



МЕЖДУНАРОДНЫЙ СОЮЗ ЭЛЕКТРОСВЯЗИ

МСЭ-Т

СЕКТОР СТАНДАРТИЗАЦИИ
ЭЛЕКТРОСВЯЗИ МСЭ

T.800

(08/2002)

СЕРИЯ Т: ОКОНЕЧНОЕ ОБОРУДОВАНИЕ ДЛЯ
ТЕЛЕМАТИЧЕСКИХ СЛУЖБ

**Информационная технология – Система
кодирования изображений JPEG 2000:
Основы системы кодирования**

Рекомендация МСЭ-Т T.800

**Информационная технология – система кодирования изображений JPEG 2000:
Основы системы кодирования**

Резюме

В настоящей Рекомендации | Международном стандарте описываются методы компрессии без потерь (с защитой битов) и с потерями при кодировании двухуровневым кодом цифровых неподвижных изображений монотонно-серого цвета, с цветовыми пакетами или с монотонными цветами.

В данной Рекомендации | Международном стандарте:

- описываются процессы декодирования для преобразования данных сжатого изображения в восстановленные данные изображения;
- описывается синтаксис кодированного потока, который содержит информацию для интерпретации данных сжатого изображения;
- описывается файловый формат;
- дано руководство по процессам кодирования для преобразования данных источника изображения в данные сжатого изображения;
- дано руководство по внедрению этих процессов в практику.

Источник

Рекомендация МСЭ-Т Т.800 подготовлена 16-й Исследовательской комиссией МСЭ-Т (2001–2004 гг.) и утверждена 29 августа 2002 года. Идентичный текст также опубликован как ИСО/МЭК 15444-1.

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.

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.

ПРИМЕЧАНИЕ

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

ПРАВА ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

МСЭ обращает внимание на то, 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 извещения об интеллектуальной собственности, защищенной патентами, которые могут потребоваться для реализации 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.

© ITU 2005

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

СОДЕРЖАНИЕ

	<i>Стр.</i>
1 Область применения	1
2 Ссылки.....	1
2.1 Тожественные Рекомендации Международные стандарты.....	1
2.2 Дополнительные ссылки	1
3 Определения	2
4 Сокращения и символы.....	6
4.1 Сокращения	6
4.2 Символы.....	7
5 Общее описание.....	8
5.1 Цель	8
5.2 Кодированный поток	8
5.3 Принципы кодирования.....	9
6 Требования к кодировщику	10
7 Требования к декодеру.....	11
7.1 Требования к синтаксису кодированного потока.....	11
7.2 Требования к дополнительному файловому формату	11
8 Требования к применению	11
Приложение А – Синтаксис кодированного потока.....	12
А.1 Маркеры, сегменты маркеров и заголовки	12
А.2 Информация в сегментах маркеров	14
А.3 Структура кодированного потока.....	15
А.4 Разграничительные маркеры и сегменты маркера	19
А.5 Сегмент маркера неизменяемой информации	20
А.6 Сегменты функционального маркера.....	22
А.7 Сегменты маркера указателя.....	32
А.8 Маркер в потоке битов и сегменты маркера.....	38
А.9 Информационные сегменты маркера	39
А.10 Ограничения по соответствию кодированного потока данной Рекомендации Международному стандарту	40
Приложение В – Изображение и упорядочение данных сжатого изображения	42
В.1 Введение в концепцию структуры данных изображения	42
В.2 Размещение компонентов на опорной сетке.....	42
В.3 Разделение области изображения на тайлы и компоненты тайла.....	44
В.4 Пример размещения компонентов на опорной сетке (информативный)	45
В.5 Разделение преобразованного компонента тайла на уровни разрешения и поддиапазоны	48
В.6 Разделение уровней разрешения на граничные области	49
В.7 Разделение поддиапазонов на блоки кодирования	50
В.8 Слои.....	51
В.9 Пакеты.....	52
В.10 Информация о кодировании в заголовке пакета	54
В.11 Тайлы и части тайла.....	59
В.12 Порядок последовательности.....	59
Приложение С – Арифметическое статистическое кодирование.....	64
С.1 Бинарное кодирование (информативное)	64
С.2 Описание арифметического кодировщика.....	65
С.3 Процедура арифметического декодирования	76

	<i>Стр.</i>
Приложение D – Моделирование битовых коэффициентов.....	84
D.1 Шаблон сканирования блока кодирования.....	84
D.2 Битовые коэффициенты и их значения.....	84
D.3 Декодирующие проходы по битовым матрицам.....	85
D.4 Инициализация и завершение.....	89
D.5 Символ сегментации для защиты от ошибок.....	90
D.6 Избирательный обход арифметического кодирования.....	90
D.7 Формирование вертикального причинного контекста.....	92
D.8 График последовательности кодирования блока.....	92
Приложение E – Квантование.....	95
E.1 Процедура инверсного квантования.....	95
E.2 Квантование скалярного коэффициента (информативное).....	97
Приложение F – Дискретное частотное (волновое) преобразование компонентов тайла.....	98
F.1 Параметры компонента тайла.....	98
F.2 Дискретные частотные преобразования.....	98
F.3 Необратимое дискретное частотное преобразование.....	98
F.4 Прямое преобразование (информативное).....	110
Приложение G – Сдвиг уровня DC и многократные преобразования компонента.....	120
G.1 Сдвиг уровня DC компонентов тайла.....	120
G.2 Обратимое сложное преобразование компонента (RCT).....	121
G.3 Обратимое сложное преобразование компонента (ICT).....	121
G.4 Выборка компонента цветности и опорная сетка.....	122
Приложение H – Кодирование изображений с областями интересов.....	123
H.1 Декодирование ROI.....	123
H.2 Описание метода Maxshift.....	123
H.3 Комментарии по кодированию области интересов (информативное).....	124
Приложение I – Синтаксис файлового формата JP2.....	127
I.1 Область применения файлового формата.....	127
I.2 Введение в файловый формат JP2.....	127
I.3 Архитектура спецификации Оттенки серого/ Цвет/ Цветовая палитра/ Многокомпонентность.....	129
I.4 Определение блока.....	131
I.5 Определенные блоки.....	133
I.6 Дополнительная информация в файле JP2 о правах на интеллектуальную собственность.....	148
I.7 Дополнение к формату файла JP2 специальной информации для продавцов.....	148
I.8 Обращение с неизвестными блоками.....	150
Приложение J – Примеры и руководства.....	151
J.1 Условные обозначения в программах адаптивного статистического декодера.....	151
J.2 Выбор размера шагов квантования для необратимых преобразований.....	153
J.3 Импульсная характеристика фильтра, соответствующая процедурам необратимой фильтрации с приподнятым основанием.....	153
J.4 Пример дискретного частотного преобразования.....	154
J.5 Частотное преобразование по рядам.....	158
J.6 Кодирование, основанное на сканировании.....	167
J.7 Защита от ошибок.....	167
J.8 Требование совместимости с файлами JFIF/SPIFF.....	168
J.9 Использование ограниченного метода консорциума ICC за пределами полной машины управления цветом ICC.....	168
J.10 Пример интерполяции множества компонентов.....	173
J.11 Пример декодирования с показом промежуточных шагов.....	173
J.12 Визуальное частотное взвешивание.....	177
J.13 Вторичный отбор компонентов устройством кодирования.....	179
J.14 Управление скоростью.....	180
J.15 Руководства по оперированию с кодированным потоком YCC.....	184

	<i>Стр.</i>
Приложение К – Библиография	186
К.1 Общее	186
К.2 Квантование и статистическое кодирование	186
К.3 Частотное преобразование	186
К.4 Кодирование области интересов.....	187
К.5 Визуальное частотное взвешивание	187
К.6 Защита от ошибок	187
К.7 Кодирование, основанное на сканировании	188
К.8 Цвет	188
Приложение L – Заявление о патентных правах.....	189
Индекс.....	190

СПИСОК РИСУНКОВ

Стр.

