до $2^{32} - 1$ включительно. Значения **scene_id** в диапазоне от 0 до 255 включительно и в диапазоне от 512 до $2^{31} - 1$ включительно может быть использовано, как определено приложением. Значения **scene_id** в диапазоне от 256 до 511 включительно и в диапазоне от 2^{31} до $2^{32} - 1$ включительно зарезервированы для будущего использования МСЭ-Т | ИСО/МЭК. Декодеры, учитывающие значения **scene_id** в диапазоне от 256 до 511 включительно или в диапазоне от 2^{31} до $2^{32} - 1$ включительно, должны их игнорировать (удалять из потока битов и отбрасывать).

scene_transition_type определяет, к какому типу сценического перехода (если такой имеется) принадлежат целевые изображения. Действующие значения **scene_transition_type** показаны в таблице D-4.

Таблица D-4 – Значения scene_transition_type

Значение	Описание
0	Нет перехода
1	Затухание к черному
2	Увеличение от черного
3	Неопределенный переход от или к постоянному цвету
4	Растворение
5	Стирание
6	Неопределенная смесь из двух сцен

Если **scene_transition_type** больше 3, целевые изображения включают содержание обеих сцен, отмеченных своими значениями **scene_id**, а также следующую сцену в порядке выхода, которая отмечена **second_scene_id** (см. ниже). Термин "текущая сцена" используют для указания на сцену, отмеченную как **scene_id**. Термин "следующая сцена" используют для указания на сцену, отмеченную как **second_scene_id**. Нет необходимости каждое последующее изображение в порядке выхода отмечать значением **scene_id**, равным **second_scene_id** в текущем сообщении SEI.

Типы сценических переходов определены следующим образом.

"Нет перехода" определяет, что целевые изображения не участвуют в плавном сценическом переходе.

ПРИМЕЧАНИЕ 2. – Если два последовательных изображения в порядке выхода имеют тип **scene_transition_type**, равный 0, и разные значения **scene_id**, то происходит разделение сцены между этими двумя изображениями.

"Затухание к черному" указывает, что целевые изображения являются частью последовательности изображений в порядке выхода, вовлеченной в сценический переход затухания к черному, т. е. яркость образцов сцены постепенно падает до нуля, а цветность образцов сцены постепенно достигает 128.

ПРИМЕЧАНИЕ 3. – Если два изображения отмечены как принадлежащие к одному сценическому переходу, а тип их сценического перехода – "Затухание к черному", то более позднее изображение в порядке выхода будет более темным, чем предыдущее.

"Увеличение от черного" указывает, что целевые изображения являются частью последовательности изображений в порядке выхода, вовлеченной в сценический переход увеличения от черного, т. е. яркость образцов сцены постепенно возрастает от нуля, а цветность образцов сцены постепенно отклоняется от 128.

ПРИМЕЧАНИЕ 4. – Если два изображения отмечены как принадлежащие к одному сценическому переходу, а тип их сценического перехода – "Увеличение от черного", то более позднее изображение в порядке выхода будет более светлым, чем предыдущее.

"Растворение" указывает, что значения образца каждого целевого изображения (перед кодированием) были созданы вычислением суммы совместных взвешенных значений образцов изображений из текущей сцены и из следующей сцены. Вес текущей сцены постепенно уменьшается от полного уровня до нуля, а вес следующей сцены постепенно возрастает от нуля до полного уровня. Если два изображения отмечены как принадлежащие к одному сценическому переходу, а их тип **scene_transition_type** – это "Растворение", то вес текущей сцены изображения в порядке выхода

будет меньше веса текущего сцены предыдущего изображения, а вес следующей сцены в порядке выхода – больше веса следующей сцены изображения.

"Стирание" указывает, что некоторые значения образца каждого целевого изображения (перед кодированием) были созданы копированием совместных значений образца изображения в текущей сцене и оставшихся значений образцов каждого из целевых изображений (перед кодированием), которые были созданы копированием совместных значений образца изображения в следующей сцене. Если два изображения отмечены как принадлежащие к одному сценическому переходу, а их тип `scene_transition_type` – это "Стирание", то число образцов, скопированных из следующей сцены более позднего изображения (в порядке выхода), будет больше числа образцов, скопированных из следующей сцены для предыдущего изображения.

second_scene_id идентифицирует следующую сцену в постепенном сценическом переходе, в котором участвуют целевые изображения. Значение `second_scene_id` не должно быть равно значению `scene_id`. Значение `second_scene_id` не должно быть равно значению `scene_id` в предыдущем изображении в порядке выхода. Если следующее изображение в порядке выхода помечено значением `scene_transition_type`, меньшим 4, а значение `second_scene_id` – то же, что и `scene_id` следующего изображения (в порядке выхода), это указывает кодеру считать одну из исходных сцен целевым изображением, а исходную сцену следующего изображения (в порядке выхода) – той же самой сценой. Если значение `second_scene_id` не равно значению `scene_id` или `second_scene_id` (если такие присутствуют) следующего изображения в порядке выхода, это указывает на то, что кодер считает целевые изображения и следующее изображение (в порядке выхода) принадлежащими к разным исходным сценам.

Если значение `scene_id` изображения равно значению `scene_id` следующего изображения в порядке выхода, а значение `scene_transition_type` обоих этих изображений меньше 4, это указывает кодеру считать эти два изображения принадлежащими к одной и той же исходной сцене. Если значения `scene_id`, `scene_transition_type` и `second_scene_id` (если присутствует) изображения равно значениям `scene_id`, `scene_transition_type` и `second_scene_id` (соответственно) следующего изображения (в порядке выхода), а значение `scene_transition_type` больше 0, это указывает, что кодер считает эти два изображения возникающими из одного и того же исходного постепенного сценического перехода.

Значение `second_scene_id` должно быть в диапазоне от 0 до $2^{32}-1$ включительно. Значения `second_scene_id` в диапазоне от 0 до 255 включительно и в диапазоне от 512 до $2^{31}-1$ включительно могут быть использованы, как определено приложением. Значения `second_scene_id` в диапазоне от 256 до 511 включительно и в диапазоне от 2^{31} до $2^{32}-1$ включительно зарезервированы для будущего использования МСЭ-Т | ИСО/МЭК. Декодеры, учитывающие значения `second_scene_id` в диапазоне от 256 до 511 включительно или в диапазоне от 2^{31} до $2^{32}-1$ включительно, должны их игнорировать (удалять из потока битов и отбрасывать).

D.2.11 Семантика сообщения SEI об информации субпоследовательности

Сообщение SEI об информации субпоследовательности используют для указания местоположения изображения в иерархии зависимости данных, которые состоят из субпоследовательностей и слоев субпоследовательностей.

Слой субпоследовательности содержит в последовательности подмножество кодированных изображений. Слои субпоследовательностей нумеруют неотрицательными целыми числами. Слой, имеющий больший номер, является высшим по отношению к слою с меньшим номером. Слои упорядочены по их иерархической зависимости друг от друга таким образом, что любое изображение в слое не должно быть предсказано от любого изображения более высокого слоя.

ПРИМЕЧАНИЕ 1. – Другими словами, любое изображение в слое 0 не должно быть предсказано от любого изображения в слое 1 или более высоком. Изображения в слое 1 могут быть предсказаны из слоя 0, изображения в слое 2 могут быть предсказаны из слоев 0 и 1, и т. д.

ПРИМЕЧАНИЕ 2. – Ожидается, что с увеличением номера слоя будет происходить последовательное повышение качества.

Субпоследовательность – это множество кодированных изображений в слое субпоследовательности. Изображение должно располагаться в одном и только в одном слое субпоследовательности. Любое изображение в субпоследовательности не должно быть предсказано от любого изображения в другой субпоследовательности в том же или более высоком слое субпоследовательности. Субпоследовательность в слое 0 может быть декодирована независимо от любого изображения, которое не принадлежит к этой субпоследовательности.

Сообщение SEI об информации субпоследовательности касается текущего блока доступа. первично кодированное изображение в блоке доступа здесь и далее считается текущим изображением.

Сообщение SEI об информации субпоследовательности не должно поступать до тех пор, пока значение `gaps_in_frame_num_value_allowed_flag` в установке параметров последовательности, на которую ссылается сообщение SEI, не будет равно 1.

sub_seq_layer_num определяет номер слоя субпоследовательности текущего изображения. Если `sub_seq_layer_num` больше 0, операции управления памятью не должны использоваться в любом заголовке секции текущего изображения. Если текущее изображение расположено в субпоследовательности с первым изображением в порядке декодирования – IDR изображением, значение `sub_seq_layer_num` должно быть равно 0. Для непарных контрольных полей значение `sub_seq_layer_num` должно быть равно 0. Значение `sub_seq_layer_num` должно быть в диапазоне от 0 до 255 включительно.

sub_seq_id идентифицирует субпоследовательность в слое. Если текущее изображение расположено в субпоследовательности с первым изображением в порядке декодирования – IDR изображением, – значение **sub_seq_id** должно быть таким же, как значение **idr_pic_id** IDR изображения. Значение **sub_seq_id** должно быть в диапазоне от 0 до 65 535 включительно.

first_ref_pic_flag, равное 1, определяет, что текущее изображение – это первое контрольное изображение субпоследовательности в порядке декодирования. Если текущее изображение – это не первое контрольное изображение субпоследовательности в порядке декодирования, **first_ref_pic_flag** должно быть равно 0.

leading_non_ref_pic_flag, равное 1, определяет, что текущее изображение – это неконтрольное изображение, предшествующее любому контрольному изображению в порядке декодирования в субпоследовательности, или что субпоследовательность содержит неконтрольные изображения. Если текущее изображение – это контрольное изображение или если текущее изображение – это неконтрольное изображение, следующее, по крайней мере, за одним контрольным изображением в порядке декодирования в субпоследовательности, то **leading_non_ref_pic_flag** должно быть равно 0.

last_pic_flag, равное 1, указывает, что текущее изображение – это не последнее изображение субпоследовательности (в порядке декодирования), включая все контрольные и неконтрольные изображения субпоследовательности. Если текущее изображение – это не последнее изображение субпоследовательности (в порядке декодирования), **last_pic_flag** должно быть равно 0.

Текущее изображение присваивают субпоследовательности следующим образом.

- Если одно или более условий – истина, то текущее изображение – это первое изображение субпоследовательности в порядке декодирования:
 - отсутствует более раннее изображение в порядке декодирования, отмеченное таким же значением **sub_seq_id** и **sub_seq_layer_num**, что и в текущем изображении;
 - значение **leading_non_ref_pic_flag** равно 1, а значение **leading_non_ref_pic_flag** равно 0 в предыдущем изображении в порядке декодирования с теми же значениями **sub_seq_id** и **sub_seq_layer_num**, что и в текущем изображении;
 - значение **first_ref_pic_flag** равно 1, а значение **leading_non_ref_pic_flag** равно 0 в предыдущем изображении в порядке декодирования с теми же значениями **sub_seq_id** и **sub_seq_layer_num**, что и в текущем изображении;
 - значение **last_pic_flag** равно 1 в предыдущем изображении в порядке декодирования с теми же значениями **sub_seq_id** и **sub_seq_layer_num**, что и в текущем изображении.
- Иначе текущее изображение принадлежит к той же субпоследовательности, что и предыдущее изображение в порядке декодирования с теми же значениями **sub_seq_id** и **sub_seq_layer_num**, что и в текущем изображении.

sub_seq_frame_num_flag, равное 0, указывает, что **sub_seq_frame_num** отсутствует. Значение **sub_seq_frame_num_flag**, равное 1, указывает, что **sub_seq_frame_num** присутствует.

sub_seq_frame_num должно быть равно 0 для первого контрольного изображения субпоследовательности и для любого неконтрольного изображения, предшествующего первому контрольному изображению субпоследовательности в порядке декодирования. Значение **sub_seq_frame_num** ограничено следующим образом.

- Если текущее изображение – это не второе поле пары дополнительных полей, то **sub_seq_frame_num** должно быть увеличено на 1 в операции по модулю **MaxFrameNum** относительно предыдущего контрольного изображения в порядке декодирования, которое принадлежит к этой субпоследовательности.
- Иначе (текущее изображение – это второе поле пары дополнительных полей) значение **sub_seq_frame_num** должно быть таким же, как значение **sub_seq_frame_num** первого поля пары дополнительных полей.

sub_seq_frame_num должно быть в диапазоне от 0 до **MaxFrameNum** – 1 включительно.

Если текущее изображение – это IDR изображение, то в слое субпоследовательностей 0 должна быть запущена новая субпоследовательность. Таким образом, **sub_seq_layer_num** должно быть равно 0, идентификатор **sub_seq_id** должен отличаться от его значения в предыдущей субпоследовательности слоя 0, **first_ref_pic_flag** должно быть равно 1, а **leading_non_ref_pic_flag** должно быть равно 0.

Если сообщение SEI об информации субпоследовательности присутствует для обоих кодированных полей пары дополнительных полей, то значения **sub_seq_layer_num**, **sub_seq_id**, **leading_non_ref_pic_flag** и **sub_seq_frame_num** (если присутствуют) должны быть идентичны для обоих изображений.

Если сообщение SEI об информации субпоследовательности присутствует только для одного кодированного поля пары дополнительных полей, то значения **sub_seq_layer_num**, **sub_seq_id**, **leading_non_ref_pic_flag** и **sub_seq_frame_num** (если присутствуют) должны быть также применимы к другому кодированному полю пары дополнительных полей.

D.2.12 Семантика сообщения SEI о характеристиках слоя субпоследовательности

Сообщение SEI о характеристиках слоя субпоследовательности определяет характеристики слоя субпоследовательности.

num_sub_seq_layers_minus1 плюс 1 определяет номер слоя субпоследовательности в данной последовательности. Значение **num_sub_seq_layers_minus1** должно быть в диапазоне от 0 до 255 включительно.

Пара значений **average_bit_rate** и **average_frame_rate** характеризует каждый слой субпоследовательности. Первая пара **average_bit_rate** и **average_frame_rate** определяет характеристики слоя субпоследовательности 0. Если присутствует, то вторая пара определяет совместно характеристики слоя субпоследовательности 0 и 1. Каждая пара в порядке декодирования определяет характеристики диапазона слоя субпоследовательности от номера слоя 0 до номера слоя, определенного циклом счетчика. Эти значения вступают в силу от пункта, где они были декодированы, и до обновления этих декодированных значений.

accurate_statistics_flag, равное 1, указывает, что значения **average_bit_rate** и **average_frame_rate** округлены до статистически корректных значений. Значение **accurate_statistics_flag**, равное 0, указывает, что **average_bit_rate** и **average_frame_rate** – это оценки, которые могут отклоняться от точных значений.

Если **accurate_statistics_flag** равно 0, качество аппроксимации, использованное при вычислении значений **average_bit_rate** и **average_frame_rate**, выбрано процессом кодирования и не определено в настоящей Рекомендации | Международном стандарте.

average_bit_rate указывает среднюю скорость битов в блоках 1000 бит/с. При вычислениях принимают во внимание все блоки NAL в диапазоне слоев определенных выше субпоследовательностей. Общую среднюю скорость битов находят в соответствии со временем удаления блока доступа, определенного в Приложении С к настоящей Рекомендации | Международному стандарту. Далее величина **bTotal** – это число битов во всех блоках NAL, следующих за сообщением SEI о характеристиках слоя субпоследовательности (включая биты блоков NAL текущего блока доступа) и предшествующих следующему блоку доступа (в порядке декодирования), включая сообщение SEI о характеристиках слоя субпоследовательности (если такое имеется), или (иначе) – это конец потока. Величина t_1 – это время удаления (в секундах) текущего блока доступа, а t_2 – это время удаления (в секундах) самого последнего блока доступа (в порядке декодирования) перед следующим сообщением SEI о характеристиках слоя субпоследовательности (если такое имеется), или (иначе) – это конец потока.