Рисунок 5-1 – Подробная блок-диаграмма.....	9
Рисунок А.1 – Пример описания составляющих сегмента маркера	13
Рисунок А.2 – Структура кодированного потока	16
Рисунок А.3 – Структура основного заголовка	17
Рисунок А.4 – Структура первого заголовка части данного тайла	18
Рисунок А.5 – Структура не первого заголовка части тайла.....	18
Рисунок А.6 – Синтаксис запуска части тайла.....	19
Рисунок А.7 – Синтаксис размеров изображения и тайла	21
Рисунок А.8 – Синтаксис способа кодирования по умолчанию	23
Рисунок А.9 – Диаграмма типов кодирования параметров маркеров SGcod и SPcod	24
Рисунок А.10 – Синтаксис типа кодирования компонента.....	26
Рисунок А.11 – Диаграмма типа кодирования параметров SPcos.....	27
Рисунок А.12 – Синтаксис области интересов.....	27
Рисунок А.13 – Синтаксис квантования по умолчанию.....	28
Рисунок А.14 – Quantization component syntax.....	30
Рисунок А.15 – Синтаксис тайла для изменения порядка последовательности	31
Рисунок А.16 – Длины частей тайла	32
Рисунок А.17 – Синтаксис длины части тайла.....	33
Рисунок А.18 – Длины пакетов, синтаксис основного заголовка	34
Рисунок А.19 – Длина пакета, синтаксис заголовка части тайла	35
Рисунок А.20 – Заголовки упакованных пакетов, синтаксис основного заголовка	36
Рисунок А.21 – Заголовки упакованных пакетов, синтаксис заголовка части тайла	37
Рисунок А.22 – Синтаксис запуска пакета	38
Рисунок А.23 – Синтаксис регистрации компонента	39
Рисунок А.24 – Синтаксис комментария	40
Рисунок В.1 – Диаграмма опорной сетки	43
Рисунок В.2 – Расположение образца компонента на опорной сетке с разными величинами XRsize и YRsize	43
Рисунок В.3 – Пример расположения верхнего левого образца компонента	44
Рисунок В.4 – Диаграмма размещения тайлов на опорной сетке.....	44
Рисунок В.5 – Пример опорной сетки.....	46
Рисунок В.6 – Пример размеров и расположений тайлов для компонента 0.....	47
Рисунок В.7 – Пример размеров и расположений тайлов для компонента 1	48
Рисунок В.8 – Граничные области одного сокращенного разрешения	49

Рисунок В.9 – Блоки кодирования и граничные области в поддиапазоне b из четырех различных тайлов ...	51
Рисунок В.10 – Диаграмма граничных областей с одним уровнем разрешения для одного компонента.....	52
Рисунок В.11 – Диаграмма блоков кодирования внутри граничной области с одним уровнем разрешения .	53
Рисунок В.12 – Пример представления дерева меток	54
Рисунок В.13 – Пример информации, поступившей в кодировщик	57
Рисунок В.14 – Пример значений порядка последовательности в двух измерениях	62
Рисунок В.15 – Пример размещения сегментов маркера РОС	63
Рисунок С.1 – Входы и выходы арифметического кодировщика	64
Рисунок С.2 – Кодировщик для кодера MQ.....	66
Рисунок С.3 – Процедура ENCODE.....	67
Рисунок С.4 – Процедура CODE1	67
Рисунок С.5 – Процедура CODE0	68
Рисунок С.6 – Процедура CODELPS с условным обменом MPS/LPS.....	69
Рисунок С.7 – Процедура CODEMPS с условным обменом MPS/LPS.....	71
Рисунок С.8 – Процедура перенормировки кодировщика.....	72
Рисунок С.9 – Процедура BYTEOUT для кодировщика.....	73
Рисунок С.10 – Инициализация кодировщика	74
Рисунок С.11 – Процедура FLUSH	75
Рисунок С.12 – Установка конечных битов в регистре С	76
Рисунок С.13 – Входы и выходы арифметического декодера.....	76
Рисунок С.14 – Декодер для кодера MQ.....	77
Рисунок С.15 – Декодирование MPS или LPS	78
Рисунок С.16 – Процедура условного обмена декодера для пути MPS.....	79
Рисунок С.17 – Процедура условного обмена декодера для ветви LPS	80
Рисунок С.18 – Процедура перенормировки декодера	81
Рисунок С.19 – Процедура BYTEIN для декодера	82
Рисунок С.20 – Инициализация декодера.....	83
Рисунок D.1 – Пример шаблона сканирования битовой матрицы блока кодирования.....	84
Рисунок D.2 – Прилегающие состояния, которые используют для формирования контекста.....	85
Рисунок D.3 – Последовательность операций для всех проходов кодирования битовой матрицы блока кодирования	93
Рисунок F.1 – Входы и выходы процедуры IDWT	98
Рисунок F.2 – Процедура IDWT для случая ($N_L = 2$).....	99
Рисунок F.3 – Процедура IDWT	100
Рисунок F.4 – Входы и выходы процедуры 2D_SR.....	100
Рисунок F.5 – Один уровень восстановления в один поддиапазон из четырех (процедура 2D_SR).....	100
Рисунок F.6 – Процедура 2D_SR.....	101
Рисунок F.7 – Параметры процедуры 2D_INTERLEAVE.....	101
Рисунок F.8 – Процедура 2D_INTERLEAVE.....	102

Рисунок F.9 – Входы и выходы процедуры HOR_SR	103
Рисунок F.10 – Процедура HOR_SR	104
Рисунок F.11 – Входы и выходы процедуры VER_SR.....	105
Рисунок F.12 – Процедура VER_SR.....	105
Рисунок F.13 – Параметры процедуры 1D_SR	106
Рисунок F.14 – Процедура 1D_SR.....	106
Рисунок F.15 – Периодическое симметричное расширение сигнала.....	106
Рисунок F.16 – Параметры процедуры ID_FILTR.....	107
Рисунок F.17 – Входы и выходы процедуры FDWT	110
Рисунок F.18 – FDWT ($N_L = 2$)	110
Рисунок F.19 – Процедура FDWT	111
Рисунок F.20 – Входы и выходы процедуры 2D_SD.....	111
Рисунок F.21 – Один уровень декомпозиции в четырех поддиапазонах (процедура 2D_SD)	112
Рисунок F.22 – Процедура 2D_SD	112
Рисунок F.23 – Входы и выходы процедуры VER_SD.....	112
Рисунок F.24 – Процедура VER_SD	113
Рисунок F.25 – Входы и выходы процедуры HOR_SD	114
Рисунок F.26 – Процедура HOR_SD.....	114
Рисунок F.27 – Параметры процедуры 2D_DEINTERLEAVE	115
Рисунок F.28 – Процедура 2D_DEINTERLEAVE.....	116
Рисунок F.29 – Параметры процедуры 1D_SD	117
Рисунок F.30 – Процедура 1D_SD	117
Рисунок F.31 – Параметры процедуры 1D_FILTD	118
Рисунок G.1 – Размещение сдвига уровня DC с преобразованием компонента.....	120
Рисунок G.2 – Размещение сдвига уровня DC без преобразования компонента.....	120
Рисунок H.1 – Инверсное частотное преобразование с обратимым фильтром 5-3	125
Рисунок H.2 – Инверсное частотное преобразование с обратимым фильтром 9-7	125
Рисунок I.1 – Концептуальная структура файла JP2	128
Рисунок I.2 – Пример описания содержимого блока	131
Рисунок I.3 – Пример описания содержимого суперблока	131
Рисунок I.4 – Организация блока	131
Рисунок I.5 – Иллюстрация длины блоков.....	132
Рисунок I.6 – Организация содержимого блока типа файла.....	134
Рисунок I.7 – Организация содержимого блока заголовка JP2	135
Рисунок I.8 – Организация содержимого блока заголовка изображения	136
Рисунок I.9 – Организация содержимого блока биты на компонент	137
Рисунок I.10 – Организация содержимого блока спецификации цвета.....	138
Рисунок I.11 – Организация содержимого блока палитры	140

Рисунок I.12 – Организация содержимого блока размещения компонентов	141
Рисунок I.13 – Организация содержимого блока определения канала	142
Рисунок I.14 – Организация содержимого блока разрешения	145
Рисунок I.15 – Организация содержимого блока разрешения захвата	145
Рисунок I.16 – Организация содержимого блока разрешения отображения по умолчанию	146
Рисунок I.17 – Организация содержимого блока непрерывного кодированного потока.....	147
Рисунок I.18 – Организация содержимого блока XML.....	148
Рисунок I.19 – Организация содержимого блока UUID.....	148
Рисунок I.20 – Организация содержимого блока UUID info	149
Рисунок I.21 – Организация содержимого блока перечня UUID	149
Рисунок I.22 – Организация содержимого блока входных данных URL	150
Рисунок J.1 – Инициализация программы условных обозначений декодера.....	151
Рисунок J.2 – Декодирование MPS или LPS в программе условных обозначений декодера.....	152
Рисунок J.3 – Введение нового байта в регистр C в программе условных обозначений декодера.....	152
Рисунок J.4 – Процедура FDWT_ROW	159
Рисунок J.5 – Процедура GET_ROW	160
Рисунок J.6 – Процедура INIT	161
Рисунок J.7 – Процедура START_VERT	162
Рисунок J.8 – Процедура RB_VERT_1	163
Рисунок J.9 – Процедура RB_VERT_2	164
Рисунок J.10 – Процедура END_1	165
Рисунок J.11 – Процедура END_2	166
Рисунок J.12 – Иллюстрация вкладов блоков кодирования в слои потока битов	181
Рисунок J.13 – Формат 4:2:2 (взаимно смещенный).....	184
Рисунок J.14 – Формат 4:2:2 (центрированный)	184
Рисунок J.15 – Формат 4:2:0 (взаимно смещенный).....	185
Рисунок J.16 – Формат 4:2:0 (центрированный)	185