Если **accurate_statistics_flag** равно 1, должны быть выполнены следующие условия.

- Если t_1 не равно t_2 , следующее условие должно быть истиной:

$$\text{average_bit_rate} == \text{Round}(\text{bTotal} \div ((t_2 - t_1) * 1000)). \quad (\text{D-6})$$

- Иначе (t_1 равно t_2) следующее условие должно быть истиной:

$$\text{average_bit_rate} == 0. \quad (\text{D-7})$$

average_frame_rate указывает среднюю частоту кадров в блоках кадров/(256 секунд). При вычислениях принимают во внимание все блоки NAL в диапазоне слоев определенных выше субпоследовательностей. Далее величина **fTotal** – это число кадров, пар дополнительных полей и непарных полей между текущим изображением (включительно) и следующим сообщением SEI о характеристиках слоя субпоследовательности (если такое имеется), или (иначе) – это конец потока. Величина t_1 – это время удаления (в секундах) текущего блока доступа, а t_2 – это время удаления (в секундах) самого последнего блока доступа (в порядке декодирования) перед следующим сообщением SEI о характеристиках слоя субпоследовательности (если такое имеется), или (иначе) – это конец потока.

Если **accurate_statistics_flag** равно 1, должны быть выполнены следующие условия.

- Если t_1 не равно t_2 , следующее условие должно быть истиной:

$$\text{average_frame_rate} == \text{Round}(\text{fTotal} * 256 \div (t_2 - t_1)). \quad (\text{D-8})$$

- Иначе (t_1 равно t_2) следующее условие должно быть истиной:

$$\text{average_frame_rate} == 0. \quad (\text{D-9})$$

D.2.13 Семантика сообщения SEI о характеристиках субпоследовательности

Сообщение SEI о характеристиках субпоследовательности указывает на характеристики субпоследовательности. Оно также указывает на зависимости **integ** предсказания между субпоследовательностями. Это сообщение должно содержаться в первом блоке доступа в порядке декодирования субпоследовательности, к которой относится сообщение SEI. Эту субпоследовательность здесь и далее считают целевой субпоследовательностью.

sub_seq_layer_num идентифицирует номер слоя данной субпоследовательности по отношению к целевой. Значение **sub_seq_layer_num** должно быть в диапазоне от 0 до 255 включительно.

sub_seq_id идентифицирует целевую субпоследовательность. Значение **sub_seq_id** должно быть в диапазоне от 0 до 65535 включительно.

duration_flag, равное 0, указывает, что продолжительность целевой субпоследовательности не определена.

sub_seq_duration определяет продолжительность целевой субпоследовательности для тактовой частоты задающего генератора 90 кГц.

average_rate_flag, равное 0, указывает, что средняя скорость битов и средняя частота кадров целевой субпоследовательности не определены.

accurate_statistics_flag указывает, насколько надежны значения **average_bit_rate** и **average_frame_rate**. **accurate_statistics_flag**, равное 1, указывает что **average_bit_rate** и **average_frame_rate** округлены от статистически корректного значения. Значение **accurate_statistics_flag**, равное 0, указывает, что **average_bit_rate** и **average_frame_rate** – это оценки, которые могут отклоняться от точных значений.

average_bit_rate указывает среднюю скорость битов в блоках 1000 бит/с целевой субпоследовательности. При вычислениях принимают во внимание все блоки NAL целевой субпоследовательности. Среднюю скорость битов находят в соответствии с временем удаления блока доступа, определенного в п. С.1.2. Далее nB – это число битов в всех блоках NAL в субпоследовательности. t_1 – это время удаления (в секундах) текущего блока доступа субпоследовательности (в порядке декодирования), а t_2 – это время удаления (в секундах) самого последнего блока доступа субпоследовательности (в порядке декодирования).

Если **accurate_statistics_flag** равно 1, то должно быть выполнены следующие условия.

- Если t_1 не равно t_2 , то следующее условие должно быть истиной:

$$\text{average_bit_rate} == \text{Round}(nB \div ((t_2 - t_1) * 1000)). \quad (D-10)$$

- Иначе (t_1 равно t_2) следующее условие должно быть истиной:

$$\text{average_bit_rate} == 0. \quad (D-11)$$

average_frame_rate указывает среднюю частоту кадров в блоках кадров/(256 секунд) целевой субпоследовательности. При вычислениях принимают во внимание все блоки NAL целевой субпоследовательности. Среднюю частоту кадров находят в соответствии с временем удаления блока доступа, определенного в п. С.1.2. Далее fC – это число кадров, пар дополнительных полей и непарных полей в субпоследовательности. t_1 – это время удаления (в секундах) текущего блока доступа субпоследовательности (в порядке декодирования), а t_2 – это время удаления (в секундах) самого последнего блока доступа субпоследовательности (в порядке декодирования).

Если **accurate_statistics_flag** равно 1, то должны быть выполнены следующие условия.

- Если t_1 не равно t_2 , то следующее условие должно быть истиной:

$$\text{average_frame_rate} == \text{Round}(fC * 256 \div (t_2 - t_1)). \quad (D-12)$$

- Иначе (t_1 равно t_2) следующее условие должно быть истиной:

$$\text{average_frame_rate} == 0. \quad (D-13)$$

num_referenced_subseqs определяет число субпоследовательностей, которые содержат изображения, использованные как контрольные изображения для *inter* предсказания изображений целевой субпоследовательности. Значение **num_referenced_subseqs** должно быть в диапазоне от 0 до 255 включительно.

ref_sub_seq_layer_num, **ref_sub_seq_id** и **ref_sub_seq_direction** идентифицируют субпоследовательность, которая содержит изображения, использованные как контрольные изображения для *inter* предсказания изображений целевой субпоследовательности. В зависимости от **ref_sub_seq_direction** применяют следующее.

- Если **ref_sub_seq_direction** равно 0, множество возможных субпоследовательностей состоит из таких субпоследовательностей, у которых **sub_seq_id** равно **ref_sub_seq_id** и которые расположены в слое субпоследовательностей с **sub_seq_layer_num** равным **ref_sub_seq_layer_num**, а первое изображение в порядке декодирования предшествует первому изображению целевой субпоследовательности в порядке декодирования.
- Иначе (**ref_sub_seq_direction** равно 1) множество возможных субпоследовательностей состоит из таких субпоследовательностей, у которых **sub_seq_id** равно **ref_sub_seq_id** и которые расположены в слое субпоследовательностей с **sub_seq_layer_num** равным **ref_sub_seq_layer_num**, а первое изображение в порядке декодирования следует за первым изображением целевой субпоследовательности в порядке декодирования.

Субпоследовательность, использованная как контрольная для целевой субпоследовательности, является одним из кандидатов, у которых первое изображение в порядке декодирования будет ближайшим к первому изображению целевой субпоследовательности.

D.2.14 Семантика сообщения SEI о полностью фиксированном кадре

Сообщение SEI о полностью фиксированном кадре указывает, что текущее изображение и любые последующие изображения в порядке выхода, которые отвечают определенным условиям, не должны затрагивать содержание изображения. В любом блоке доступа должно присутствовать не более одного сообщения SEI о полностью фиксированном кадре.

full_frame_freeze_repetition_period определяет длительность хранения сообщения SEI о полностью фиксированном кадре и может определять интервал подсчета порядка изображений, внутри которого в потоке битов должно присутствовать другое сообщение SEI о полностью фиксированном кадре или сообщение SEI об отмене режима полностью фиксированного кадра или конец кодированной видеопоследовательности. Значение **full_frame_freeze_repetition_period** должно быть в диапазоне от 0 до 16 384 включительно.

full_frame_freeze_repetition_period, равное 0, определяет, что сообщение SEI о полностью фиксированном кадре применимо только к текущему декодированному изображению.

full_frame_freeze_repetition_period, равное 1, определяет, что сообщение SEI о полностью фиксированном кадре повторяют в порядке выхода до тех пор, пока любое из следующих условий является истиной:

- начинается новая кодированная видеопоследовательность;
- изображение в блоке доступа, в котором содержится сообщение SEI о полностью фиксированном кадре или сообщение SEI об отмене режима полностью фиксированного кадра, является выходом со значением $\text{PicOrderCnt}()$ большим, чем $\text{PicOrderCnt}(\text{CurrPic})$.

full_frame_freeze_repetition_period больше 1 определяет, что сообщение SEI о полностью фиксированном кадре повторяют, пока любые из следующих условий являются истиной:

- начинается новая кодированная видеопоследовательность;
- изображение в блоке доступа, в котором содержится сообщение SEI о полностью фиксированном кадре или об отмене режима полностью фиксированного кадра, является выходом, значение $\text{PicOrderCnt}()$ которого больше $\text{PicOrderCnt}(\text{CurrPic})$ и меньше или равно $\text{PicOrderCnt}(\text{CurrPic}) + \text{full_frame_freeze_repetition_period}$.

full_frame_freeze_repetition_period больше 1 указывает, что другое сообщение SEI о полностью фиксированном кадре или об отмене режима полностью фиксированного кадра должно присутствовать в блоке доступа для изображения, которое является выходом, значение $\text{PicOrderCnt}()$ которого больше $\text{PicOrderCnt}(\text{CurrPic})$ и меньше или равно $\text{PicOrderCnt}(\text{CurrPic}) + \text{full_frame_freeze_repetition_period}$, если не заканчивается поток битов или не начинается новая кодированная видеопоследовательность без такого изображения.

D.2.15 Семантика сообщения SEI об отмене режима полностью фиксированного кадра

Сообщение SEI о полностью фиксированном кадре отменяет действие любого сообщения SEI о полностью фиксированном кадре, которое было отправлено с изображениями, предшествующими текущему изображению в порядке выхода. Сообщение SEI об отмене режима полностью фиксированного кадра указывает, что текущее изображения и последующее изображения в порядке выхода должны воздействовать на содержимое изображения. В любом блоке доступа должно присутствовать не более одного сообщения SEI об отмене режима полностью фиксированного кадра. Сообщение SEI об отмене режима полностью фиксированного кадра не должно быть представлено в блоке доступа, содержащем сообщение SEI о полностью фиксированном кадре. Если сообщение SEI о полностью фиксированном кадре присутствует в блоке доступа, содержащем поле дополнительной пары полей, где значения $\text{PicOrderCnt}(\text{CurrPic})$ для двух полей дополнительной пары полей равны между собой, сообщение SEI об отмене режима полностью фиксированного кадра не должно присутствовать ни в одном из этих двух блоков доступа.

D.2.16 Семантика сообщения SEI о стоп-кадре

Сообщение SEI о стоп-кадре указывает, что текущий кадр отмечен для использования, как это определено применением, в качестве стоп-кадра неподвижного видеоизображения.

snapshot_id определяет идентификационный номер стоп-кадра. Значение **snapshot_id** должно быть в диапазоне от 0 до $2^{32} - 1$ включительно.

Значения **snapshot_id** в диапазоне от 0 до 255 включительно и в диапазоне от 512 до $2^{31} - 1$ включительно может быть использовано, как определено приложением. Значения **snapshot_id** в диапазоне от 256 до 511 включительно и в диапазоне от 2^{31} до $2^{32} - 1$ включительно зарезервированы для будущего использования МСЭ-Т | ИСО/МЭК. Декодеры, учитывающие значение **snapshot_id** в диапазоне от 256 до 511 включительно или в диапазоне от 2^{31} до $2^{32} - 1$ включительно, должны игнорировать **snapshot_id** (удалять из потока битов и отбрасывать).

D.2.17 Семантика сообщения SEI о запуске сегмента последовательного улучшения

Сообщение SEI о запуске сегмента последовательного улучшения определяет начало ряда следующих друг за другом кодированных изображений, помеченных как текущее изображение, за которыми идет последовательность одного или более изображений, характерных в большей мере улучшением качества текущего изображения, чем представлением непрерывной сцены движения.

Помеченный набор следующих друг за другом кодированных изображений должен продолжаться, пока одно из следующих условий является истиной. Если одно из нижеперечисленных условий становится истиной, то следующая секция, предназначенная для декодирования, больше не принадлежит помеченному набору следующих друг за другом кодированных изображений.

1. Следующая секция, предназначенная для декодирования, принадлежит IDR изображению.
2. Значение `num_refinement_steps_minus1` больше 0, а значение `frame_num` следующей предназначенной для декодирования секции – это $(\text{currFrameNum} + \text{num_refinement_steps_minus1} + 1) \% \text{MaxFrameNum}$, где `currFrameNum` – значение `frame_num` изображения в блоке доступа, содержащем сообщение SEI.
3. Значение `num_refinement_steps_minus1` равно 0, и декодирован конец сообщения SEI о запуске сегмента последовательного улучшения с тем же значением `progressive_refinement_id`, что и в этом сообщении SEI.

Порядок декодирования изображений в помеченном наборе следующих друг за другом изображений должен быть таким же, как порядок на выходе. Значение **progressive_refinement_id** указывает на идентификационный номер последовательной операции улучшения. Значение `progressive_refinement_id` должно быть в диапазоне от 0 до $2^{32} - 1$ включительно.

Значения `progressive_refinement_id` в диапазоне от 0 до 255 включительно и в диапазоне от 512 до $2^{31} - 1$ включительно могут быть использованы, как это определено приложением. Значения `progressive_refinement_id` в диапазоне от 256 до 511 включительно и в диапазоне от 2^{31} до $2^{32} - 1$ включительно зарезервированы для будущего использования МСЭ-Т | ИСО/МЭК. Декодеры, учитывающие значение `progressive_refinement_id` в диапазоне от 256 до 511 включительно или в диапазоне от 2^{31} до $2^{32} - 1$ включительно, должны игнорировать `progressive_refinement_id` (удалять из потока битов и отбрасывать).

num_refinement_steps_minus1 определяет число контрольных кадров в помеченном наборе следующих друг за другом кодированных изображений следующим образом.

- Если `num_refinement_steps_minus1` равно 0, число контрольных кадров в помеченном наборе следующих друг за другом кодированных изображений неизвестно.
- Иначе число контрольных кадров в помеченном наборе следующих друг за другом кодированных изображений равно `num_refinement_steps_minus1 + 1`.

`num_refinement_steps_minus1` должно быть в диапазоне от 0 до `MaxFrameNum - 1` включительно.

D.2.18 Семантика сообщения SEI об окончании сегмента последовательного улучшения

Сообщение SEI об окончании сегмента последовательного улучшения заканчивает набор следующих друг за другом кодированных изображений, которые были помечены для использования сообщения SEI о запуске сегмента последовательного улучшения как исходного изображения, за которым поступала последовательность одного или более изображений с улучшением качества исходного изображения. Этим заканчивается текущее изображение.

progressive_refinement_id определяет идентификационный номер операции последовательного улучшения. Значение `progressive_refinement_id` должно быть в диапазоне от 0 до $2^{32} - 1$ включительно.