СПИСОК ТАБЛИЦ

	<i>Стр.</i>
Таблица А.1 – Определения маркеров.....	13
Таблица А.2 – Список маркеров и сегментов маркеров.....	14
Таблица А.3 – Информация в сегментах маркеров	15
Таблица А.4 – Значение параметров запуска кодированного потока.....	19
Таблица А.5 – Значения параметров запуска части тайла	20
Таблица А.6 – Число частей тайла, TNsot, значение параметра.....	20
Таблица А.7 – Значения параметров запуска данных	20
Таблица А.8 – Значения параметров конца кодированного потока.....	20
Таблица А.9 – Значения параметров размеров изображения и тайла.....	22
Таблица А.10 – Параметр ресурса Rsiz.....	22
Таблица А.11 – Параметр компонента Ssiz.....	22
Таблица А.12 – Значение параметров типа кодирования по умолчанию.....	23
Таблица А.13 – Значение параметров типа кодирования для маркера Scod.....	24
Таблица А.14 – Значение параметров типа кодирования для маркера SGcod.....	24
Таблица А.15 – Значения типов кодирования параметров маркеров SPcod и SPcoc	24
Таблица А.16 – Порядок последовательности параметров SGcod, SPcoc и Ppoc.....	25
Таблица А.17 – Преобразование множества компонентов параметров SGcod.....	25
Таблица А.18 – Высота и ширина и блоков кодирования параметров SPcod и SPcoc.....	25
Таблица А.19 – Тип блока кодирования параметров SPcod и SPcoc	25
Таблица А.20 – Преобразование параметров SPcod и SPcoc.....	26
Таблица А.21 – Границы ширины и высоты параметров SPcod и SPcoc	26
Таблица А.22 – Значения параметров кодирования компонентов.....	27
Таблица А.23 – Тип кодирования значений параметров для параметра Scoc.....	27
Таблица А.24 – Значения параметров области интересов	28
Таблица А.25 – Значения параметров области интересов для параметра Srgn.....	28
Таблица А.26 – Значения области интересов, полученные из параметра SPrgn (Srgn = 0)	28
Таблица А.27 – Значения параметров квантования по умолчанию	29
Таблица А.28 – Значения квантования по умолчанию для параметров Sqcd и Sqcc.....	29
Таблица А.29 – Значения размеров обратимых шагов для параметров SPqcd и SPqcc (только при обратимом преобразовании).....	29
Таблица А.30 – Значения квантования для параметров SPqcd и SPqcc (только для необратимого преобразования).....	30
Таблица А.31 – Значения параметров квантования компонентов.....	31
Таблица А.32 – Изменение порядка последовательности, значения параметров тайла	32
Таблица А.33 – Значения параметров длины части тайла	33

Таблица А.34 – Размеры параметров Stlm.....	34
Таблица А.35 – Значение параметров длины пакетов и основного заголовка.....	35
Таблица А.36 – Список длины пакетов Iplm, Iplt	35
Таблица А.37 – Длина пакета, значения параметров заголовков частей тайла	36
Таблица А.38 – Заголовки упакованных пакетов, значения параметров основного заголовка.....	37
Таблица А.39 – Значения параметров заголовка пакета и заголовков частей тайла	37
Таблица А.40 – Значения параметров запуска пакетов.....	38
Таблица А.41 – Значения параметров конца заголовка пакета	39
Таблица А.42 – Значения параметров регистрации компонентов.....	39
Таблица А.43 – Значения параметров комментария.....	40
Таблица А.44 – Значения регистрации для параметра Rcom.....	40
Таблица А.45 – Ограничения кодированных потоков.....	41
Таблица В.1 – Величины (x_{0b}, y_{0b}) для поддиапазона b.....	49
Таблица В.2 – Пример формирования слоя (показан только один компонент).....	52
Таблица В.3 – Пример формирования пакета	53
Таблица В.4 – Кодовые слова и количества проходов кодирования для блока кодирования	56
Таблица В.5 – Пример потока битов заголовка пакета	58
Таблица С.1 – Структуры регистра кодировщика	66
Таблица С.2 – Значения Q_e и вероятностная оценка.....	69
Таблица С.2 – Значения Q_e и вероятностная оценка (продолж.).....	70
Таблица С.3 – Структуры регистра декодера.....	77
Таблица D.1 – Контексты для кодирующих проходов восстановления значимости и очистки.....	86
Таблица D.2 – Вклады от вертикальных и горизонтальных окружений в знак контекста	86
Таблица D.3 – Контексты знаков от вертикальных горизонтальных вкладов	87
Таблица D.4 – Контексты для подстройки величин кодирующих проходов	87
Таблица D.5 – Обрабатываемая декодером длина при проходах очистки	88
Таблица D.6 – Пример порядка кодирования части битовой матрицы и восстановления значимости.....	88
Таблица D.7 – Исходные состояния для всех контекстов.....	89
Таблица D.8 – Шаблоны завершения работы арифметического кодера	89
Таблица D.9 – Избирательный обход арифметического кодирования	91
Таблица D.10 – Решения в последовательности операций модели контекста	94
Таблица D.11 – Последовательности операций декодирования модели контекста	94
Таблица E.1 – Коэффициент передачи поддиапазона	96
Таблица F.1 – Уровень декомпозиции n_b для поддиапазона b.....	99
Таблица F.2 – Расширение влево.....	107
Таблица F.3 – Расширение вправо	107
Таблица F.4 – Определение параметров фильтрации для необратимого фильтра 9-7	109
Таблица F.5 – Определение коэффициентов g_n	109

Таблица F.6 – Промежуточные выражения (r_0, r_1, s_0, t_0)	109
Таблица F.7 – Промежуточные выражения.....	110
Таблица F.8 – Расширение влево.....	117
Таблица F.9 – Расширение вправо	118
Таблица I.1 – Бинарная структура блока	132
Таблица I.2 – Определенные блоки	133
Таблица I.3 – Разрешенные значения Brand.....	134
Таблица I.4 – Формат содержания блока типа файла.....	135
Таблица I.5 – Формат содержимого блока заголовка изображения.....	137
Таблица I.6 – Значения BPC	137
Таблица I.7 – Форма содержимого блока биты на компонент	138
Таблица I.8 – Значения BPC ⁱ	138
Таблица I.9 – Разрешенные значения METH	139
Таблица I.10 – Разрешенные значения EnumCS	139
Таблица I.11 – Формат содержимого блока спецификации цвета	140
Таблица I.12 – Формат содержимого блока палитры.....	141
Таблица I.13 – Значения B ⁱ	141
Таблица I.14 – Значения поля MTPY ⁱ	142
Таблица I.15 – Формат содержимого блока размещения компонента.....	142
Таблица I.16 – Значения поля Tur ⁱ	143
Таблица I.17 – Значения поля Assoc ⁱ	143
Таблица I.18 – Цвета, указанные в поле Assoc ⁱ	144
Таблица I.19 – Формат блока определения канала	145
Таблица I.20 – Формат содержимого блока разрешения захвата.....	146
Таблица I.21 – Формат содержимого блока разрешения отображения по умолчанию.....	147
Таблица I.22 – Формат содержимого блока непрерывного кодированного потока	147
Таблица I.23 – Формат содержимого блока UUID	149
Таблица I.24 – Значения структуры данных содержимого блока перечня UUID.....	149
Таблица I.25 – Значения структуры данных содержимого блока входных данных URL	150
Таблица J.1 – Коэффициенты поддиапазона 1НН (необратимая частотная трансформация 9-7).....	153
Таблица J.2 – Определение импульсных характеристик для синтеза блока необратимых фильтров 9-7	154
Таблица J.3 – Образцы компонентов источника тайла	154
Таблица J.4 – Коэффициенты поддиапазона 2LL (необратимая частотная трансформация 9-7).....	155
Таблица J.5 – Коэффициенты поддиапазона 2НЛ (необратимая частотная трансформация 9-7)	155
Таблица J.6 – Коэффициенты поддиапазона 2ЛН (необратимая частотная трансформация 9-7)	155
Таблица J.7 – Коэффициенты поддиапазона 2НН (необратимая частотная трансформация 9-7).....	155
Таблица J.8 – Коэффициенты поддиапазона 1НЛ (необратимая частотная трансформация 9-7)	155
Таблица J.9 – Коэффициенты поддиапазона 1ЛН (необратимая частотная трансформация 9-7)	156