Сообщение SEI об окончании сегмента последовательного улучшения определяет конец любого сегмента последовательного улучшения, первоначально инициированного сообщением SEI о запуске, с тем же значением `progressive_refinement_id`.

Значения `progressive_refinement_id` в диапазоне от 0 до 255 включительно и в диапазоне от 512 до $2^{31} - 1$ включительно могут быть использованы, как это определено приложением. Значения `progressive_refinement_id` в диапазоне от 256 до 511 включительно и в диапазоне от 2^{31} до $2^{32} - 1$ включительно зарезервированы для будущего использования МСЭ-Т | ИСО/МЭК. Декодеры, учитывающие значение `progressive_refinement_id` в диапазоне от 256 до 511 включительно или в диапазоне от 2^{31} до $2^{32} - 1$ включительно, должны игнорировать `progressive_refinement_id` (удалять из потока битов и отбрасывать).

D.2.19 Семантика сообщения SEI о наборе группы секций ограниченного движения

Это сообщение SEI указывает, что `inter` предсказание в границах группы секции ограничено, как определено ниже. Если таковое присутствует, это сообщение должно появляться только там, где это связано с блоком доступа IDR, как это определено в п. 7.4.1.2.3.

Набор целевых изображений для этого сообщения SEI содержит все последовательные первично кодированные изображения в порядке декодирования, начиная со связанного с ними первично кодированного IDR изображения (включительно) и заканчивая следующим первично кодированным IDR изображением (исключая) или самым последним первично кодированным изображением в потоке битов (включительно) в порядке декодирования, если за этим не следует первично кодированное IDR изображение. Набор группы секций – это собрание одной или более групп секций, идентифицированных значением элемента синтаксиса `slice_group_id[i]`.

Это сообщение SEI указывает, что для каждого изображения в наборе целевых изображений процесс `inter` предсказания ограничен следующим образом. Отсутствует значение образца вне набора группы секций и отсутствует значение образца в местоположении фрагмента образца, которые находят с помощью одного или более значений образца вне набора группы секций, использованных для `inter` предсказания любого образца в этой группе секций.

num_slice_groups_in_set_minus1 + 1 определяет число групп секций в наборе из групп секций. Разрешенный диапазон `num_slice_groups_in_set_minus1` от 0 до `num_slice_groups_minus1` включительно. Разрешенный диапазон `num_slice_groups_minus1` определен в Приложении А.

slice_group_id[i] идентифицирует группу(ы) секций, которые находятся в наборе групп секций. Разрешенный диапазон от 0 до num_slice_groups_in_set_minus1 включительно. Размер элемента синтаксиса slice_group_id[i] равен $\text{Ceil}(\text{Log}_2(\text{num_slice_groups_minus1} + 1))$ битов.

exact_sample_value_match_flag, равное 0, указывает, что в наборе целевых изображений, когда макроблоки, не принадлежащие к этому набору групп секций еще не декодированы, значение каждого образца в группе секций не должно быть точно таким же, как значение этого же образца в случае, когда все макроблоки декодированы. Значение exact_sample_value_match_flag, равное 1, указывает, что в наборе целевых изображений, когда макроблоки, не принадлежащие к этому набору групп секций, еще не декодированы, значение каждого образца в группе секций должно быть точно таким же, как значение этого же образца в случае, когда все макроблоки декодированы.

ПРИМЕЧАНИЕ 1. – Если disable_deblocking_filter_idc равно 2 во всех секциях набора целевых изображений, то exact_sample_value_match_flag должно быть 1.

pan_scan_rect_flag, равное 0, указывает, что значение pan_scan_rect_id отсутствует. Значение pan_scan_rect_flag, равное 1, указывает, что значение pan_scan_rect_id присутствует.

pan_scan_rect_id указывает, что определенный набор группы секций, по меньшей мере, покрывает прямоугольник сканирования, идентифицированный значением pan_scan_rect_id в наборе целевых изображений.

ПРИМЕЧАНИЕ 2. – Множество сообщений SEI motion_constrained_slice_group_set может быть связано с тем же самым IDR изображением. Следовательно, в наборе целевых изображений активной может быть более одной группы секций.

ПРИМЕЧАНИЕ 3. – В наборе целевых изображений размер, форма и расположение группы секций могут изменяться.

D.2.20 Семантика сообщения SEI о характеристиках зернистости пленки

Это сообщение SEI предоставляет кодеру параметризованную модель для синтеза зернистости пленки. Например, кодер может использовать сообщение SEI о характеристиках зернистости пленки для описания зернистости пленки, которая присутствовала в исходном видеоматериале, но была удалена средствами фильтрации предварительной обработки. Синтез имитируемой зернистости пленки в декодированных изображениях для процесса отображения имеет необязательный характер и не влияет на процесс декодирования, описанный в настоящей Рекомендации | Международном стандарте. Если синтез имитируемой зернистости пленки в декодированных изображениях для процесса отображения выполняется, не требуется, чтобы метод, с помощью которого осуществляется синтез, был аналогичным параметризованной модели зернистости пленки, которая предоставляется в сообщении SEI о характеристиках зернистости пленки.

ПРИМЕЧАНИЕ 1. – Процесс отображения в настоящей Рекомендации | Международном стандарте не описывается.

film_grain_characteristics_cancel_flag, равное 1, указывает, что данное сообщение SEI отменяет сохранение любого предыдущего сообщения SEI о характеристиках зернистости пленки в порядке выхода. film_grain_characteristics_cancel_flag, равное 0, указывает, что далее следует информация для моделирования зернистости пленки.

model_id указывает модель имитации зернистости пленки согласно таблице D-5. Значение model_id должно находиться в диапазоне от 0 до 1 включительно.

Таблица D-5 – Значения model_id

Значение	Описание
0	Фильтрация по частоте
1	Авторегрессия
2	Зарезервировано
3	Зарезервировано

separate_colour_description_present_flag, равное 1, указывает, что в синтаксисе сообщения SEI о характеристиках зернистости пленки присутствует отдельное описание цветового пространства для характеристик зернистости пленки, определенных в сообщении SEI. separate_colour_description_present_flag, равное 0, указывает, что описание цветового пространства для характеристик зернистости пленки, определенных в сообщении SEI, такое же, что и описание для кодированной видеопоследовательности согласно п. E.2.1.

ПРИМЕЧАНИЕ 2. – Если separate_colour_description_present_flag равно 1, цветовое пространство, описываемое для характеристик зернистости пленки, определенных в сообщении SEI, может отличаться от цветового пространства, определенного для кодированного видео, описанного в п. E.2.1.

film_grain_bit_depth_luma_minus8 плюс 8 определяет глубину цвета, используемую для компонента яркости характеристик зернистости пленки, определенных в сообщении SEI. Если film_grain_bit_depth_luma_minus8 не присутствует в сообщении SEI о характеристиках зернистости пленки, значение film_grain_bit_depth_luma_minus8 должно приниматься равным bit_depth_luma_minus8.

Значение filmGrainBitDepth[0] находят следующим образом:

$$\text{filmGrainBitDepth}[0] = \text{film_grain_bit_depth_luma_minus8} + 8. \quad (\text{D-14})$$

film_grain_bit_depth_chroma_minus8 плюс 8 определяет глубину цвета, используемую для компонентов Cb и Cr характеристик зернистости пленки, указанных в сообщении SEI. Если **film_grain_bit_depth_chroma_minus8** не представлено в сообщении SEI о характеристиках зернистости пленки, значение **film_grain_bit_depth_chroma_minus8** должно приниматься равным **bit_depth_chroma_minus8**.

Значение **filmGrainBitDepth[c]** при $c = 1$ и 2 находят следующим образом:

$$\text{filmGrainBitDepth}[c] = \text{film_grain_bit_depth_chroma_minus8} + 8 \quad \text{при } c = 1, 2. \quad (\text{D-15})$$

film_grain_full_range_flag имеет семантику, аналогичную описанной в п. E.2.1 для элемента синтаксиса **video_full_range_flag**, за исключением следующего.

- **film_grain_full_range_flag** определяет цветовое пространство характеристик зернистости пленки, описанных в сообщении SEI, а не цветовое пространство, используемое для кодированной видеопоследовательности.
- Если **film_grain_full_range_flag** не присутствует в сообщении SEI о характеристиках зернистости пленки, значение **film_grain_full_range_flag** должно приниматься равным **video_full_range_flag**.

film_grain_colour primaries имеет семантику, аналогичную описанной в п. E.2.1 для элемента синтаксиса **colour primaries syntax element**, за исключением следующего.

- **film_grain_colour primaries** определяет цветовое пространство характеристик зернистости пленки, описанных в сообщении SEI, а не цветовое пространство, используемое для кодированной видеопоследовательности.
- Если **film_grain_colour primaries** не присутствует в сообщении SEI о характеристиках зернистости пленки, значение **film_grain_colour primaries** должно приниматься равным **colour primaries**.

film_grain_transfer characteristics имеет семантику, аналогичную описанной в п. E.2.1 для элемента синтаксиса **transfer characteristics syntax element**, за исключением следующего.

- **film_grain_transfer characteristics** определяет цветовое пространство характеристик зернистости пленки, описанных в сообщении SEI, а не цветовое пространство, используемое для кодированной видеопоследовательности.
- Если **film_grain_transfer characteristics** не присутствует в сообщении SEI о характеристиках зернистости пленки, значение **film_grain_transfer characteristics** должно приниматься равным **transfer characteristics**.

film_grain_matrix coefficients имеет семантику, аналогичную описанной в п. E.2.1 для элемента синтаксиса **matrix coefficients syntax element**, за исключением следующего.

- **film_grain_matrix coefficients** определяет цветовое пространство характеристик зернистости пленки, описанных в сообщении SEI, а не цветовое пространство, используемое для кодированной видеопоследовательности.
- Если **film_grain_matrix coefficients** не присутствует в сообщении SEI о характеристиках зернистости пленки, значение **film_grain_matrix coefficients** должно приниматься равным **matrix coefficients**.
- Значения, допустимые для **film_grain_matrix coefficient**, не ограничиваются значением **chroma_format_idc**.

chroma_format_idc характеристик зернистости пленки, описанных в сообщении SEI о характеристиках зернистости пленки, должно приниматься равным 3 (4:4:4).

ПРИМЕЧАНИЕ 3. – Поскольку для работы функции создания зернистости пленки, используемой процессом отображения, не требуется применение определенного метода, декодер может, если желательно, до создания зернистости пленки выполнять понижающее преобразование информации модели для цветности, с тем чтобы имитировать зернистость пленки для других форматов цветности (4:2:0 или 4:2:2), а не повышающее преобразование декодированного видео (используя метод, не описываемый в настоящей Рекомендации | Международном стандарте).

blending_mode_id указывает режим смешивания, используемый для смешивания имитируемой зернистости пленки с декодированными изображениями, согласно таблице D-6. Значение **blending_mode_id** должно находиться в диапазоне от 0 до 1 включительно.

Таблица D-6 – Значения **blending_mode_id**

Значение	Описание
0	Аддитивный
1	Мультипликативный
2	Зарезервировано
3	Зарезервировано

В зависимости от `blending_mode_id` режим смешивания описывается следующим образом.

- Если `blending_mode_id` равно 0, режим смешивания является аддитивным и описывается как:

$$I_{\text{grain}}[x, y, c] = \text{Clip3}(0, (1 \ll \text{filmGrainBitDepth}[c]) - 1, I_{\text{decoded}}[x, y, c] + G[x, y, c]). \quad (\text{D-16})$$

- Иначе (`blending_mode_id` не равно 1) режим смешивания является мультипликативным и описывается как:

$$I_{\text{grain}}[x, y, c] = \text{Clip3}(0, (1 \ll \text{filmGrainBitDepth}[c]) - 1, I_{\text{decoded}}[x, y, c] * (1 + G[x, y, c])), \quad (\text{D-17})$$

где $I_{\text{decoded}}[x, y, c]$ представляет значение образца с координатами x, y компонента цвета с декодированного изображения I_{decoded} , $G[x, y, c]$ и является значением имитируемой зернистости пленки в той же позиции и того же компонента цвета, а $\text{filmGrainBitDepth}[c]$ – это количество битов, используемых для каждого образца в беззнаковом бинарном представлении фиксированной длины массива $I_{\text{grain}}[x, y, c]$.

log2_scale_factor определяет масштабный коэффициент, используемый в уравнениях характеристики зернистости пленки.

comp_model_present_flag[c], равное 0, показывает, что зернистость пленки не моделируется в c -ом компоненте цвета, где c , равное 0, относится к компоненту яркости, c , равное 1, относится к компоненту C_b , а c , равное 2, относится к компоненту C_r . **comp_model_present_flag[c]**, равное 1, показывает, что элементы синтаксиса, описывающие моделирование зернистости пленки в компоненте цвета c , представлены в сообщении SEI.

num_intensity_intervals_minus1[c] плюс 1 определяет число интервалов интенсивности, для которых оценивался определенный набор значений модели.

ПРИМЕЧАНИЕ 4. – Интервалы интенсивности могут перекрываться для имитирования зернистости пленки нескольких поколений.

num_model_values_minus1[c] плюс 1 определяет количество значений модели, представленных для каждого интервала интенсивности, в котором моделировалась зернистость пленки. Значение **num_model_values_minus1[c]** должно находиться в интервале от 0 до 5 включительно.

intensity_interval_lower_bound[c][i] определяет низшую границу интервала i уровней интенсивности, к которым применяется набор значений модели.

intensity_interval_upper_bound[c][i] определяет высшую границу интервала i уровней интенсивности, к которым применяется набор значений модели.

В зависимости от `model_id` выбор наборов значений модели описывается следующим образом.

- Если `model_id` равно 0, для выбора наборов значений модели с индексом $s[j]$, которые применяются ко всем образцам в данном блоке, используется среднее значение каждого блока b образцов 16×16 в I_{decoded} , обозначаемое b_{avg} :

```
for( i = 0, j = 0; i <= num_intensity_intervals_minus1; i++ )
    if( b_avg >= intensity_interval_lower_bound[ c ][ i ] && b_avg <= intensity_interval_upper_bound[ c ][ i ] ) {
        s[ j ] = i
        j++
    }. \quad (\text{D-18})
```

- Иначе (`model_id` равно 1) наборы значений модели, используемые для создания зернистости пленки, выбираются для каждого значения образца в I_{decoded} следующим образом:

```
for( i = 0, j = 0; i <= num_intensity_intervals_minus1; i++ )
    if( I_decoded[ x, y, c ] >= intensity_interval_lower_bound[ c ][ i ] &&
        I_decoded[ x, y, c ] <= intensity_interval_upper_bound[ c ][ i ] ) {
        s[ j ] = i
        j++
    }. \quad (\text{D-19})
```

Образцы, не попавшие в какой-либо определенный интервал, функцией создания зернистости не изменяются. Образцы, которые попадают в несколько интервалов, вызывают создание нескольких поколений зернистости. Создание нескольких поколений зернистости возникает при добавлении зернистости, рассчитанной независимо для каждого интервала интенсивности.

comp_model_value[c][i][j] представляет каждое из значений модели, присутствующее для компонента цвета c и интервала интенсивности i. Набор значений модели имеет различное значение, определяемое значением model_id. Значение comp_model_value[c][i][j] должно ограничиваться следующим образом, а также возможно дополнительное ограничение, определенное в иной части настоящего пункта.