Таблица J.10 – Коэффициенты поддиапазона 1НН (необратимая частотная трансформация 9-7).....	156
Таблица J.11 – Коэффициенты поддиапазона 2LL (обратимое частотное преобразование 5-3).....	156
Таблица J.12 – Коэффициенты поддиапазона 2НН (обратимое частотное преобразование 5-3)	156
Таблица J.13 – Коэффициенты поддиапазона 2LN (обратимое частотное преобразование 5-3)	157
Таблица J.14 – Коэффициенты поддиапазона 2НН (обратимое частотное преобразование 5-3).....	157
Таблица J.15 – Коэффициенты поддиапазона 1НН (обратимое частотное преобразование 5-3)	157
Таблица J.16 – Коэффициенты поддиапазона 1LN (обратимое частотное преобразование 5-3)	157
Таблица J.17 – Коэффициенты поддиапазона 1НН (обратимое частотное преобразование 5-3).....	158
Таблица J.18 – Инструментальные средства защиты от ошибок	167
Таблица J.19 – Теги обработки, которые используют ограниченные характеристики ICC	169
Таблица J.20 – Декодирование заголовка первого пакета	175
Таблица J.21 – Декодирование заголовка второго пакета.....	175
Таблица J.22 – Арифметическое декодирование первого блока кодирования	176
Таблица J.23 – Арифметическое декодирование второго блока кодирования	177
Таблица J.24 – Рекомендованные частотные взвешивания	179
Таблица J.25 – Рекомендованные частотные взвешивания для (цветных) изображений с множеством компонентов	179
Таблица J.26 – Значения CRG (регистрация компонентов).....	185
Таблица L.1 – Заявление о правах на интеллектуальную собственность.....	189

**МЕЖДУНАРОДНЫЙ СТАНДАРТ
РЕКОМЕНДАЦИЯ МСЭ-Т**

**Информационная технология – система кодирования изображений JPEG 2000:
Основы системы кодирования**

1 Область применения

В этой Рекомендации | Международном стандарте описываются методы компрессии без потерь (с защитой битов) и с потерями при кодировании двухуровневым кодом цифровых неподвижных изображений монотонно-серого цвета, с цветовыми пакетами или с монотонными цветами.

В этой Рекомендации | Международном стандарте:

- описываются процессы декодирования для преобразования данных сжатого изображения в восстановленные данные изображения;
- описывается синтаксис кодированного потока, который содержит информацию для интерпретации данных сжатого изображения;
- описывается файловый формат;
- дано руководство по процессам кодирования для преобразования данных источника изображения в данные сжатого изображения;
- дано руководство по внедрению этих процессов в практику.

2 Ссылки

В перечисленных ниже Рекомендациях и Международных стандартах содержатся положения, которые в рамках данного текста являются положениями настоящей Рекомендации | Международного стандарта. На момент публикации приведенные редакции имели силу. Все Рекомендации и стандарты могут пересматриваться. В связи с этим всем пользователям настоящей Рекомендации | Международного стандарта предоставляется возможность пользоваться самыми новыми редакциями Рекомендаций и стандартов, перечисленных ниже. Члены МЭК и ИСО имеют доступ к официальному списку текущих действующих Международных стандартов. В Бюро Стандартизации электросвязи МСЭ имеется список текущих действующих Рекомендаций МСЭ-Т.

2.1 Тожественные Рекомендации | Международные стандарты

- ITU-T Recommendation T.81 (1992) | ISO/IEC 10918-1:1994, *Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines.*
- ITU-T Recommendation T.88 (2000) | ISO/IEC 14492:2001, *Information technology – Lossy/lossless coding of bi-level images.*
- ISO/IEC 646:1991, *Information technology – ISO 7-bit coded character set for information interchange.*
- ISO 8859-15:1999, *Information technology – 8-bit single-byte coded graphic character sets – Part 15: Latin alphabet No. 9.*
- ITU-T Recommendation T.84 (1996) | ISO/IEC 10918-3:1997, *Information technology – Digital compression and coding of continuous-tone still images: Extensions.*
- ITU-T Recommendation T.84 (1996)/Amd.1 (1999) | ISO/IEC 10918-3:1997/Amd.1:1999, *Information technology – Digital compression and coding of continuous-tone still images: Extensions – Amendment 1: Provisions to allow registration of new compression types and versions in the SPIFF header.*
- ITU-T Recommendation T.86 (1998) | ISO/IEC 10918-4:1999, *Information technology – Digital compression and coding of continuous-tone still images: Registration of JPEG Profiles, SPIFF Profiles, SPIFF Tags, SPIFF colour Spaces, APPn Markers, SPIFF, Compression types and Registration Authorities (REGAUT).*
- ITU-T Recommendation T.87 (1998) | ISO/IEC 14495-1:2000, *Lossless and near-lossless compression of continuous-tone still images – Baseline.*

2.2 Дополнительные ссылки

- Specification ICC.1:1998-09, *File format for Color Profiles.*
- IEC 61966-2-1:1999, *Multimedia systems and equipment – Colour measurement and management – Part 2-1: Colour management – Default RGB colour space – sRGB.*
- W3C REC-xml-19980210, *Extensible Markup Language (XML 1.0).*
- IETF RFC 2279 (1998), *UTF-8, a transformation format of ISO 10646.*

- ISO/IEC 11578:1996, *Information technology – Open Systems Interconnection – Remote Procedure Call*.
- IEC 61966-2-1:1999/Amd.1:2003, *Multimedia systems and equipment – Colour measurement and management – Part 2-1: Colour management – Default RGB colour space – sRGB*.

3 Определения

В данной Рекомендации | Международном стандарте, используются следующие определения.

- 3.1** $\lfloor x \rfloor$, **нижняя функция**: обозначает наибольшее целое, которое не превосходит x .
- 3.2** $\lceil x \rceil$, **верхняя функция**: обозначает наименьшее целое, которое не достигает x .
- 3.3** **обратимый фильтр 5-3**: специальная пара фильтров, которую используют при передаче сигналов. В обратимой паре фильтров имеется 5 отводов в низкочастотном фильтре и 3 – в высокочастотном.
- 3.4** **необратимый фильтр 9-7**: специальная пара фильтров, которую используют при передаче сигналов. В необратимой паре фильтров имеется 9 отводов в низкочастотном фильтре и 7 – в высокочастотном.
- 3.5** **AND**: битовое представление логического оператора AND.
- 3.6** **арифметический кодер**: статистический кодер, который преобразует строки переменной длины в коды переменной длины и обратно (кодирование и декодирование).
- 3.7** **вспомогательный канал**: канал, который приложение использует вне области задач преобразования цветового пространства. Например, дополнительными могут быть каналы затенения (непрозрачности) или насыщенности.
- 3.8** **бит**: сокращение термина "бинарный знак". Единица информации, представленная нулем или единицей.
- 3.9** **битовая матрица**: двухразмерная матрица битов. В данной Рекомендации | Международном Стандарте предполагается, что в битовой матрице все биты имеют одну и ту же величину для всех коэффициентов или образцов. Это может относиться к компоненту, фрагменту компонента, блоку кодирования, области интересов и др.
- 3.10** **поток битов**: действительная последовательность битов, возникающая в результате кодирования последовательности символов. В эту последовательность в основном не включены маркеры или сегменты маркеров в основных заголовках и заголовках частей тайлов, а также маркер ЕОС. В эту последовательность включаются любые заголовки пакетов и маркеры потоков, а также сегменты маркеров, не обнаруженные в основных заголовках или заголовках частей тайлов.
- 3.11** **биты представления ("big endian")**: биты представления величины, которые встречаются в последовательности от наиболее значащего к наименее значащему биту.
- 3.12** **блок**: часть формата файла, определенная длиной и однозначностью типа блока. Блоки некоторых типов могут включать другие блоки.
- 3.13** **содержимое блока**: относится к данным, заключенным в структуре блока. Содержимое конкретного блока хранится в поле DBox структуры данных блока.
- 3.14** **тип блока**: определяет вид информации, которая должна храниться в блоке. Тип конкретного блока хранится в поле TBox структуры данных блока.
- 3.15** **байт**: восемь битов.
- 3.16** **канал**: один логический компонент изображения. Канал может быть непосредственным представлением одного компонента кодированного потока или может быть создан приложением цветовой гаммы для компонента из кодированного потока.
- 3.17** **проход очистки**: кодирующий проход, выполняемый на единственной матрице битов коэффициентов блока кодирования. Первый, единственный кодирующий проход для первой значащей битовой матрицы является проходом очистки. Третий, последний проход каждой оставшейся битовой матрицы – это проход очистки.
- 3.18** **кодированный поток**: набор из одного или более потоков битов, основного заголовка, фрагментов заголовков и маркеров ЕОС, которые требуются для их декодирования и превращения в данные изображения. Все это является данными изображения в сжатой форме со всей сигнализацией, необходимой для декодирования.
- 3.19** **блок кодирования**: сгруппированные в прямоугольник коэффициенты из одного и того же поддиапазона фрагментов заголовков.
- 3.20** **сканирование блока кодирования**: порядок, в котором просматриваются коэффициенты блока кодирования во время кодирующего прохода. Блок кодирования обрабатывается полосами, каждая из которых состоит из четырех рядов (или всех остающихся рядов, если их число меньше четырех) и занимает ширину блока кодирования. Каждая полоса обрабатывается столбец за столбцом сверху донизу и слева направо.