- Если model_id равно 0, comp_model_value[c][i][j] должно находиться в диапазоне от 0 до $2^{\text{filmGrainBitDepth}[c]-1}$ включительно.
- Иначе (model_id равно 1) comp_model_value[c][i][j] должно находиться в диапазоне от $-2^{(\text{filmGrainBitDepth}[c]-1)}$ до $2^{(\text{filmGrainBitDepth}[c]-1)} - 1$ включительно.

В зависимости от model_id, синтез зернистости пленки моделируется следующим образом.

- Если model_id равно 0, модель фильтрации по частоте разрешает имитирование исходной зернистости пленки для $c = 0..2$, $x = 0..PicWidthInSamples_L$ и $y = 0..PicHeightInSamples_L$ в соответствии с:

$$G[x, y, c] = (\text{comp_model_value}[c][s][0] * Q[x, y, c] + \text{comp_model_value}[c][s][5] * G[x, y, c-1]) \gg \log_2_scale_factor, \quad (D-20)$$

где $Q[c]$ – двумерный вероятностный процесс, генерируемый фильтрацией блоков 16×16 gaussRv, с имеющими случайное значение элементами gaussRv_{ij}, которые генерируются с нормированным гауссовым распределением (независимые и одинаково распределенные гауссовы случайные величины с нулевым средним и единичной дисперсией), и где значение элемента $G[x, y, c-1]$, находящегося в правой части этого выражения, принимается равным 0, если c-1 меньше 0.

ПРИМЕЧАНИЕ 5. – Нормированная гауссова переменная может генерироваться из двух независимых равномерно распределенных случайных значений в интервале от 0 до 1 (и не равных 0), обозначенных как uRv₀ и uRv₁, используя преобразование Бокса-Мюллера, в соответствии со следующим выражением:

$$\text{gaussRv}_{ij} = \sqrt{-2 * \text{Ln}(uRv_0)} * \text{Cos}(2 * \pi * uRv_1), \quad (D-21)$$

где $\text{Ln}(x)$ – натуральный логарифм x (логарифм по основанию e, где e – константа основания натурального логарифма 2,718 281 828...), $\text{Cos}(x)$ – тригонометрическая косинусоидная функция, работающая с аргументом x, измеряемым в радианах, а π – константа Архимеда 3,141 592 653....

Полосовая фильтрация блоков gaussRv может выполняться в области дискретного косинусного преобразования (DCT) следующим образом:

```
for( y = 0; y < 16; y++ )
  for( x = 0; x < 16; x++ )
    if( ( x < comp_model_value[ c ][ s ][ 3 ] && y < comp_model_value[ c ][ s ][ 4 ] ) ||
        x > comp_model_value[ c ][ s ][ 1 ] || y > comp_model_value[ c ][ s ][ 2 ] )
      gaussRv[ x, y ] = 0
filteredRv = IDCT16x16( gaussRv ),
```

(D-22)

где IDCT16x16(z) обозначает обратное дискретное косинусное преобразование (IDCT), работающее с аргументом-матрицей 16×16 (z), как описано ниже:

$$\text{IDCT16x16}(z) = r * z * r^T, \quad (D-23)$$

где индекс T указывает транспозицию матрицы, а r – матрица 16×16 с элементами r_{ij}, описываемыми следующим образом:

$$r_{ij} = \frac{((i == 0) ? 1 : \sqrt{2})}{4} \text{Cos}\left(\frac{i * (2 * j + 1) * \pi}{32}\right), \quad (D-24)$$

где $\text{Cos}(x)$ – тригонометрическая косинусоидная функция, работающая с аргументом x, измеряемым в радианах, а π – константа Архимеда 3,141 592 653.

$Q[c]$ формируется блоками, прошедшими частотную фильтрацию filteredRv.

ПРИМЕЧАНИЕ 6. – Кодированные значения модели базируются на блоках 16×16 , но конкретный декодер может использовать другие размеры блоков. Например, декодеры, реализующие IDCT на блоках 8×8 , должны выполнять понижающее преобразование с коэффициентом 2 набора кодированных значений модели comp_model_value[c][s][i] для i, равном 1..4.

ПРИМЕЧАНИЕ 7. – Для уменьшения степени видимых блоков, которая может быть результатом построения мозаики из прошедших частотную фильтрацию блоков filteredRv, декодеры могут применять фильтр нижних частот на границах между прошедшими частотную фильтрацию блоками.

- Иначе (`model_id` равно 1) модель авторегрессии разрешает имитацию исходной зернистости пленки для $c = 0..2$, $x = 0..PicWidthInSamples_L$ и $y = 0..PicHeightInSamples_L$, как описано ниже.

$$\begin{aligned}
 G[x, y, c] = & (comp_model_value[c][s][0] * n[x, y, c] + \\
 & comp_model_value[c][s][1] * (G[x-1, y, c] + ((comp_model_value[c][s][4] * G[x, y-1, c]) >> \\
 & \log_2_scale_factor)) + \\
 & comp_model_value[c][s][3] * (((comp_model_value[c][s][4] * G[x-1, y-1, c]) >> \\
 & \log_2_scale_factor) + G[x+1, y-1, c]) + \\
 & comp_model_value[c][s][5] * (G[x-2, y, c] + \\
 & ((comp_model_value[c][s][4] * comp_model_value[c][s][4] * G[x, y-2, c]) >> \\
 & (2 * \log_2_scale_factor))) + \\
 & comp_model_value[c][s][2] * G[x, y, c-1]) >> \log_2_scale_factor,
 \end{aligned} \tag{D-25}$$

где $n[x, y, c]$ – случайное значение с нормированным гауссовым распределением (независимые и одинаково распределенные гауссовы случайные величины с нулевым средним и единичной дисперсией), и где значение элемента $G[x, y, c]$, находящегося в правой части этого выражения, принимается равным 0, если следующие условия являются истиной:

- x меньше 0;
- y меньше 0;
- x больше или равно `PicWidthInSamples_L`;
- c меньше 0.

`comp_model_value[c][i][0]` предоставляет первое значение модели для модели, определяемой `model_id`. Значение `comp_model_value[c][i][0]` соответствует стандартному отклонению составляющей гауссова шума в функциях генерации, описываемых выражениями D-20–D-23.

`comp_model_value[c][i][1]` предоставляет второе значение модели для модели, определяемой `model_id`. Значение `comp_model_value[c][i][1]` должно быть больше или равно 0 и меньше 16.

Значение не представленного в сообщении SEI о характеристиках зернистости пленки элемента `comp_model_value[c][i][1]` должно приниматься следующим образом.

- Если `model_id i` равно 0, `comp_model_value[c][i][1]` должно приниматься равным 8.
- Иначе (`model_id` равно 1) `comp_model_value[c][i][1]` должно приниматься равным 0.

`comp_model_value[c][i][1]` интерпретируется следующим образом.

- Если `model_id` равно 0, `comp_model_value[c][i][1]` указывает горизонтальную верхнюю частоту среза, которая должна использоваться для фильтрации DCT случайных значений блока 16x16.
- Иначе (`model_id` равно 1) `comp_model_value[c][i][1]` указывает пространственную корреляцию первого порядка для смежных образцов $(x-1, y)$ и $(x, y-1)$.

`comp_model_value[c][i][2]` предоставляет третье значение модели для модели, определяемой `model_id`. Значение `comp_model_value[c][i][2]` должно быть больше или равно 0 и меньше 16.

Значение не представленного в сообщении SEI о характеристиках зернистости пленки элемента `comp_model_value[c][i][2]` должно приниматься следующим образом.

- Если `model_id` равно 0, `comp_model_value[c][i][2]` должно приниматься равным `comp_model_value[c][i][1]`.
- Иначе (`model_id` равно 0) `comp_model_value[c][i][2]` должно приниматься равным 0.

`comp_model_value[c][i][2]` интерпретируется следующим образом.

- Если `model_id` равно 0, `comp_model_value[c][i][2]` указывает вертикальную верхнюю частоту среза, которая должна использоваться для фильтрации DCT случайных значений блока 16x16.
- Иначе (`model_id` равно 1) `comp_model_value[c][i][2]` указывает корреляцию цветов последовательных компонентов цвета.

`comp_model_value[c][i][3]` предоставляет четвертое значение модели для модели, определяемой `model_id`. Значение `comp_model_value[c][i][3]` должно быть больше или равно 0 и меньше или равно `comp_model_value[c][i][1]`.

Значение не представленного в сообщении SEI о характеристиках зернистости пленки элемента `comp_model_value[c][i][3]` должно приниматься равным 0.

comp_model_value[c][i][3] интерпретируется следующим образом.

- Если model_id равно 0, comp_model_value[c][i][3] указывает горизонтальную нижнюю частоту среза, которая должна использоваться для фильтрации DCT случайных значений блока 16x16.
- Иначе (model_id равно 1) comp_model_value[c][i][3] указывает пространственную корреляцию первого порядка для смежных образцов (x-1, y-1) и (x+1, y-1).

comp_model_value[c][i][4] предоставляет пятое значение модели для модели, определяемой model_id. Значение comp_model_value[c][i][4] должно быть больше или равно 0 и меньше или равно comp_model_value[c][i][2].

Значение не представленного в сообщении SEI о характеристиках зернистости пленки элемента comp_model_value[c][i][4] должно приниматься равным model_id.

comp_model_value[c][i][4] интерпретируется следующим образом.

- Если model_id равно 0, comp_model_value[c][i][4] указывает вертикальную нижнюю частоту среза, которая должна использоваться для фильтрации DCT случайных значений блока 16x16.
- Иначе (model_id равно 1) comp_model_value[c][i][4] указывает коэффициент пропорциональности имитируемой зернистости.

comp_model_value[c][i][5] предоставляет шестое значение модели для модели, определяемой model_id. Значение не представленного в сообщении SEI о характеристиках зернистости пленки элемента comp_model_value[c][i][5] должно приниматься равным 0.

comp_model_value[c][i][5] интерпретируется следующим образом.

- Если model_id равно 0, comp_model_value[c][i][5] указывает корреляцию цветов последовательных компонентов цвета.
- Иначе (model_id равно 1) comp_model_value[c][i][5] указывает пространственную корреляцию второго порядка для смежных образцов (x, y-2) и (x-2, y).

film_grain_characteristics_repetition_period определяет длительность хранения сообщения SEI о характеристиках зернистости пленки и может определять интервал подсчета порядка изображений, внутри которого в потоке битов должно присутствовать другое сообщение SEI о характеристиках зернистости пленки или конец закодированной видеопоследовательности. Значение film_grain_characteristics_repetition_period должно находиться в диапазоне от 0 до 16 384 включительно.

film_grain_characteristics_repetition_period, равное 0, определяет, что сообщение SEI о характеристиках зернистости пленки применимо только к текущему декодированному изображению.

film_grain_characteristics_repetition_period, равное 1, определяет, что сообщение SEI о характеристиках зернистости пленки сохраняется в порядке выхода, пока любое из следующих условий не станет истиной:

- начинается новая закодированная видеопоследовательность; или
- наличие в блоке доступа изображения, содержащего сообщение SEI о характеристиках зернистости пленки, которое является выходом, имеющим PicOrderCnt(), большее, чем PicOrderCnt(CurrPic).

film_grain_characteristics_repetition_period, большее 1, показывает, что сообщение SEI о характеристиках зернистости пленки сохраняется, до тех пор пока любое из следующих условий не станет истиной:

- начинается новая закодированная видеопоследовательность; или
- наличие в блоке доступа изображения, содержащего сообщение SEI о характеристиках зернистости пленки, которое является выходом, имеющим PicOrderCnt(), значение которого больше PicOrderCnt(CurrPic) и меньше или равно PicOrderCnt(CurrPic) + film_grain_characteristics_repetition_period.

film_grain_characteristics_repetition_period, большее 1, указывает, что должно быть представлено другое сообщение SEI о характеристиках зернистости пленки для изображения в блоке доступа, которое является выходом, имеющим PicOrderCnt(), большее, чем PicOrderCnt(CurrPic) и меньше или равно PicOrderCnt(CurrPic) + film_grain_characteristics_repetition_period, если не заканчивается поток битов или не начинается новая закодированная видеопоследовательность, выходом которой не является такое изображение.

D.2.21 Семантика сообщения SEI о предпочтениях отображения с учетом фильтра устранения блочности

Это сообщение SEI обеспечивает декодер информацией о предпочтениях кодера в отношении отображения каждого декодированного изображения, являющегося выходом, – либо отображение кадрированного результата процесса фильтра устранения блочности, описанного в п. 8.7, либо отображение кадрированного результата процесса построения изображения, который описан в п. 8.5.12 и выполняется до процесса фильтра устранения блочности.

ПРИМЕЧАНИЕ 1. – Процесс отображения в настоящей Рекомендации | Международном стандарте не описывается. Также в настоящей Рекомендации | Международном стандарте не определяются средства, с помощью которых кодер определяет в качестве своих предпочтений, что именно следует отображать, в сообщении SEI о предпочтениях отображения с учетом фильтра устранения блочности, а выражение предпочтений в сообщении SEI о предпочтениях отображения с учетом фильтра устранения блочности не накладывает каких-либо требований на процесс отображения.

deblocking_display_preference_cancel_flag, равное 1, указывает что сообщение SEI отменяет сохранение любого предыдущего сообщения SEI о предпочтениях отображения с учетом фильтра устранения блочности в порядке выхода. **deblocking_display_preference_cancel_flag**, равное 0, указывает, что далее следуют **display_prior_to_deblocking_preferred_flag** и **deblocking_display_preference_repetition_period**.

ПРИМЕЧАНИЕ 2. – В отсутствие сообщения SEI о предпочтениях отображения с учетом фильтра устранения блочности или после приема сообщения SEI о предпочтениях отображения с учетом фильтра устранения блочности, в котором **deblocking_display_preference_cancel_flag** равно 1, декодер должен принять, что при отображении каждого декодированного изображения, являющегося выходом, предпочтение отдается отображению кадрированного результата процесса фильтра устранения блочности, описанного в п. 8.7, перед отображением кадрированного результата процесса построения изображения, который выполняется до процесса фильтра устранения блочности и описан в п. 8.5.12.

display_prior_to_deblocking_preferred_flag, равное 1, указывает, что предпочтением кодера в отношении процесса отображения (каковой не определяется в настоящей Рекомендации | Международном стандарте) каждого изображения, которое кадрируется и является выходом, как это описано в Приложении С, является отображение кадрированного результата процесса построения изображения, который выполняется до процесса фильтра устранения блочности и описан в п. 8.5.12, а не отображение кадрированного результата процесса фильтра устранения блочности, описанного в п. 8.7. **display_prior_to_deblocking_preferred_flag**, равное 0, указывает, что предпочтением кодера в отношении процесса отображения (каковой не определяется в настоящей Рекомендации | Международном стандарте) каждого изображения, которое кадрируется и является выходом, как это описано в Приложении С, является отображение кадрированного результата процесса фильтра устранения блочности, описанного в п. 8.7, а не отображение кадрированного результата процесса построения изображения, который выполняется до процесса фильтра устранения блочности и описан в п. 8.5.12.