- 3.21 кодер:** реализация процессов кодирования или декодирования.
- 3.22 кодирующий проход (проход кодирования):** законченный проход через блок кодирования, при котором используют соответствующие величины коэффициентов и содержание. Существует три типа кодирующих проходов: проход воспроизведения значимости, проход подстройки величины и проход очистки. В результате каждого прохода (после арифметического кодирования, если при этом не используется избирательный обход арифметического кодирования) появляется поток данных сжатого изображения.
- 3.23 коэффициент:** величина, которая является результатом преобразования.
- 3.24 канал цвета:** канал, который действует как вход в систему преобразования цвета. Например, каналами цвета являются каналы красного или серого цвета.
- 3.25 компонент:** двухразмерный массив образцов. Как правило, изображение состоит из нескольких компонентов, например, представляющих красный, зеленый и синий цвета.
- 3.26 данные сжатого изображения:** часть или весь поток битов. Может также представлять набор потоков битов в части или в целом кодированном потоке.
- 3.27 согласованный считыватель:** приложение, которое считывает и правильно интерпретирует файл JP2.
- 3.28 контекст:** функция коэффициентов, которые были ранее декодированы и использованы для декодирования представленных коэффициентов.
- 3.29 метка контекста:** произвольный индекс для распознавания различных значений контекста. Эти метки используют скорее для удобства записи, чем в качестве обязательных элементов.
- 3.30 вектор контекста:** бинарный вектор, состоящий из значений положений коэффициентов, включенных в контекст.
- 3.31 декодер:** реализация процесса декодирования и дополнительно процесса преобразования цвета.
- 3.32 процесс декодирования:** процесс, который принимает на входе полностью или частично кодированный поток и выдает на выходе полностью или частично восстановленное изображение.
- 3.33 уровень декомпозиции:** набор поддиапазонов сигналов, в которых каждый коэффициент имеет одно и то же пространственное воздействие или пространственный интервал по отношению к образцам компонентов источника. Это – поддиапазоны HL, LH и HH, которые состоят из одних и тех же двухразмерных декомпозиций поддиапазона. Кроме того, для последнего уровня декомпозиции включается поддиапазон LL.
- 3.34 разграничительные маркеры и сегменты маркеров:** маркеры и сегменты маркеров, которые предоставляют информацию о начальных и конечных точках структур в кодированном потоке.
- 3.35 дискретное волновое (частотное) преобразование (DWT):** преобразование, которое итеративно преобразует один сигнал в два или более фильтрованных и прореженных сигналов, соответствующих разным частотным диапазонам. Это преобразование действует на дискретных интервалах пространства.
- 3.36 кодировщик:** реализация процесса кодирования.
- 3.37 процесс кодирования:** процесс, который принимает на входе полностью или частично источник изображения и выдает на выходе кодированный поток.
- 3.38 формат файла (файловый формат):** кодированный поток и дополнительная поддержка данными и информацией, не явно требуемой для декодирования кодированного потока. Примеры такой поддержки данными составляют поля текста с титрами, информация о защите и предыстория, данные о поддержке размещения множества кодированных потоков внутри заданного файла данных, а также данные для поддержки обмена между платформами или преобразованием в другие файловые форматы.
- 3.39 маркеры неизменяемой информации и неизменяемых сегментов информации:** и те, и другие маркеры дают информацию об оригинале изображения.
- 3.40 функциональные маркеры и сегменты функциональных маркеров:** и те, и другие дают информацию о процедурах кодирования.
- 3.41 разрешение сетки:** пространственное разрешение опорной сетки, определяющее расстояние между соседними точками опорной сетки.
- 3.42 защитные биты:** дополнительные наиболее значащие биты, которые должны быть добавлены к данным образца.
- 3.43 заголовок:** любая часть кодированного потока, которая содержит только маркеры и сегменты маркеров (основной заголовок и заголовок части тайла) или сигнальную часть пакета (заголовок пакета).

3.44 поддиапазон НН: поддиапазон, который получается горизонтальной высокочастотной фильтрацией и вертикальной высокочастотной фильтрацией. Этот поддиапазон содействует реконструкции (восстановлению) с помощью инверсной горизонтальной и вертикальной высокочастотной фильтрации.

3.45 поддиапазон НЛ: поддиапазон, который получается горизонтальной высокочастотной фильтрацией и вертикальной низкочастотной фильтрацией. Этот поддиапазон содействует реконструкции с помощью инверсной горизонтальной высокочастотной фильтрации и вертикальной низкочастотной фильтрации.

3.46 изображение: набор из всех компонентов.

3.47 площадь изображения: прямоугольная часть опорной сетки, зарегистрированная с помощью сдвигов от начала и вдоль опорной сетки.

3.48 сдвиг площади изображения: число точек опорной сетки снизу и справа от начала опорной сетки, где может находиться начало площади изображения.

3.49 данные изображения: компоненты и образцы компонентов, формирующие изображение. Данные изображения могут относиться к любому источнику данных изображения или восстановленным данным изображения.

3.50 маркеры в потоке битов и сегменты маркеров в потоке битов: маркеры и сегменты маркеров, которые функционально обеспечивают устойчивость к ошибкам.

3.51 информационные маркеры и сегменты информационных маркеров: маркеры и сегменты маркеров, которые предоставляют дополнительную информацию.

3.52 необратимость: преобразование, последовательность, система, квантование и другие процессы, которые вследствие системных ошибок или ошибок квантования неспособны к восстановлению. Процесс необратимости может привести только к сжатию с потерями.

3.53 файл JP2: наименование файла в файловом формате, описанном в этой Рекомендации | Международном стандарте. Структурно файл JP2 является непрерывной последовательностью блоков.

3.54 JPEG: используется глобально в отношении процессов кодирования и декодирования в следующих Рекомендациях/Международных стандартах:

- ITU-T Rec. T.81 | ISO/IEC 10918-1, Information technology – Digital compression and coding of continuous-tone still images: Requirements and guidelines;
- ITU-T Rec. T.83 | ISO/IEC 10918-2, Information technology – Digital compression and coding of continuous-tone still images: Compliance testing;
- ITU-T Rec. T.84 | ISO/IEC 10918-3, Information technology – Digital compression and coding of continuous-tone still images: Extensions;
- ITU-T Rec. T.84 | ISO/IEC 10918-3/Amd.1, Information technology – Digital compression and coding of continuous-tone still images: Extensions – Amendment 1: Provisions to allow registration of new compression types and versions in the SPIFF header;
- ITU-T Rec. T.86 | ISO/IEC 10918-4, Information technology – Digital compression and coding of continuous-tone still images: Registration of JPEG Profiles, SPIFF Profiles, SPIFF Tags, SPIFF colour Spaces, APPn Markers, SPIFF, Compression types and Registration Authorities (REGAUT).

3.55 JPEG 2000: используется глобально в отношении процессов кодирования и декодирования в этой Рекомендации | Международном стандарте и в их реализованных приложениях.

3.56 поддиапазон ЛН: поддиапазон, который получается горизонтальной низкочастотной фильтрацией и вертикальной высокочастотной фильтрацией. Этот поддиапазон содействует восстановлению с помощью инверсной горизонтальной низкочастотной фильтрации и вертикальной высокочастотной фильтрации.

3.57 поддиапазон ЛЛ: поддиапазон, который получается горизонтальной низкочастотной фильтрацией и вертикальной низкочастотной фильтрацией. Этот поддиапазон содействует восстановлению с помощью инверсной горизонтальной низкочастотной фильтрации и вертикальной низкочастотной фильтрации.

3.58 слой: набор данных сжатого изображения, полученных с помощью кодирующих проходов из одного или более блоков кодирования компонентов тайла. Следует сохранять порядок, в котором слои кодируются и декодируются.

3.59 без потерь: термин для описания эффекта полных процессов кодирования и декодирования, в которых процесс на выходе декодера идентичен процессу на входе кодера, то есть может быть обеспечено восстановление сигнала без искажений. Все процессы или шаги, используемые для кодирования и декодирования, являются обратимыми.