ПРИМЕЧАНИЕ 3. – Наличие или отсутствие сообщения SEI о предпочтениях отображения с учетом фильтра устранения блочности и значения **display_prior_to_deblocking_preferred_flag** не затрагивает требований в отношении процесса декодирования, определенного в настоящей Рекомендации | Международном стандарте. Скорее оно только обеспечивает индикацию того, когда, в дополнение к выполнению требований настоящей Рекомендации | Международного стандарта к процессу декодирования, может быть получено повышенное качество зрительного ряда путем выполнения процесса отображения (каковой не определяется в настоящей Рекомендации | Международном стандарте) одним из двух способов. В кодерах, которые используют сообщение SEI о предпочтениях отображения с учетом фильтра устранения блочности, должно быть предусмотрено знание о том, что если кодер не ограничивает свое использование емкости DPB, определенное в Приложении А в отношении используемых профиля и уровня, некоторые декодеры могут не обладать достаточной емкостью памяти для хранения результата процесса построения изображения, который выполняется до процесса фильтра устранения блочности и описан в п. 8.5.12, в дополнение к результату процесса фильтра устранения блочности, который описан в п. 8.7, при изменении порядка и задержке изображений для отображения, и, следовательно, такие декодеры не смогут использовать индикацию предпочтений. Ограничивая свое использование емкости DPB, кодер может использовать, по крайней мере, половину емкости DPB, описание которого содержится в Приложении А, позволяя таким образом декодеру использовать остальную емкость для хранения не фильтрованных изображений, которые были указаны как предпочтительные для отображения, до момента поступления этих изображений на выход.

dec_frame_buffering_constraint_flag, равное 1, указывает, что использование емкости буферизации кадров буфера декодированного изображения (DPB) HRD, определяемое элементом **max_dec_frame_buffering**, ограничено таким образом, что для кодированной видеопоследовательности не потребуется буфер декодированного изображения с количеством буферов кадров, большим, чем $\text{Max}(1, \text{max_dec_frame_buffering})$, для обеспечения выхода декодированных фильтрованных и не фильтрованных изображений, как указано сообщением SEI о предпочтениях отображения с учетом фильтра устранения блочности, значения времени выхода которых определяются элементом **dpb_output_delay** сообщений SEI о синхронизации изображений. **dec_frame_buffering_constraint_flag**, равное 0, указывает, что использование емкости буферизации кадров в HRD может быть или может не быть ограничено таким образом, на что будет указывать значение **dec_frame_buffering_constraint_flag**, равное 1.

Для целей определения ограничения, накладываемого в случае, когда **dec_frame_buffering_constraint_flag** равно 1, значение объема емкости буферизации кадров, используемого в любой данный момент времени каждым буфером DPB, который содержит изображение, должно приниматься следующим образом.

- Если в отношении буфера кадров выполняются два следующих условия, буфер кадров рассматривается как использующий для хранения емкость, равную емкости двух буферов кадров.
 - Буфер кадров содержит кадр или одно или более полей, помеченных как "используется для контроля", и
 - буфер кадров содержит изображение, в отношении которого выполняются два следующих условия:
 - значение времени выхода HRD изображения больше, чем данный момент времени; и
 - в сообщении SEI о предпочтениях отображения с учетом фильтра устранения блочности указано, что предпочтением кодера в отношении процесса отображения является отображение кадрированного результата процесса построения изображения, который выполняется до процесса фильтра устранения блочности и описан в п. 8.5.12, а не кадрированного результата процесса фильтра устранения блочности, описанного в п. 8.7.
- Иначе буфер кадров рассматривается как использующий для хранения емкость DPB одного буфера кадров.

Если `dec_frame_buffering_constraint_flag` равно 1, емкость буферизации кадров, используемая всеми буферами кадров в DPB, которые содержат полученные таким образом изображения, не должна превышать $\text{Max}(1, \text{max_dec_frame_buffering})$ в период работы HRD для декодированной последовательности.

Значение `dec_frame_buffering_constraint_flag` должно быть одинаковым во всех сообщения SEI о предпочтениях отображения с учетом фильтра устранения блочности данной кодовой последовательности.

`deblocking_display_preference_repetition_period` определяет длительность хранения сообщения SEI о характеристиках зернистости пленки и может определять интервал подсчета порядка изображений, внутри которого в потоке битов должно присутствовать другое сообщение SEI о характеристиках зернистости пленки или конец кодированной видеопоследовательности. Значение `deblocking_display_preference_repetition_period` должно находиться в интервале от 0 до 16 384 включительно.

`deblocking_display_preference_repetition_period`, равное 0, определяет, что сообщение SEI о предпочтениях отображения с учетом фильтра устранения блочности применяется только к текущему кодированному изображению.

`deblocking_display_preference_repetition_period`, равное 1, определяет, что сообщение SEI о предпочтениях отображения с учетом фильтра устранения блочности сохраняется в порядке выхода, до тех пор пока какое-либо из следующих условий не станет истиной.

- Начинается новая кодированная видеопоследовательность.
- Наличие в блоке доступа изображения, содержащего сообщение SEI о предпочтениях отображения с учетом фильтра устранения блочности, которое является выходом, имеющим `PicOrderCnt()`, значение которого больше, чем `PicOrderCnt(CurrPic)`.

`deblocking_display_preference_repetition_period` больше 1 определяет, что сообщение SEI о предпочтениях отображения с учетом фильтра устранения блочности сохраняется, до тех пор пока какое-либо из следующих условий не станет истиной.

- Начинается новая кодированная видеопоследовательность.
- Наличие в блоке доступа изображения, содержащего сообщение SEI о предпочтениях отображения с учетом фильтра устранения блочности, которое является выходом, имеющим `PicOrderCnt()`, значение которого больше `PicOrderCnt(CurrPic)` и меньше или равно `PicOrderCnt(CurrPic) + deblocking_display_preference_repetition_period`.

`deblocking_display_preference_repetition_period` больше 1 указывает, что для изображения в блоке доступа, которое является выходом, имеющим `PicOrderCnt()`, значение которого больше `PicOrderCnt(CurrPic)` и меньше или равно `PicOrderCnt(CurrPic) + deblocking_display_preference_repetition_period`, должно быть представлено другое сообщение SEI о предпочтениях отображения с учетом фильтра устранения блочности, если не заканчивается поток битов или не начинается новая кодированная видеопоследовательность, выходом которой не является такое изображение.

D.2.22 Семантика сообщения SEI об информации, касающейся стереоизображения

Это сообщение SEI обеспечивает для декодера индикацию того, что вся кодированная видеопоследовательность состоит из пар изображений, формирующих содержание стереоизображения.

Сообщение SEI об информации, касающейся стереоизображения, не должно присутствовать в любом блоке доступа кодированной видеопоследовательности, если сообщение SEI об информации, касающейся стереоизображения, не присутствует в первом блоке доступа этой кодированной видеопоследовательности.

`field_views_flag`, равное 1, указывает, что все изображения в текущей кодированной видеопоследовательности являются полями, и все поля определенной четности представляют собой левостороннее изображение, а все поля противоположной четности представляют собой правостороннее изображение стереоизображения. `field_views_flag`, равное 0, указывает, что все изображения в текущей кодированной видеопоследовательности являются кадрами, и чередующиеся кадры в порядке выхода представляют изображение для стереоизображения. Значение `field_views_flag` должно быть одинаковым во всех сообщениях SEI об информации, касающейся стереоизображения, в пределах любой кодированной последовательности.

Если сообщение SEI об информации, касающейся стереоизображения, присутствует, и `field_views_flag` равно 1, левостороннее и правостороннее изображения пары стереовидео должны кодироваться как пара дополнительных полей, время отображения первого поля этой пары полей в порядке выхода должно быть задержано в целях совпадения со временем отображения второго поля этой пары полей в порядке выхода, а пространственные местоположения образцов в каждом конкретном поле должны интерпретироваться для целей отображения как представляющие полные изображения, как это показано на рисунке 6-1, а не как пространственно разнесенные поля в пределах одного кадра, как показано на рисунке 6-2.

ПРИМЕЧАНИЕ. – Процесс отображения не определяется в настоящей Рекомендации | Международном стандарте.

`top_field_is_left_view_flag`, равное 1, указывает, что верхние поля в кодированной видеопоследовательности представляют левостороннее, а нижние поля в этой кодированной видеопоследовательности представляют правостороннее изображение. `top_field_is_left_view_flag`, равное 0, указывает, что нижние поля в данной кодированной видеопоследовательности представляют левостороннее, а верхние поля в этой кодированной видеопоследовательности представляют собой правостороннее изображение. Если элемент `top_field_is_left_view_flag` присутствует, его значение должно быть одинаковым во всех сообщениях SEI об информации, касающейся стереоизображения, в пределах любой кодированной видеопоследовательности.

current_frame_is_left_view_flag, равное 1, указывает, что текущее изображение является левосторонним изображением в паре стереоизображения. **current_frame_is_left_view_flag**, равное 0, указывает, что текущее изображение является правосторонним изображением в паре стереоизображения.

next_frame_is_second_view_flag, равное 1, указывает, что текущее изображение и следующее изображение в порядке выхода образуют пару стереоизображения и что время отображения текущего изображения должно быть задержано, с тем чтобы оно совпало со временем отображения следующего изображения в порядке выхода. **next_frame_is_second_view_flag**, равное 0, указывает, что текущее изображение и предыдущее изображение в порядке выхода образуют пару стереоизображения и что время отображения текущего изображения не должно задерживаться для спаривания изображения.

left_view_self_contained_flag, равное 1, указывает, что операции *inter* предсказания в пределах процесса декодирования левосторонних изображений данной кодированной видеопоследовательности не относятся к контрольным изображениям, которые являются правосторонними изображениями. В пределах любой кодированной видеопоследовательности значение **left_view_self_contained_flag** во всех сообщениях SEI об информации, касающейся стереоизображения, должно быть одинаковым.

right_view_self_contained_flag, равное 1, указывает, что операции *inter* предсказания в пределах процесса декодирования правосторонних изображений данной кодированной видеопоследовательности не относятся к контрольным изображениям, которые являются левосторонними изображениями. **right_view_self_contained_flag**, равное 0, указывает, что некоторые операции *inter* предсказания в рамках процесса декодирования правосторонних изображений данной кодированной видеопоследовательности могут относиться или могут не относиться к контрольным изображениям, которые являются левосторонними изображениями. В пределах любой кодированной видеопоследовательности значение **right_view_self_contained_flag** во всех сообщениях SEI об информации, касающейся стереоизображения, должно быть одинаковым.

D.2.23 Семантика зарезервированного сообщения SEI

Это сообщение состоит из данных, зарезервированных для будущих применений МСЭ-Т | ИСО/МЭК, совместимых с ранее использованными. Кодеры, соответствующие настоящей Рекомендации | Международному стандарту, не должны отправлять зарезервированные сообщения SEI до тех пор, пока использование таких сообщений не будет определено МСЭ-Т | ИСО/МЭК. Декодеры, соответствующие настоящей Рекомендации | Международному стандарту, которые сталкиваются с зарезервированными сообщениями SEI, должны отбрасывать их содержимое без какого-либо их воздействия на процесс декодирования, за исключением случаев, описанных в будущих Рекомендациях | Международных стандартах и определенных МСЭ-Т | ИСО/МЭК.

reserved_sei_message_payload_byte – это байт, зарезервированный для будущего использования МСЭ-Т | ИСО/МЭК.

Приложение Е

Информация об используемости изображения (VUI)

(Данное Приложение является неотъемлемой частью настоящей Рекомендации | Международного стандарта)

Это Приложение определяет синтаксис и семантику параметров VUI при установке параметров последовательностей.

Параметры VUI не требуются для создания образцов яркости или цветности процессом декодирования. Также не требуется соответствие декодеров настоящей Рекомендации | Международному стандарту (см. Приложение С о спецификации соответствия) для обработки этой информации в порядке выхода. Однако некоторые параметры VUI требуются для проверки на соответствие потока битов и синхронизации декодера по выходу.

В Приложении Е спецификация присутствия параметров VUI также выполняется, если эти параметры (или некоторое их подмножество) передаются декодерам (или HRD) с помощью других средств, не определяемых настоящей Рекомендацией | Международным стандартом. При наличии в потоке битов параметры VUI должны следовать синтаксису и семантике, определенным в пп. 7.3.2.1 и 7.4.2.1 и в этом Приложении. Если для некоторых приложений содержание параметров VUI передают какими-нибудь средствами, отличными от тех, которые присутствуют в потоке битов, представление содержания параметров VUI не требует использования того же синтаксиса, который определен в данном Приложении. Для целей подсчета битов пригодны только соответствующие биты, присутствующие в потоке битов.

E.1 Синтаксис VUI

E.1.1 Синтаксис параметров VUI

vui_parameters() {	C	Дескриптор
aspect_ratio_info_present_flag	0	u(1)
if(aspect_ratio_info_present_flag) {		
aspect_ratio_idc	0	u(8)
if(aspect_ratio_idc == Extended_SAR) {		
sar_width	0	u(16)
sar_height	0	u(16)
}		
}		
overscan_info_present_flag	0	u(1)
if(overscan_info_present_flag)		
overscan_appropriate_flag	0	u(1)
video_signal_type_present_flag	0	u(1)
if(video_signal_type_present_flag) {		
video_format	0	u(3)
video_full_range_flag	0	u(1)
colour_description_present_flag	0	u(1)
if(colour_description_present_flag) {		
colour_primaries	0	u(8)
transfer_characteristics	0	u(8)
matrix_coefficients	0	u(8)
}		
}		
chroma_loc_info_present_flag	0	u(1)
if(chroma_loc_info_present_flag) {		
chroma_sample_loc_type_top_field	0	ue(v)
chroma_sample_loc_type_bottom_field	0	ue(v)
}		
timing_info_present_flag	0	u(1)
if(timing_info_present_flag) {		
num_units_in_tick	0	u(32)
time_scale	0	u(32)
fixed_frame_rate_flag	0	u(1)
}		
nal_hrd_parameters_present_flag	0	u(1)
if(nal_hrd_parameters_present_flag)		
hrd_parameters()		
vcl_hrd_parameters_present_flag	0	u(1)
if(vcl_hrd_parameters_present_flag)		
hrd_parameters()		
if(nal_hrd_parameters_present_flag vcl_hrd_parameters_present_flag)		
low_delay_hrd_flag	0	u(1)
pic_struct_present_flag	0	u(1)
bitstream_restriction_flag	0	u(1)
if(bitstream_restriction_flag) {		

motion_vectors_over_pic_boundaries_flag	0	u(1)
max_bytes_per_pic_denom	0	ue(v)
max_bits_per_mb_denom	0	ue(v)
log2_max_mv_length_horizontal	0	ue(v)
log2_max_mv_length_vertical	0	ue(v)
num_reorder_frames	0	ue(v)
max_dec_frame_buffering	0	ue(v)
}		
}		

E.1.2 Синтаксис параметров HRD

hrd_parameters() {	C	Дескриптор
cpb_cnt_minus1	0	ue(v)
bit_rate_scale	0	u(4)
cpb_size_scale	0	u(4)
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {		
bit_rate_value_minus1 [SchedSelIdx]	0	ue(v)
cpb_size_value_minus1 [SchedSelIdx]	0	ue(v)
cbr_flag [SchedSelIdx]	0	u(1)
}		
initial_cpb_removal_delay_length_minus1	0	u(5)
cpb_removal_delay_length_minus1	0	u(5)
dpb_output_delay_length_minus1	0	u(5)
time_offset_length	0	u(5)
}		

E.2 Семантика VUI

E.2.1 Семантика параметров VUI

aspect_ratio_info_present_flag равно 1, определяет, что присутствует **aspect_ratio_idc**. **aspect_ratio_info_present_flag**, равно 0, определяет, что **aspect_ratio_idc** отсутствует.

aspect_ratio_idc определяет значение коэффициента пропорциональности образца к яркости образцов. В таблице E-1 показаны значение этого кода. Если **aspect_ratio_idc** указывает на **Extended_SAR**, коэффициент пропорциональности образца представляют значением **sar_width** и **sar_height**. Если элемент синтаксиса **aspect_ratio_idc** отсутствует, значение **aspect_ratio_idc** должно быть принято равным 0.