3.60 с потерями: термин для описания эффекта полных процессов кодирования и декодирования, в которых процесс на выходе декодера не идентичен процессу на входе кодера, то есть имеются искажения (которые можно

- оценить математически). По крайней мере, один из процессов или шагов, используемых для кодирования и декодирования, является необратимым.
- 3.61 проход настройки величины:** тип кодирующего прохода.
- 3.62 основной заголовок:** группа маркеров или сегментов маркеров в начале кодированного потока, описывающая параметры изображения и параметры кодирования, которые можно использовать для любого тайла или компонента тайла.
- 3.63 маркер:** код из двух байтов, в котором первый байт – шестнадцатеричное значение FF (0xFF), а второй байт – значение между 1 (0x01) и FE (0xFE).
- 3.64 сегмент маркера:** маркер и связанное с ним (не пустое) множество параметров.
- 3.65 mod:** $\text{mod}(y,x) = z$, где z – такая величина, что $0 \leq z < x$, и такая, что $y - z$ есть множество x .
- 3.66 пакет:** часть потока битов, состоящая из заголовка пакета и данных сжатого изображения из одного слоя, одной области, одного уровня разрешения и одного компонента тайла.
- 3.67 заголовок пакета:** часть пакета, которая содержит необходимую информацию для декодирования этого пакета.
- 3.68 указатели маркеров и указатели сегментов маркеров:** маркеры и сегменты маркеров, в которых содержится информация о расположении структур кодированного потока.
- 3.69 граничная область:** одна прямоугольная область преобразованного компонента тайла внутри каждого уровня разрешения, которую используют для ограничения размеров пакетов.
- 3.70 точность:** число битов, заключенных в конкретном образце; коэффициент или иное двоичное численное представление.
- 3.71 последовательность:** порядок кодированного потока, при котором декодирование каждого последующего бита приводит к "лучшему" восстановлению изображения. Выбор метрики для "лучшего" восстановления является функцией приложения. Примеры некоторых последовательностей показывают возрастающее разрешение и верность воспроизведения изображения.
- 3.72 квантование:** метод снижения точности индивидуальных коэффициентов для уменьшения числа битов, применяемый для статистического кодирования. Эквивалентно делению при сжатии (компрессии) и умножению при декомпрессии. Квантование может быть выполнено точной операцией с заданной величиной или уменьшением (усечением) числа проходов кодирования по кодированному потоку.
- 3.73 растровый порядок:** особый порядок данных любого типа внутри массива. Растровый порядок начинается с точки данных слева вверху, передвигается непосредственно к правой точке данных и т. д. к концу ряда. После достижения конца ряда, следующей точкой данных последовательности будет крайняя левая точка следующего ряда, расположенного непосредственно под предыдущим. Этот порядок сохраняется до конца массива.
- 3.74 восстановленное (реконструированное) изображение:** изображение, которое появляется на выходе декодера.
- 3.75 восстановленный образец:** восстановленный декодером образец. При кодировании без потерь этот образец всегда равен оригиналу, но может отличаться от него при кодировании с потерями.
- 3.76 опорная сетка:** стандартный прямоугольный массив точек, который используют как опорный для других прямоугольных массивов данных. Примерами являются компоненты и тайлы.
- 3.77 опорный тайл:** прямоугольная вторичная сетка любого размера, связанная с опорной сеткой.
- 3.78 область интересов (ROI):** наборы коэффициентов, которые рассматривают как особо важные по определенным некоторыми пользователями критериям.
- 3.79 уровень разрешения:** эквивалент уровня декомпозиции с одним исключением: поддиапазон LL также является отдельным уровнем разрешения.
- 3.80 обратимость:** преобразование, система или другой процесс, на который не действуют ошибки системные или квантования, поэтому возможно восстановление сигнала без потерь.
- 3.81 образец:** один элемент в двухразмерном массиве, который включает в себя компонент.
- 3.82 символ сегментации:** специальный символ, который для защиты от ошибок кодируют с помощью единообразного контекста в конце каждого прохода кодирования.
- 3.83 избирательный обход арифметического кодирования:** метод кодирования, при котором некоторые блоки кодирования не кодируются арифметическим кодером. Вместо этого биты, которые предназначались для кодирования, присоединяются к потоку битов без кодирования.

- 3.84 сдвиг:** умножение или деление на число в степени два.
- 3.85 знаковый бит:** бит, который показывает знак числа: положительное число (ноль) или отрицательное число (единица).
- 3.86 знаковая система счисления:** двоичное представление целого, при котором расстояние от начала выражается положительным числом, а направление от начала (положительное или отрицательное) выражается отдельным единственным знаковым битом.
- 3.87 проход воспроизведения значимости:** проход кодирования, выполняемый на единственной битовой матрице коэффициентов блока кодирования.
- 3.88 положение значимости:** положение коэффициента в особой битовой матрице. Если в знаковой системе счисления коэффициент имеет первое значение 1 бит в данной или предыдущей битовой матрице, это положение считается "значащим". В противном случае оно считается "незначащим".
- 3.89 источник изображения:** изображение, используемое на входе кодировщика.
- 3.90 поддиапазон:** группа коэффициентов преобразования, полученная в результате обработки одной и той же последовательности низкочастотным и высокочастотным фильтрами, как по горизонтали, так и по вертикали.
- 3.91 коэффициент поддиапазона:** коэффициент преобразования в заданном поддиапазоне.
- 3.92 декомпозиция поддиапазона:** преобразование компонента тайла в поддиапазоны.
- 3.93 суперблок:** блок, в котором содержится последовательность смежных блоков (и только последовательность смежных блоков). Поскольку файл JP2 содержит только последовательность смежных блоков, этот файл рассматривают как суперблок. Когда рассматривают взаимосвязь между двумя блоками, термин "суперблок" относится к тому блоку, который непосредственно содержит другой блок.
- 3.94 тайл (tile):** элемент мозаичного изображения – прямоугольный массив точек на опорной сетке, зарегистрированный и смещенный относительно начала опорной сетки и определенный по длине и высоте.
- 3.95 компонент тайла:** все образцы данного компонента в тайле.
- 3.96 индекс тайла:** индекс текущего тайла в диапазоне от нуля до номера этого элемента минус единица.
- 3.97 часть тайла (tile-part):** часть кодированного потока с данными сжатого изображения для некоторых, всех или одного тайла. Часть тайла включает, по крайней мере, один или все пакеты, которые заполняют кодированный тайл.
- 3.98 заголовок части тайла:** группа маркеров или сегментов маркеров в начале каждой части тайла в кодированном потоке, которая описывает параметры кодирования тайла.
- 3.99 индекс части тайла:** индекс текущей части тайла в данном тайле с диапазоном от нуля до числа частей тайла минус единица.
- 3.100 преобразование:** математическое перемещение из одного пространства сигналов в другое.
- 3.101 коэффициент преобразования:** величина, которая получается в результате преобразования.
- 3.102 XOR:** логический оператор – "исключающее ИЛИ".

4 Сокращения и символы

4.1 Сокращения

В данной Рекомендации | Международном стандарте, используются следующие сокращения.

МККТ	Международный консультативный комитет по телеграфии и телефонии, сейчас МСЭ-Т
ИСС	Международный консорциум цвета
ИСТ	Необратимое преобразование компонента
МЭК	Международная электротехническая комиссия

ИСО	Международная организация по стандартизации
ITTF	Рабочая группа по информационной технологии
МСЭ	Международный союз электросвязи
МСЭ-Т	Международный союз электросвязи – Сектор стандартизации электросвязи (бывший МККТ)
JPEG	Объединенная группа экспертов по машинной обработке фотографий – Объединенный комитет ИСО/МСЭ, ответственный за разработку стандартов для кодирования монотонных неподвижных изображений. Отсылаем также к стандартам, разработанным этим комитетом: Рекомендация МСЭ-Т. Т.81 ИСО/МЭК 10918-1, Рекомендация МСЭ-Т. Т.83 ИСО/МЭК 10918-2, Рекомендация МСЭ-Т. Т.84 ИСО/МЭК 10918-3 и Рекомендация МСЭ-Т. Т.87 ИСО/МЭК 14495.
JURA	Организация по регистрации применений JPEG
1D-DWT	Одноразмерное дискретное частотное (волновое) преобразование
FDWT	Прямое дискретное частотное (волновое) преобразование
IDWT	Инверсное дискретное частотное (волновое) преобразование
LSB	Наименьший значащий бит (разряд)
MSB	Наибольший значащий бит (разряд)
PCS	Пространственные характеристики соединения
RCT	Обратимое преобразование компонента
ROI	Область интересов
ОСШ	Отношение сигнал/шум
UCS	Универсальный набор символов
URI	Единообразный идентификатор ресурса
URL	Единообразный локатор ресурса
UTF-8	Формат 8 преобразования UCS
UUID	Универсальный идентификатор однозначности
XML	Язык расширенной разметки
W3C	Консорциум всемирной сети www

4.2 Символы

В данной Рекомендации | Международном стандарте, используются следующие символы.