Таблица Е-1 – Значение указателя коэффициента пропорциональности образца

aspect_ratio_idc	Коэффициент пропорциональности образца	Примеры использования (для информации)
0	Неопределенный	
1	1:1 ("квадрат")	1280x720 16:9 кадр без выхода за горизонтальные пределы полезной площади экрана 1920x1080 16:9 кадр без выхода за горизонтальные пределы полезной площади экрана (кадрирован с 1920x1088) 640x480 4:3 кадр без выхода за горизонтальные пределы полезной площади экрана
2	12:11	720x576 4:3 кадр с выходом за горизонтальные пределы полезной площади экрана 352x288 4:3 кадр без выхода за горизонтальные пределы полезной площади экрана
3	10:11	720x480 4:3 кадр с выходом за горизонтальные пределы полезной площади экрана 352x240 4:3 кадр без выхода за горизонтальные пределы полезной площади экрана
4	16:11	720x576 16:9 кадр с выходом за горизонтальные пределы полезной площади экрана 528x576 4:3 кадр без выхода за горизонтальные пределы полезной площади экрана
5	40:33	720x480 16:9 кадр с выходом за горизонтальные пределы полезной площади экрана 528x480 4:3 кадр без выхода за горизонтальные пределы полезной площади экрана
6	24:11	352x576 4:3 кадр без выхода за горизонтальные пределы полезной площади экрана 480x576 16:9 кадр с выходом за горизонтальные пределы полезной площади экрана
7	20:11	352x480 4: кадр без выхода за горизонтальные пределы полезной площади экрана 480x480 16:9 кадр с выходом за горизонтальные пределы полезной площади экрана
8	32:11	352x576 16:9 кадр без выхода за горизонтальные пределы полезной площади экрана
9	80:33	352x480 16:9 кадр без выхода за горизонтальные пределы полезной площади экрана
10	18:11	480x576 4:3 кадр с выходом за горизонтальные пределы полезной площади экрана
11	15:11	480x480 4:3 кадр с выходом за горизонтальные пределы полезной площади экрана
12	64:33	528x576 16:9 кадр без выхода за горизонтальные пределы полезной площади экрана
13	160:99	528x480 16:9 кадр без выхода за горизонтальные пределы полезной площади экрана
14	4:3	1440x1080 16:9 кадр без выхода за горизонтальные пределы полезной площади экрана
15	3:2	1280x1080 16:9 кадр без выхода за горизонтальные пределы полезной площади экрана
16	2:1	960x1080 16:9 кадр без выхода за горизонтальные пределы полезной площади экрана
14..254	Зарезервировано	
255	Extended_SAR	

sar_width определяет горизонтальные размеры образца (в произвольных блоках).

sar_height определяет коэффициент пропорциональности вертикальных размеров образца (в тех же произвольных блоках, что и **sar_width**).

sar_width и **sar_height** должны быть выражены в относительных простых числах или равными 0. Если **aspect_ratio_idc** равно 0 или **sar_width** равно 0 или **sar_height** равно 0, то в настоящей Рекомендации | Международном стандарте коэффициент пропорциональности образца должен рассматриваться как неопределенный.

overscan_info_present_flag, равное 1, указывает, что присутствует **overscan_appropriate_flag**. Если **overscan_info_present_flag** равно 0 или отсутствует, то предпочтительный метод отображения видеосигнала не определен.

overscan_appropriate_flag, равное 1, указывает, что разделенные на блоки декодированные изображения на выходе пригодны для отображения при использовании кадров без выхода за пределы полезной площади экрана. Значение **overscan_appropriate_flag**, равное 0, указывает, что разделенные на блоки декодированные изображения на выходе содержат важную визуальную информацию во всей области за краями прямоугольника разделения изображения таким образом, что разделенные декодированные изображения на выходе нельзя вывести на экран с выходом за горизонтальные пределы полезной площади экрана. Вместо этого эти изображения следует выводить на экран, используя либо точное совпадение между площадями экрана и прямоугольника разделения, либо с низким качеством сканирования.

ПРИМЕЧАНИЕ 1. – Например, **overscan_appropriate_flag**, равное 1, можно использовать для развлекательных телевизионных программ или для "живого" показа людей во время видеоконференций. Значение **overscan_appropriate_flag**, равное 0, может быть используется для фиксации содержимого экрана или камеры наблюдения.

video_signal_type_present_flag, равное 1, указывает, что присутствуют значения **video_format**, **video_full_range_flag** и **colour_description_present_flag**. Значение **video_signal_type_present_flag**, равное 0, указывает, что синтаксические структуры **video_format**, **video_full_range_flag** и **colour_description_present_flag** отсутствуют.

video_format указывает на показанный в таблице E-2 формат представления изображений перед их кодированием в соответствии с настоящей Рекомендацией | Международным стандартом. Если элемент синтаксиса **video_format** не присутствует, значение **video_format** должно быть принято равным 5.

Таблица E-2 – Значение video_format

video_format	Значение
0	Компонент
1	PAL
2	NTSC
3	SECAM
4	MAC
5	Неопределенный видеоформат
6	Зарезервировано
7	Зарезервировано

video_full_range_flag определяет сигналы уровня черного и диапазона яркости и цветности как выведенные из аналоговых компонентов сигналов E'_Y , E'_{PB} и E'_{PR} или E'_R , E'_G и E'_B .

Если элемент синтаксиса **video_full_range_flag** отсутствует, значение **video_full_range_flag** должно приниматься равным 0.

colour_description_present_flag, равное 1, определяет, что **colour primaries**, **transfer characteristics** и **matrix coefficients** присутствуют. Значение **colour_description_present_flag**, равное 0, определяет, что **colour primaries**, **transfer characteristics** и **matrix coefficients** отсутствуют.

colour primaries указывает координаты цветности исходных основных цветов, показанных в таблице E-3 в терминах определения CIE 1931 параметров x и y , как это указано в ISO/CIE 10527.

Если элемент синтаксиса **colour primaries** отсутствует, то значение **colour primaries** должно быть принято равным 2 (цветность не определена или определена приложением).

Таблица Е-3 – Основные цвета

Значение	Основные цвета			Замечание для информации
0	Зарезервировано			Для будущего использования МСЭ-Т/ИСО/МЭК
1	основные цвета	x	y	Рекомендация МСЭ-R ВТ.709-5
	зеленый	0,300	0,600	
	синий	0,150	0,060	
	красный	0,640	0,330	
	белый D65	0,3127	0,3290	
2	Не определена			Характеристики изображения неизвестны или определяются применением.
3	Зарезервировано			
4	Рекомендация МСЭ-R ВТ. 470-2 Система М			Рекомендация МСЭ-R ВТ.470-2 Система М
	основные цвета	x	y	
	зеленый	0,21	0,71	
	синий	0,14	0,08	
	белый С	0,310	0,316	
5	Рекомендация МСЭ-R ВТ.470-2 Системы В, G			Рекомендация МСЭ-R ВТ.470-6 Системы В, G
	основные цвета	x	y	
	зеленый	0,29	0,60	
	синий	0,15	0,06	
	белый D65	0,3127	0,3290	
6	основные цвета	x	y	Общество инженеров кино и телевидения, 170М (1999)
	зеленый	0,310	0,595	
	синий	0,155	0,070	
	красный	0,630	0,340	
	белый D65	0,3127	0,3290	
7	основные цвета	x	y	Общество инженеров кино и телевидения, 240М (1999)
	зеленый	0,310	0,595	
	синий	0,155	0,070	
	красный	0,630	0,340	
	белый D65	0,3127	0,3290	
8	основные цвета	x	y	Обычная пленка (цветные фильтры с использованием источника света С)
	зеленый	0,243	0,692 (Wratten 58)	
	синий	0,145	0,049 (Wratten 47)	
	красный	0,681	0,319 (Wratten 25)	
	белый С	0,310	0,316	
9–255	Зарезервировано			Для будущего использования МСЭ-Т/ИСО/МЭК

transfer_characteristics указывает на оптоэлектронную характеристику передачи исходного изображения, как это определено в таблице Е-4 в виде функции от линейной оптической интенсивности на входе L_c в аналоговом диапазоне от 0 до 1.

Если элемент синтаксиса **transfer_characteristics** отсутствует, то значение **transfer_characteristics** должно быть принято равным 2 (характеристики передачи не определены или определяются приложением).

Таблица Е-4 – Характеристика передачи

Значение	Основные цвета	Замечание для информации
0	Зарезервировано	Для будущего использования МСЭ-Т/ИСО/МЭК
1	$V = 1,099 L_c^{0,45} - 0,099$ для $1 \geq L_c \geq 0,018$ $V = 4,500 L_c$ для $0,018 > L_c$	Рекомендация МСЭ-R ВТ.709
2	Не определено	Характеристики изображения неизвестны или определяются применением
3	Зарезервировано	Для будущего использования МСЭ-Т/ИСО/МЭК
4	Принимается гамма экрана 2,2	Рекомендация МСЭ-R ВТ.470-2 Система М
5	Принимается гамма экрана 2,8	Рекомендация МСЭ-R ВТ.470-2 Системы В, G
6	$V = 1,099 L_c^{0,45} - 0,099$ для $1 \geq L_c \geq 0,018$ $V = 4,500 L_c$ для $0,018 > L_c$	Общество инженеров кино и телевидения, 170М (1999 г.)
7	$V = 1,1115 L_c^{0,45} - 0,1115$ для $L_c \geq 0,0228$ $V = 4,0 L_c$ для $0,0228 > L_c$	Общество инженеров кино и телевидения, 240М (1999 г.)
8	$V = L_c$	Линейная характеристика передачи
9	$V = 1,0 - \text{Log}_{10}(L_c) \div 2$ для $1 \geq L_c \geq 0,01$ $V = 0,0$ для $0,01 > L_c$	Логарифмическая характеристика передачи (100:1 диапазона)
10	$V = 1,0 - \text{Log}_{10}(L_c) \div 2,5$ для $1 \geq L_c \geq 0,0031622777$ $V = 0,0$ для $0,0031622777 > L_c$	Логарифмическая характеристика передачи (316,22777:1 диапазона)
11–255	Зарезервировано	Для будущего использования МСЭ-Т/ИСО/МЭК

matrix_coefficients описывает матрицу коэффициентов, используемую для вывода сигналов яркости и цветности из основных тонов зеленого, синего и красного, как это определено в таблице Е-5.

matrix_coefficients не должно быть равно 0, если оба следующие условия не являются истиной:

- BitDepth_C равно BitDepth_Y;
- chroma_format_idc равно 3 (4:4:4).

Описание использования matrix_coefficients, равного 0, при всех прочих условиях зарезервировано для будущего использования МСЭ-Т/ИСО/МЭК.

matrix_coefficients не должно быть равно 8, если одно или оба следующие условия являются истиной:

- BitDepth_C равно BitDepth_Y;
- BitDepth_C равно BitDepth_Y + 1 и chroma_format_idc равно 3 (4:4:4).

Описание использования matrix_coefficients, равного 8, при всех прочих условиях зарезервировано для будущего использования МСЭ-Т/ИСО/МЭК.

Если элемент синтаксиса matrix_coefficients присутствует, значение matrix_coefficients должно приниматься равным 2.

Интерпретация matrix_coefficients определяется следующим образом.

- E'_R, E'_G и E'_B – аналоговые сигналы со значениями в диапазоне от 0 до 1.
- Белый описывается как имеющий E'_R, равное 1, E'_G, равное 1, и E'_B, равное 1.
- Черный описывается как имеющий E'_R, равное 0, E'_G, равное 0, и E'_B, равное 0.
- Если video_full_range_flag равно 0, применяются следующие уравнения.
 - Если matrix_coefficients равно 1, 4, 5, 6 или 7, применяются следующие уравнения:

$$Y = \text{Round}((1 \ll (\text{BitDepth}_Y - 8)) * (219 * E'_Y + 16)); \quad (\text{E-1})$$

$$Cb = \text{Round}((1 \ll (\text{BitDepth}_C - 8)) * (224 * E'_{PB} + 128)); \quad (\text{E-2})$$

$$Cr = \text{Round}((1 \ll (\text{BitDepth}_C - 8)) * (224 * E'_{PR} + 128)). \quad (\text{E-3})$$

- Иначе, если `matrix_coefficients` равно 0 или 8, применяются следующие уравнения:

$$R = (1 \ll (\text{BitDepth}_Y - 8)) * (219 * E'_R + 16); \quad (\text{E-4})$$

$$G = (1 \ll (\text{BitDepth}_Y - 8)) * (219 * E'_G + 16); \quad (\text{E-5})$$

$$B = (1 \ll (\text{BitDepth}_Y - 8)) * (219 * E'_B + 16). \quad (\text{E-6})$$

- Иначе, если `matrix_coefficients` равно 2, интерпретация элемента синтаксиса `matrix_coefficients` неизвестна или определяется конкретным применением.
- Иначе (`matrix_coefficients` не равно 0, 1, 2, 4, 5, 6, 7 или 8) интерпретация элемента синтаксиса `matrix_coefficients` зарезервирована для будущего определения МСЭ-Т/ИСО/МЭК.

- Иначе (`video_full_range_flag` равно 1) применяются следующие уравнения.

- Если `matrix_coefficients` равно 1, 4, 5, 6 или 7, применяются следующие уравнения:

$$Y = \text{Round}(((1 \ll \text{BitDepth}_Y) - 1) * E'_Y); \quad (\text{E-7})$$

$$Cb = \text{Round}(((1 \ll \text{BitDepth}_C) - 1) * E'_{PB} + (1 \ll (\text{BitDepth}_C - 1))); \quad (\text{E-8})$$

$$Cr = \text{Round}(((1 \ll \text{BitDepth}_C) - 1) * E'_{PR} + (1 \ll (\text{BitDepth}_C - 1))). \quad (\text{E-9})$$

- Иначе, если `matrix_coefficients` равно 0 или 8, применяются следующие уравнения:

$$R = ((1 \ll \text{BitDepth}_Y) - 1) * E'_R; \quad (\text{E-10})$$

$$G = ((1 \ll \text{BitDepth}_Y) - 1) * E'_G; \quad (\text{E-11})$$

$$B = ((1 \ll \text{BitDepth}_Y) - 1) * E'_B. \quad (\text{E-12})$$

- Иначе, если `matrix_coefficients` равно 2, интерпретация элемента синтаксиса `matrix_coefficients` неизвестна или определяется конкретным применением.
- Иначе (`matrix_coefficients` не равно 0, 1, 2, 4, 5, 6, 7 или 8) интерпретация элемента `matrix_coefficients` зарезервирована для будущего определения МСЭ-Т/ИСО/МЭК.