0x---	Обозначает шестнадцатеричное число
$\backslash mnn$	Трехразрядное число после наклонной черты показывает значение единственного байта в строке знаков; три знака определяют восьмеричное значение этого байта
ϵ_b	Показатель величины квантования для поддиапазона, определенного в маркерах QCD и QCC
μ_b	Мантисса величины квантования для поддиапазона, определенного в маркерах QCD и QCC
M_b	Максимальное число битовых матриц, закодированных в заданном блоке кодирования
N_L	Число уровней декомпозиции, определенных в маркерах COD и COC
R_b	Динамический диапазон в образце компонента, определенного в маркере SIZ
COC	Маркер типа кодирования компонента
COD	Маркер общепринятого (по умолчанию) типа кодирования
COM	Маркер комментария
CRG	Маркер регистрации компонента
EPH	Маркер конца заголовка пакета
EOC	Маркер конца кодированного потока
PLM	Маркер основного заголовка и длины пакета
PLT	Маркер заголовка и длины части тайла
POC	Маркер изменения порядка последовательности

PPM	Маркер основного заголовка и заголовков упакованных пакетов
PPT	Маркер заголовков упакованных пакетов и заголовка части тайла
QCC	Маркер квантования компонента
QCD	Маркер общепринятого (по умолчанию) квантования
RGN	Маркер области интересов
SIZ	Маркер размеров изображения и тайла
SOC	Маркер запуска кодированного потока
SOP	Маркер запуска пакета
SOD	Маркер запуска данных
SOT	Маркер запуска части тайла
TLM	Маркер длины части тайла

5 Общее описание

В этой Рекомендации | Международном стандарте описана система сжатия изображения, которая предоставляет большую гибкость не только для сжатия изображений, но также для доступа к кодированному потоку. Кодированный поток обладает рядом механизмов для размещения и извлечения частей данных сжатого изображения, для их переприема, хранения, отображения или редактирования. Такой подход дает возможность заданному приложению хранить и восстанавливать данные сжатого изображения без декодирования.

Разделение рядом способов данных оригинала изображения и данных сжатого изображения позволяет извлечь данные изображения из данных сжатого изображения для воспроизведения оригинала изображения с меньшим разрешением или с меньшей точностью. Это также позволяет лучше совместить кодированный поток с каналом передачи, устройством хранения или отображения, независимо от размера, числа компонентов и точности образца оригинала изображения. Для более эффективного использования в заданном приложении кодированным потоком можно управлять без декодирования.

Таким образом, расширенные возможности данной Рекомендации | Международного стандарта позволят более эффективно использовать один кодированный поток для ряда приложений. Например, источник изображения большого размера может создать кодированный поток, который будет легко обработан для воспроизведения изображения на миниатюрном дисплее.

В общем, данная Рекомендация | Международный стандарт рассматривает три области: пространственную (образцы), преобразовательную (коэффициенты) и данные сжатого изображения. Некоторые объекты (например, компоненты тайла) играют роль во всех трех областях. Другие (например, блоки кодирования или пакеты) играют роль только в одной области (например, соответственно данные преобразованного или сжатого изображения). Разделение объекта на другие объекты в одной и той же области (например, компонента на компоненты тайла) описано отдельно для каждой из этих областей.

5.1 Цель

В этой Рекомендации | Международном стандарте описаны четыре основных элемента:

- Кодировщик: устройство для реализации процесса кодирования. Кодировщик принимает на входе цифровые данные источника изображения и спецификацию параметров и с помощью ряда процедур вырабатывает на выходе кодированный поток.
- Декодер: устройство реализации процесса декодирования. Декодер принимает на входе данные сжатого изображения и спецификацию параметров и с помощью ряда специальных процедур вырабатывает на выходе данные восстановленного изображения.
- Синтаксис кодированного потока: представление данных сжатого изображения, которое включает все спецификации параметров, требуемые для процесса декодирования.
- Дополнительный файловый формат: дополнительный файловый формат требуется для обмена между приложениями окружения. Кодированный поток может быть использован другими файловыми форматами или самостоятельно без этого файлового формата.

5.2 Кодированный поток

Кодированный поток – это линейный поток битов от первого бита до последнего. Для удобства поток можно разделить на байты (8 битов), начиная с первого бита потока, то есть с "раннего" бита в байте, рассматриваемого как наиболее значимый бит этого байта, например, в шестнадцатеричном представлении. Этот поток байтов можно разделить на группы следующих друг за другом байтов. Иногда шестнадцатеричное представление неявно подразумевается в тексте при описании байтов или групп байтов, которые не имеют "естественного" численного представления.

5.3 Принципы кодирования

Основные процедуры этой Рекомендации | Международного стандарта показаны на рисунке 5-1. Там изображен только порядок декодирования. Данные сжатого изображения уже концептуально соотнесены с частями данных изображения. Процедуры представлены в Приложениях в порядке процесса декодирования. Ниже подводится итог процесса кодирования.

ПРИМЕЧАНИЕ 1. – Приложения от А до I рассматриваются как обязательные (нормативные) для этой Рекомендации | Международного стандарта. Однако некоторые подразделы с описаниями, примечания, а также все примеры являются информативными.

Многие изображения состоят из множества компонентов. В этой Рекомендации | Международном стандарте используется сложное преобразование компонентов для декорреляции трех компонентов. Это единственная функция в данной Рекомендации | Международном стандарте, которая соотносит компоненты между собой. (См. Приложение 5.)

Компоненты изображения могут быть разделены на тайлы. Компоненты тайлов представляют собой прямоугольные массивы, связанные с той же самой частью каждого из компонентов, из которых составляется изображение. Таким образом, разделение изображения на тайлы фактически создает компоненты тайла, которые можно извлечь и декодировать независимо друг от друга. Эта независимость и представляет один из методов извлечения области изображения. (См. Приложение В.)

Компоненты тайла разлагают на разные уровни декомпозиции, используя частотное преобразование. Эти уровни декомпозиции содержат несколько поддиапазонов с коэффициентами, которые описывают горизонтальные и вертикальные пространственные частотные характеристики оригиналов компонентов тайла. Подобно преобразованию Фурье эти коэффициенты содержат частотную информацию о локальной области в большей степени, чем целиком об изображении, то есть небольшое количество коэффициентов полностью описывает один образец. Один уровень декомпозиции связан со следующим уровнем пространственным коэффициентом два, то есть каждый последующий уровень декомпозиции поддиапазона имеет приблизительно вдвое меньшее разрешение по горизонтали и вертикали, чем предыдущий. Изображения с более низким, чем у оригинала, разрешением воспроизводятся декодированием подмножества выбранных поддиапазонов. (См. Приложение F.)

Хотя коэффициентов столько же, сколько и образцов, информационное содержание заключено преимущественно только в нескольких коэффициентах. При квантовании информационное содержание большого числа малозначительных коэффициентов еще более снижается (Приложение E). Дополнительная обработка статистическим кодером уменьшает число битов, которое требуется для представления этих квантованных коэффициентов, иногда значительно по сравнению с оригиналом изображения. (См. Приложения C, D и B.)

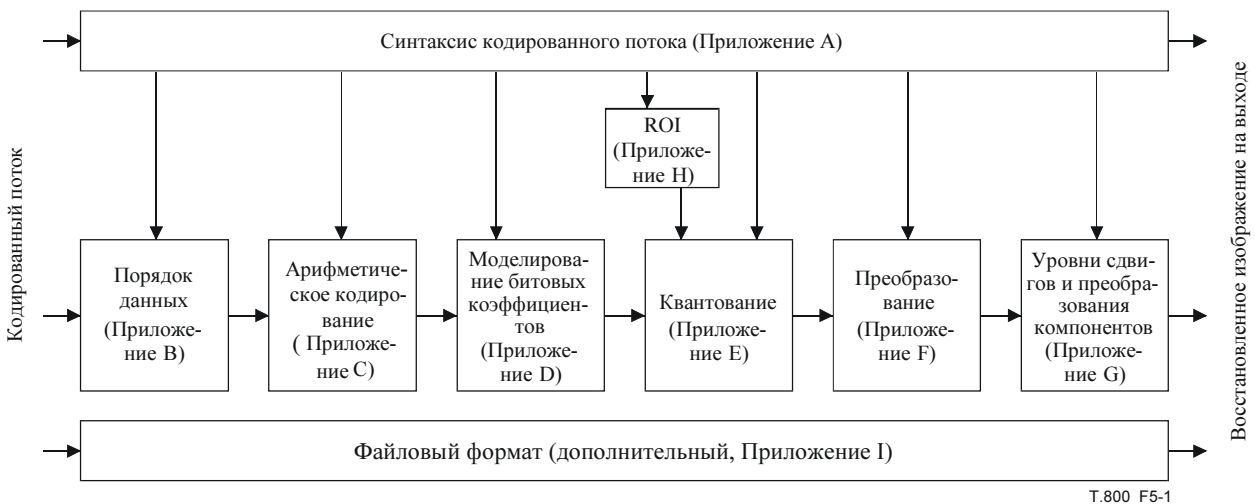


Рисунок 5-1 – Подробная блок-диаграмма

В дальнейшем отдельные поддиапазоны компонента тайла разбиваются на блоки кодирования. Прямоугольные матрицы коэффициентов должны извлекаться независимо. Отдельные битовые матрицы коэффициентов в этих блоках кодируются тремя проходами кодирования. При каждом из этих проходов кодирования собирается зависящая от контекста информация о битовой матрице данных сжатого изображения. (См. Приложение D.) Арифметический кодер использует эту информацию и ее внутреннюю структуру для декодирования сжатого потока битов. (См. Приложение С.) Различные завершающие механизмы создают из независимо извлеченных различных уровней данные сжатого изображения.

Поток битов данных сжатого изображения, созданный этими кодирующими проходами, группируется в слои. Слои представляют произвольную группировку из блоков кодирования, полученных в результате проходов кодирования. (См. Приложение В.)

ПРИМЕЧАНИЕ 2. – Хотя допускается большая гибкость в организации слоев, основное условие заключается в том, чтобы каждый последующий слой улучшал качество изображения.

Коэффициенты поддиапазонов для каждого уровня разрешения разделяют на прямоугольные площади, которые называются граничными областями. (См. Приложение В.)

Пакеты являются основополагающими единицами сжатого кодированного потока. Пакет содержит данные сжатого изображения из одного слоя, одной области, одного уровня разрешения и одного компонента тайла. Пакеты предоставляют другой способ извлечения пространственной области независимо от кодированного потока. В кодированном потоке эти пакеты перемежаются несколькими разными методами. (См. Приложение В.)

Представлен механизм, который позволяет кодировать данные сжатого изображения, соответствующие области интересов в оригинале компонентов тайла и размещать их заранее в потоке битов. (См. Приложение Н.)

Представлены несколько механизмов, позволяющих обнаруживать и исправлять битовые ошибки, которые могут появляться в каналах с высоким уровнем шума. (См. D.5 и J.7.)

Относящийся к тайлу кодированный поток состоит из пакетов, размещенных в одной или более частях тайла. Заголовок части тайла состоит из ряда маркеров и сегментов маркеров и содержит информацию о различных механизмах и методах кодирования, которые необходимы, чтобы разместить, извлечь, декодировать и восстановить каждый компонент тайла. В начале целого кодированного потока расположен основной заголовок, который состоит из маркеров и сегментов маркеров и содержит аналогичную информацию, а также информацию об оригинале изображения. (См. Приложение А.)

Кодированный поток дополнительно свертывается в файловый формат, который позволяет приложениям интерпретировать значение информации об изображении, а также значение другой информации. Помимо данных о кодированном потоке, этот файловый формат может содержать и другие данные (см. Приложение I).

Ниже кратко перечислены процедуры, на которые подразделяются действия над оригиналом изображения:

- Компоненты изображения разделяют на прямоугольные тайлы. Компонент тайла является базовой единицей оригинала изображения или восстановленного изображения.
- Над компонентом тайла выполняют частотное (волновое) преобразование, которое создает уровни декомпозиции.
- Эти уровни декомпозиции состоят из поддиапазонов коэффициентов, которые описывают частотные характеристики локальных участков компонентов тайла (в большей степени, чем описывают собственно компонент тайла).
- Поддиапазоны коэффициентов квантуют и собирают в прямоугольные массивы блоков кодирования.
- Каждую матрицу коэффициентов в блоке кодирования кодируют статистически тремя типами кодирующих проходов.
- Некоторые из этих коэффициентов могут кодироваться заранее, чтобы обозначить область интересов.

После этих действий данные изображения полностью превращены в данные сжатого изображения. Следующие процедуры соединяют потоки битовых знаков в кодированные потоки:

- После проходов кодирования данные сжатого изображения объединяют в слои.
- Из сжатых данных одной области, одного слоя, одного уровня разрешения и одного компонента тайла формируют пакеты. Эти пакеты являются базовой единицей данных сжатого изображения.
- Все пакеты одного тайла перемежают одним из нескольких способов и размещают в одной или более частях тайла.
- Части тайла содержат описательный заголовок части тайла и могут перемежаться несколькими способами.
- Кодирующий поток имеет в начале основной заголовок, в котором описан оригинал изображения и различные способы декомпозиции и кодирования.
- Содержание изображения и его компонентов в контексте данного приложения описывают в дополнительном файловом формате.

6 Требования к кодировщику

Процесс кодирования превращает данные источника изображения в данные сжатого изображения. Процесс кодирования описан в Приложениях А, В, С, D, E, F, G, и H. Все процессы кодирования определены как информативные.

Кодировщик является реализацией процесса кодирования. Чтобы соответствовать этой Рекомендации | Международному стандарту, кодировщик должен превращать данные источника изображения в данные сжатого изображения, которые удовлетворяют синтаксису кодированного потока, описанному в Приложении А.

7 Требования к декодеру

Процесс декодирования превращает данные сжатого изображения в данные восстановленного изображения. Процесс декодирования описывается и специфицируется в Приложениях с А до Н. Все процессы декодирования являются нормативными (обязательными).

Декодер является реализацией процесса декодирования. Чтобы соответствовать этой Рекомендации | Международному стандарту, декодер должен для восстановления изображения преобразовывать все или выделенные части любых данных сжатого изображения, которые удовлетворяют синтаксису кодированного потока, описанному в Приложении А.

Не существует нормативных или необходимых требований по применению кодировщика и декодера. В некоторых случаях исключительно для иллюстративных целей используют отдельные методы применения.

7.1 Требования к синтаксису кодированного потока

В Приложении А описан синтаксис кодированного потока, который представляет данные сжатого изображения для обмена между приложениями. Любые данные сжатого изображения должны удовлетворять присвоенному синтаксису и коду, соответствующему процессу кодирования, который определен данной Рекомендацией | Международным стандартом.

В эту Рекомендацию | Международный стандарт не включены определения соответствия и совместимости. Значения параметров синтаксиса, описанные в Приложении А, не предназначены для предоставления возможностей, которые требуются для соответствия.

7.2 Требования к дополнительному файловому формату

В Приложении I описан дополнительный файловый формат, в котором содержатся метаданные об изображении, в дополнение к данным кодированного потока. Эти данные позволяют, например, выводить на дисплей или печатать со специальным разрешением. Там где используется дополнительный файловый формат, он должен соответствовать синтаксису и коду файлового формата, которые предназначены для процессов кодирования и определены в данной Рекомендации | Международном стандарте.

8 Требования к применению

Данная Рекомендация | Международный стандарт не содержит нормативных или необходимых требований по применению. В некоторых случаях исключительно для иллюстративных целей используются отдельные методы применения.

Приложение А

Синтаксис кодированного потока

(Это Приложение составляет неотъемлемую часть данной Рекомендации | Международного стандарта)

В этом Приложении и всех его подразделах блок схемы и таблицы являются нормативными (обязательными) только в том смысле, что они определяют выходные данные, которые при альтернативных применениях должны повторяться.

В Приложении описывается синтаксис и семантика маркера и сегментов маркера, которые определяются этой Рекомендацией | Международным стандартом. Маркеры и сегменты маркеров представляют кодированный поток информации для этой Рекомендации | Международного стандарта. Кроме того, в Приложении представлены синтаксис маркера и сегмента маркера, разработанных для предполагаемого использования в будущих спецификациях, которые включают эту Рекомендацию | Международный стандарт в качестве обязательной ссылки.

В данную Рекомендацию | Международный стандарт не включено определение соответствия или совместимости. Значение параметров синтаксиса, описанного в этом Приложении, не предполагает раскрытие возможностей, необходимых для совместимости.

А.1 Маркеры, сегменты маркеров и заголовки

В этой Рекомендации | Международном стандарте маркеры и сегменты маркеров используют для разграничений и передачи характеристик источника изображения и кодированного потока. Этот набор маркеров и сегментов маркеров является минимальной информацией, необходимой для выявления особенностей этой Рекомендации | Международного стандарта, и не является файловым форматом. Минимальный файловый формат представлен в Приложении I.

Основной заголовок и заголовок элемента изображения представляют набор маркеров и сегментов маркеров. Основной заголовок находится в начале кодированного потока. Заголовки частей тайла находятся в начале каждой части тайла (см. ниже). Некоторые маркеры и сегменты маркеров ограничены только одним из двух типов заголовков, в то время как в других может использоваться любой.

Длина каждого маркера составляет 2 байта. Первый байт состоит из единственного байта 0xFF. Второй байт описывает конкретный маркер и может иметь любую длину в диапазоне от 0x01 до 0xFE. Многие из этих маркеров уже использовались в Рекомендации МСЭ-Т Т.81 | ИСО/МЭК 10918-1 и Рекомендация МСЭ-Т Т.84 | ИСО/МЭК 10918-3 и должны рассматриваться как резервные, если их не используют специально.

Сегмент маркера включает маркер и связанные с ним параметры, которые называются параметрами сегмента маркера. В каждом сегменте маркера первые два байта после маркера должны иметь значение без знака, которое выражает длину в байтах параметров сегмента маркера (включая два байта длины параметра, но не два байта собственно маркера). Когда в кодированном потоке появляется сегмент маркера, который не описан в этой Рекомендации | Международном стандарте, декодер должен