- Если `matrix_coefficients` не равно 0 или 8, применяются следующие уравнения:

$$E'_Y = K_R * E'_R + (1 - K_R - K_B) * E'_G + K_B * E'_B; \quad (\text{E-13})$$

$$E'_{PB} = 0.5 * (E'_B - E'_Y) \div (1 - K_B); \quad (\text{E-14})$$

$$E'_{PR} = 0.5 * (E'_R - E'_Y) \div (1 - K_R). \quad (\text{E-15})$$

ПРИМЕЧАНИЕ 2. – Тогда E'_Y – аналоговый сигнал со значениями в диапазоне от 0 до 1, E'_{PB} и E'_{PR} – аналоговые сигналы со значениями в диапазоне от -0,5 до 0,5, а белый эквивалентно задан значениями $E'_Y = 1$, $E'_{PB} = 0$, $E'_{PR} = 0$.

– Иначе, если $matrix_coefficients$ равно 0, применяются следующие уравнения:

$$Y = Round(G); \quad (E-16)$$

$$Cb = Round(B); \quad (E-17)$$

$$Cr = Round(R). \quad (E-18)$$

– Иначе ($matrix_coefficients$ равно 8) применяется следующее.

– Если $BitDepth_C$ равно $BitDepth_Y$, применяются следующие уравнения:

$$Y = Round(0.5 * G + 0.25 * (R + B)); \quad (E-19)$$

$$Cb = Round(0.5 * G - 0.25 * (R + B)); \quad (E-20)$$

$$Cr = Round(0.5 * (R - B)); \quad (E-21)$$

ПРИМЕЧАНИЕ 3. – Для обозначения Y_{CgCo} в Таблица E-5, C_b и C_r в уравнениях E-20 и E-21 могут обозначаться как C_g и C_o , соответственно. Обратное преобразование для приведенных выше четырех уравнений должно рассчитываться следующим образом.

$$t = Y - C_b; \quad (E-22)$$

$$G = Y + C_b; \quad (E-23)$$

$$B = t - C_r; \quad (E-24)$$

$$R = t + C_r. \quad (E-25)$$

– Иначе ($BitDepth_C$ не равно $BitDepth_Y$) применяются следующие уравнения:

$$Cr = Round(R) - Round(B); \quad (E-26)$$

$$t = Round(B) + (Cr \gg 1); \quad (E-27)$$

$$Cb = Round(G) - t; \quad (E-28)$$

$$Y = t + (Cb \gg 1). \quad (E-29)$$

ПРИМЕЧАНИЕ 4. – Для обозначения Y_{CgCo} в Таблица E-5, C_b и C_r в уравнениях E-28 и E-26 могут обозначаться как C_g и C_o , соответственно. Обратное преобразование для приведенных выше четырех уравнений должно рассчитываться следующим образом.

$$t = Y - (Cb \gg 1); \quad (E-30)$$

$$G = t + C_b; \quad (E-31)$$

$$B = t - (Cr \gg 1); \quad (E-32)$$

$$R = B + C_r. \quad (E-33)$$

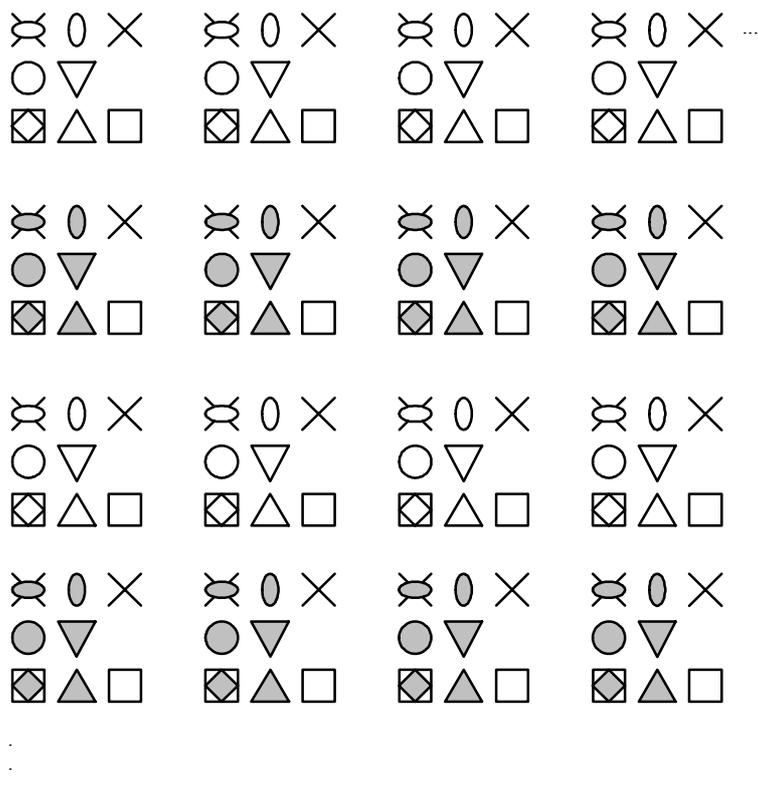
Таблица Е-5 – Матрица коэффициентов

Значение	Матрица	Замечание для информации
0	GBR	Зарезервировано
1	$K_R = 0,2126; K_B = 0,0722$	Рекомендация МСЭ-R ВТ.709-5 и Общество инженеров кино и телевидения, RP177 (1993 г.)
2	Не определена	Характеристики изображения неизвестны или определяются применением.
3	Зарезервирована	Для будущего использования МСЭ-T/ИСО/МЭК
4	$K_R = 0,30; K_B = 0,11$	Федеральная комиссия связи (FCC) Соединенных Штатов, Титул 47, Свод федеральных постановлений (2003 г.), 73.682 (а) (20)
5	$K_R = 0,299; K_B = 0,114$	Рекомендация МСЭ-R ВТ.470-6 Системы В, G
6	$K_R = 0,299; K_B = 0,114$	Общество инженеров кино и телевидения, 170М (1999 г.)
7	$K_R = 0,212; K_B = 0,087$	Общество инженеров кино и телевидения, 240М (1999 г.)
8	YcgCo	См. уравнения Е-19–Е-33
9–255	Зарезервирована	Для будущего использования МСЭ-T/ИСО/МЭК

chroma_loc_info_present_flag, равное 1, определяет, что **chroma_sample_loc_type_top_field** и **chroma_sample_loc_type_bottom_field** присутствуют. Значение **chroma_loc_info_present_flag**, равное 0, определяет, что **chroma_sample_loc_type_top_field** и **chroma_sample_loc_type_bottom_field** отсутствуют.

chroma_sample_loc_type_top_field и **chroma_sample_loc_type_bottom_field** определяют расположение образцов цветности для верхнего и нижнего поля, как показано на рисунке Е-1. Значение **chroma_sample_loc_type_top_field** и **chroma_sample_loc_type_bottom_field** должно быть в диапазоне от 0 до 5 включительно. Если **chroma_sample_loc_type_top_field** и **chroma_sample_loc_type_bottom_field** отсутствуют, то значения **chroma_sample_loc_type_top_field** и **chroma_sample_loc_type_bottom_field** должны быть приняты равными 0.

ПРИМЕЧАНИЕ 5. – При последовательном кодировании исходного материала поля **chroma_sample_loc_type_top_field** и **chroma_sample_loc_type_bottom_field** должны иметь одно и то же значение.



Интерпретация символов:

Указания положений образцов яркости:

⊗ = Верхнее поле образца □ = Нижнее поле образца

Указания положений образцов цветности, где заполнение серым указывает на тип образца нижнего поля, а отсутствие заполнения указывает на тип образца верхнего поля:

○ = Образец цветности типа 0 ○ = Образец цветности типа 0
 ○ = Образец цветности типа ▽ = Образец цветности типа
 ◇ = Образец цветности типа ▲ = Образец цветности типа

Рисунок Е-1 – Расположение образцов цветности для верхнего и нижнего полей как функции chroma_sample_loc_type_top_field и chroma_sample_loc_type_bottom_field

timing_info_present_flag, равное 1, указывает, что num_units_in_tick, time_scale и fixed_frame_rate_flag присутствует в потоке битов. Значение timing_info_present_flag, равное 0, указывает, что num_units_in_tick, time_scale и fixed_frame_rate_flag отсутствуют в потоке битов.

num_units_in_tick – это число блоков синхронизации задающего генератора, работающего на частоте time_scale Гц, которое соответствует одному приращению (называемому временным тактом) счетчика тактов. num_units_in_tick должно быть больше 0. Временной такт – это минимальный интервал времени, который может быть представлен в кодированных данных. Например, если тактовая частота видеосигнала равна 60000 ÷ 1001 Гц, то time_scale может быть равно 60 000, а num_units_in_tick может быть равно 1001. См. уравнение С-1.

time_scale – это число блоков синхронизации, которые поступают в одну секунду. Например, система координации времени, которая измеряет время, используя тактовую частоту 27 МГц, имеет time_scale, равное 27 000 000. Значение time_scale должно быть больше 0.

fixed_frame_rate_flag, равное 1, указывает, что временное расстояние на выходе HRD между двумя любыми последовательными изображениями в порядке выхода ограничено следующим образом. Значение fixed_frame_rate_flag, равное 0, указывает, что не наложено никаких ограничений на временное расстояние на выходе HRD между двумя любыми последовательными изображениями в порядке выхода.

Для каждого изображения n , где n указывает n -ое изображение (в порядке выхода), которое является выходом, и изображение $n - 1$ – это не последнее изображение в потоке битов (в порядке выхода), которое является выходом, значение $\Delta t_{fi,dpb}(n)$ определяют как:

$$\Delta t_{fi,dpb}(n) = \Delta t_{o,dpb}(n) \div \text{DeltaTfiDivisor}, \quad (\text{E-34})$$

где $\Delta t_{o,dpb}(n)$ определено в уравнении C-13, а DeltaTfiDivisor определяется по таблице E-6 на основании значений $\text{pic_struct_present_flag}$, field_pic_flag и pic_struct для кодированной видеопоследовательности, содержащей изображение n . Входы в таблице E-6, отмеченные знаком "-", указывают на отсутствие зависимости DeltaTfiDivisor от соответствующего элемента синтаксиса.

Если $\text{fixed_frame_rate_flag}$ равно 1 для кодированной видеопоследовательности, содержащей изображение n , значение, вычисленное для $\Delta t_{fi,dpb}(n)$, должно быть равно t_c , которое определяется в уравнении C-1 (используя значение t_c для кодированной видеопоследовательности, содержащей изображение n), если какое-либо или оба следующие условия являются истиной для последующего изображения n_1 , которое определяется для использования в уравнении C-13:

- изображение n_1 находится в той же кодированной видеопоследовательности, что и изображение n ;
- изображение n_1 находится в другой кодированной видеопоследовательности, и $\text{fixed_frame_rate_flag}$ в кодированной видеопоследовательности, содержащей изображение n_1 , равно 1, и значение $\text{num_units_in_tick} \div \text{time_scale}$ является одинаковым для обеих кодированных видеопоследовательностей.

Таблица E-6 – Делитель для вычисления $\Delta t_{fi,dpb}(n)$

pic_struct_present_flag	field_pic_flag	pic_struct	DeltaTfiDivisor
0	1	–	1
1	–	1	1
1	–	2	1
0	0	–	2
1	–	0	2
1	–	3	2
1	–	4	2
1	–	5	3
1	–	6	3
1	–	7	4
1	–	8	6

nal_hrd_parameters_present_flag, равное 1, указывает, что параметры NAL HRD (принадлежащие к Типу II соответствия потока битов) присутствуют. Значение **nal_hrd_parameters_present_flag**, равное 0, указывает, что параметры NAL HRD отсутствуют.

ПРИМЕЧАНИЕ 6. – Если **nal_hrd_parameters_present_flag** равно 0, то соответствие потока битов нельзя проверить без присутствия параметров NAL HRD, включая последовательность параметров информации NAL HRD и все сообщения SEI о периодах буферизации и синхронизации изображения, не определенных в настоящей Рекомендации | Международном стандарте другими средствами.

Если **nal_hrd_parameters_present_flag** равно 1, то параметры NAL HRD (в пп. E.1.2 и E.2.2) непосредственно следуют за флагом.

Переменную **NalHrdBpPresentFlag** находят следующим образом.

- Если любое из следующих утверждений – истина, то значение **NalHrdBpPresentFlag** должно быть установлено равным 1:
 - **nal_hrd_parameters_present_flag** присутствует в потоке битов и равно 1;
 - необходимость представлять периоды буферизации работы NAL HRD в потоке битов в сообщениях SEI о периоде буферизации определяется приложением с помощью средств, не определенных в настоящей Рекомендации | Международном стандарте.
- Иначе значение **NalHrdBpPresentFlag** должно быть установлено равным 0.

vcl_hrd_parameters_present_flag, равное 1, определяет, что параметры VCL HRD (принадлежащие ко всем типам соответствия потока битов) присутствуют. Значение **vcl_hrd_parameters_present_flag**, равное 0, определяет, что параметры VCL HRD отсутствуют.

ПРИМЕЧАНИЕ 7. – Если **vcl_hrd_parameters_present_flag** равно 0, то соответствие потока битов нельзя проверить без присутствия параметров NAL HRD и всех сообщений SEI о периодах буферизации и синхронизации изображения, не определенных в настоящей Рекомендации | Международном стандарте другими средствами.

Если `vcl_hrd_parameters_present_flag` равно 1, то параметры NAL HRD (в пп. E.1.2 и E.2.2) непосредственно следуют за флагом.

Переменную `VclHrdBpPresentFlag` находят следующим образом.

- Если любое из следующих утверждений – истина, то значение `VclHrdBpPresentFlag` должно быть установлено равным 1:
 - `vcl_hrd_parameters_present_flag` присутствует в потоке битов и равно 1;
 - необходимость представлять периоды буферизации работы VCL HRD в потоке битов в сообщениях SEI о периоде буферизации определяется приложением с помощью некоторых средств, не определенных в настоящей Рекомендации | Международном стандарте.
- Иначе значение `VclHrdBpPresentFlag` должно быть установлено равным 0.

Переменную `CpbDpbDelaysPresentFlag` находят следующим образом.

- Если любое из следующих утверждений – истина, то значение `CpbDpbDelaysPresentFlag` должно быть установлено равным 1:
 - `nal_hrd_parameters_present_flag` присутствует в потоке битов и равно 1;
 - `vcl_hrd_parameters_present_flag` присутствует в потоке битов и равно 1;
 - необходимость представлять задержки на выходе CPB и DPB в потоке битов в сообщениях SEI о синхронизации изображения определяется приложением с помощью некоторых средств, не определенных в настоящей Рекомендации | Международном стандарте.
- Иначе значение `CpbDpbDelaysPresentFlag` должно быть установлено равным 0.

low_delay_hrd_flag определяет режим работы HRD, как это определено в Приложении C. Если `fixed_frame_rate_flag` равно 1, то `low_delay_hrd_flag` должно быть равно 0.

ПРИМЕЧАНИЕ 8. – Если `low_delay_hrd_flag` равно 1, то разрешены "большие изображения", которые нарушают номинальное время удалений CPB из-за числа битов, использованных блоком доступа. Предполагается, но не требуется, чтобы такие "большие изображения" встречались только случайно.

pic_struct_present_flag, равное 1, указывает, что сообщения SEI о синхронизации изображения (п. D.2.2), которые включают элемент синтаксиса `pic_struct`, присутствуют. Значение `pic_struct_present_flag`, равное 0, указывает, что элемент синтаксиса `pic_struct` отсутствует в сообщениях SEI о синхронизации изображения. Если элемент `pic_struct_present_flag` не присутствует, его значение должно приниматься равным 0.

bitstream_restriction_flag, равное 1, указывает на присутствие ограничений в отношении потока битов кодированной видеопоследовательности. Значение `bitstream_restriction_flag`, равное 0, указывает на отсутствие ограничений в отношении потока битов кодированной видеопоследовательности.

motion_vectors_over_pic_boundaries_flag, равное 0, указывает на отсутствие образца за границами изображения и на отсутствие образца в местоположении фрагмента образца, чьи значения находят, используя один или более образцов за границами изображения с помощью `inter` предсказания любого образца. Значение `motion_vectors_over_pic_boundaries_flag`, равное 1, указывает, что один или более образцов за границами изображения могут быть использованы в `inter` предсказании. Если элемент синтаксиса `motion_vectors_over_pic_boundaries_flag` отсутствует, то значение `motion_vectors_over_pic_boundaries_flag` должно быть принято равным 1.

max_bytes_per_pic_denom указывает число байтов, не превосходящих сумму размеров блоков NAL VCL, связанных с любым кодированным изображением в кодированной видеопоследовательности.

Число байтов, которые представляют изображение в потоке блоков NAL, определено для этой цели как общее число байтов данных блока NAL VCL (т. е. общее число переменных `NumBytesInNALunit` блоков NAL VCL) для данного изображения. Значение `max_bytes_per_pic_denom` должно быть в диапазоне от 0 до 16 включительно.

В зависимости от `max_bytes_per_pic_denom` применяют следующее.

- Если `max_bytes_per_pic_denom` равно 0, не указывают никаких пределов.
- Иначе (`max_bytes_per_pic_denom` не равно 0) никакое кодированное изображение не должно быть представлено в кодированной видеопоследовательности более чем следующим числом байтов:

$$(\text{PicSizeInMbs} * \text{RawMbBits}) \div (8 * \text{max_bytes_per_pic_denom}). \quad (\text{E-35})$$

Если элемент синтаксиса `max_bytes_per_pic_denom` отсутствует, то значение `max_bytes_per_pic_denom` должно быть принято равным 2.

max_bits_per_mb_denom указывает максимальное число кодированных битов данных `macroblock_layer()` для любого макроблока в любом изображении кодированной видеопоследовательности. Значение `max_bits_per_mb_denom` должно быть в диапазоне от 0 до 16 включительно.

В зависимости от `max_bits_per_mb_denom` применяют следующее.

- Если `max_bits_per_mb_denom` равно 0, то не определены никакие пределы.
- Иначе (`max_bits_per_mb_denom` не равно 0) никакое кодированное значение `macroblock_layer()` не должно быть представлено в потоке битов более чем следующим числом битов:

$$(128 + \text{RawMbBits}) \div \text{max_bits_per_mb_denom}. \quad (\text{E-36})$$

В зависимости от `entropy_coding_mode_flag` биты данных `macroblock_layer()` учитывают следующим образом.

- Если `entropy_coding_mode_flag` равно 0, то число битов данных `macroblock_layer()` задано числом битов в структуре синтаксиса `macroblock_layer()` макроблока.
- Иначе (`entropy_coding_mode_flag` равно 1) число битов данных макроблока `macroblock_layer()` задано числом раз `read_bits(1)`, описанным в пп. 9.3.3.2.2 и 9.3.3.2.3, если анализ значения `macroblock_layer()` связан с макроблоком.

Если `max_bits_per_mb_denom` отсутствует, то значение `max_bits_per_mb_denom` должно быть принято равным 1.

`log2_max_mv_length_horizontal` и **`log2_max_mv_length_vertical`** указывают максимальное абсолютное значение декодированных горизонтальных и вертикальных компонентов векторов движения, соответственно, в $\frac{1}{4}$ блоков образцов яркости для всех изображений в кодированной видеопоследовательности. Величина `n` указывает, что никакое значение компонента вектора движения не должно превышать диапазон от -2^n до $2^n - 1$ включительно в блоках размещения образца яркости $\frac{1}{4}$. Значение `log2_max_mv_length_horizontal` должно быть в диапазоне от 0 до 16 включительно. Значение `log2_max_mv_length_vertical` должно быть в диапазоне от 0 до 16 включительно. Если `log2_max_mv_length_horizontal` отсутствует, то значения `log2_max_mv_length_horizontal` и `log2_max_mv_length_vertical` должны быть приняты равными 16.

ПРИМЕЧАНИЕ 9. – Максимальное абсолютное значение декодированных горизонтальных и вертикальных компонентов векторов движения ограничено также границами профилей и уровней, как это определено в Приложении А.

`num_reorder_frames` указывает максимальное число кадров, пар дополнительных полей или непарных полей, которые предшествуют любому кадру, паре дополнительных полей или непарным полям в кодированной видеопоследовательности в порядке декодирования и следуют этому порядку на выходе. Значение `num_reorder_frames` должно быть в диапазоне от 0 до `max_dec_frame_buffering` включительно. Если элемент синтаксиса `num_reorder_frames` отсутствует, то значение `num_reorder_frames` должно быть принято равным `max_dec_frame_buffering`.

`max_dec_frame_buffering` определяет требуемый размер буфера (DPB) декодированного изображения HRD в единицах буферов кадров. Кодированная видеопоследовательность не должна требовать буфера декодированного изображения с размером больше чем $\text{Max}(1, \text{max_dec_frame_buffering})$ буферов кадров, чтобы обеспечить выход декодированных изображений за время выхода, определенного сообщением SEI о синхронизации изображения `dpb_output_delay`. Значение `max_dec_frame_buffering` должно быть в диапазоне от `num_ref_frames` до `MaxDpbSize` (как это определено в п. А.3.1 или А.3.2) включительно. Если элемент синтаксиса `max_dec_frame_buffering` отсутствует, то значение `max_dec_frame_buffering` должно быть принято равным `MaxDpbSize`.

Е.2.2 Семантика параметров HRD

`cpb_cnt_minus1` плюс 1 определяет число спецификаций альтернативных буферов CPB в потоке битов. Значение `cpb_cnt_minus1` должно быть в диапазоне от 0 до 31 включительно. Если `low_delay_hrd_flag` равно 1, то `cpb_cnt_minus1` должно быть равно 0. Если `cpb_cnt_minus1` отсутствует, это значение должно считаться равным 0.

`bit_rate_scale` (совместно с `bit_rate_value_minus1[SchedSelIdx]`) определяет максимальную скорость битов на входе `SchedSelIdx`-ого буфера CPB.

`cpb_size_scale` (совместно с `cpb_size_value_minus1[SchedSelIdx]`) определяет размер CPB для `SchedSelIdx`-ого буфера CPB.

`bit_rate_value_minus1[SchedSelIdx]` (совместно с `bit_rate_scale`) определяет максимальную скорость битов на входе `SchedSelIdx`-ого буфера CPB. Значение `bit_rate_value_minus1[SchedSelIdx]` должно быть в диапазоне от 0 до $2^{32} - 2$ включительно. Для любой величины `SchedSelIdx > 0` значение `bit_rate_value_minus1[SchedSelIdx]` должно быть больше, чем `bit_rate_value_minus1[SchedSelIdx - 1]`. Скорость в битах в секунду задают как:

$$\text{BitRate}[\text{SchedSelIdx}] = (\text{bit_rate_value_minus1}[\text{SchedSelIdx}] + 1) * 2^{(6 + \text{bit_rate_scale})}. \quad (\text{E-37})$$

Если элемент синтаксиса `bit_rate_value_minus1[SchedSelIdx]` не присутствует, значение `BitRate[SchedSelIdx]` должно определяться следующим образом:

- если `profile_idc` равно 66, 77 или 88, `BitRate[SchedSelIdx]` должно приниматься равным $1000 * \text{MaxBR}$ для параметров VCL HRD и равным $1200 * \text{MaxBR}$ для параметров NAL HRD, где `MaxBR` описывается в п. А.3.1;
- иначе `BitRate[SchedSelIdx]` должно приниматься равным `cpbBrVclFactor * MaxBR` для параметров VCL HRD и равным `cpbBrNalFactor * MaxBR` для параметров NAL HRD, где `cpbBrVclFactor`, `cpbBrNalFactor` и `MaxBR` описываются в п. А.3.3.

`cpb_size_value_minus1[SchedSelIdx]` используют совместно с `cpb_size_scale` для определения размера `SchedSelIdx`-ого буфера CPB. Значение `cpb_size_value_minus1[SchedSelIdx]` должно быть в диапазоне от 0 до $2^{32} - 2$ включительно. Для любой величины `SchedSelIdx` больше, чем 0, значение `cpb_size_value_minus1[SchedSelIdx]` должно быть меньше чем или равно `cpb_size_value_minus1[SchedSelIdx - 1]`.

Размер буфера CPB в битах задают как:

$$\text{CpbSize[SchedSelIdx]} = (\text{cpb_size_value_minus1[SchedSelIdx]} + 1) * 2^{(4 + \text{cpb_size_scale})}. \quad (\text{E-38})$$

Если элемент синтаксиса `cpb_size_value_minus1[SchedSelIdx]` отсутствует, значение `CpbSize[SchedSelIdx]` должно определяться следующим образом.

- Если `profile_idc` равно 66, 77 или 88, `CpbSize[SchedSelIdx]` должно приниматься равным $1000 * \text{MaxCPB}$ для параметров VCL HRD и равным $1200 * \text{MaxCPB}$ для параметров NAL HRD, где `MaxCPB` описывается в п. А.3.1.
- Иначе `CpbSize[SchedSelIdx]` должно приниматься равным `cpbBrVclFactor * MaxCPB` для параметров VCL HRD и равным `cpbBrNalFactor * MaxCPB` для параметров NAL HRD, где `cpbBrVclFactor`, `cpbBrNalFactor` и `MaxCPB` описываются в п. А.3.3.

`cbr_flag[SchedSelIdx]`, равное 0, определяет, что при декодировании данного потока битов декодером HRD, используя спецификацию `SchedSelIdx`-го буфера CPB, планировщик доставки гипотетического потока (HSS) работает в режиме переменной скорости битов. Значение `cbr_flag[SchedSelIdx]`, равное 1, определяет, что HSS работает в режиме постоянной скорости битов (CBR). Если элемент синтаксиса `cbr_flag[SchedSelIdx]` отсутствует, то значение `cbr_flag` должно быть принято равным 0.

`initial_cpb_removal_delay_length_minus1` определяет длину в битах элементов синтаксиса `initial_cpb_removal_delay[SchedSelIdx]` и `initial_cpb_removal_delay_offset[SchedSelIdx]` в сообщении SEI о периоде буферизации. Длина `initial_cpb_removal_delay[SchedSelIdx]` и `initial_cpb_removal_delay_offset[SchedSelIdx]` равна `initial_cpb_removal_delay_length_minus1 + 1`. Если элемент синтаксиса `initial_cpb_removal_delay_length_minus1` присутствует более чем в одной структуре синтаксиса `hrd_parameters()` среди структур синтаксиса параметров VUI, то значение параметров `initial_cpb_removal_delay_length_minus1` должно быть равным в обеих структурах синтаксиса `hrd_parameters()`. Если элемент синтаксиса `initial_cpb_removal_delay_length_minus1` отсутствует, то он должен быть принят равным 23.

`cpb_removal_delay_length_minus1` определяет длину в битах элемента синтаксиса `cpb_removal_delay`. Длина элемента синтаксиса `cpb_removal_delay` в сообщении SEI о синхронизации изображения равна `cpb_removal_delay_length_minus1 + 1`. Если элемент синтаксиса `cpb_removal_delay_length_minus1` присутствует более чем в одной структуре синтаксиса `hrd_parameters()` среди структур синтаксиса параметров VUI, то значение параметров `cpb_removal_delay_length_minus1` должно быть равным в обеих структурах синтаксиса `hrd_parameters()`. Если элемент синтаксиса `cpb_removal_delay_length_minus1` отсутствует, то он должен быть принят равным 23.

`dpb_output_delay_length_minus1` определяет длину в битах элемента синтаксиса `dpb_output_delay`. Длина элемент синтаксиса `dpb_output_delay` в сообщении SEI о синхронизации изображения равна `dpb_output_delay_length_minus1 + 1`. Если элемент синтаксиса `dpb_output_delay_length_minus1` присутствует более чем в одной структуре синтаксиса `hrd_parameters()` среди структур синтаксиса параметров VUI, то значение параметров `dpb_output_delay_length_minus1` должно быть равным в обеих структурах синтаксиса `hrd_parameters()`. Если элемент синтаксиса `dpb_output_delay_length_minus1` отсутствует, то он должен быть принят равным 23.

`time_offset_length` больше, чем 0, определяет длину в битах элемента синтаксиса `time_offset`. Значение `time_offset_length` равно 0, указывает на отсутствие элемента синтаксиса `time_offset`. Если элемент синтаксиса `time_offset_length` присутствует более чем в одной структуре синтаксиса `hrd_parameters()` среди структур синтаксиса параметров VUI, то значение параметров `time_offset_length` должно быть равным в обеих структурах синтаксиса `hrd_parameters()`. Если элемент синтаксиса `time_offset_length` отсутствует, то он должен быть принят равным 24.

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия D	Общие принципы тарификации
Серия E	Общая эксплуатация сети, телефонная служба, функционирование служб и человеческие факторы
Серия F	Нетелефонные службы электросвязи
Серия G	Системы и среда передачи, цифровые системы и сети
Серия H	Аудиовизуальные и мультимедийные системы
Серия I	Цифровая сеть с интеграцией служб
Серия J	Кабельные сети и передача сигналов телевизионных и звуковых программ и других мультимедийных сигналов
Серия K	Защита от помех
Серия L	Конструкция, прокладка и защита кабелей и других элементов линейно-кабельных сооружений
Серия M	Управление электросвязью, включая СУЭ и техническое обслуживание сетей
Серия N	Техническое обслуживание: международные каналы передачи звуковых и телевизионных программ
Серия O	Требования к измерительной аппаратуре
Серия P	Качество телефонной передачи, телефонные установки, сети местных линий
Серия Q	Коммутация и сигнализация
Серия R	Телеграфная передача
Серия S	Оконечное оборудование для телеграфных служб
Серия T	Оконечное оборудование для телематических служб
Серия U	Телеграфная коммутация
Серия V	Передача данных по телефонной сети
Серия X	Сети передачи данных, взаимосвязь открытых систем и безопасность
Серия Y	Глобальная информационная инфраструктура, аспекты межсетевого протокола и сети последующих поколений
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи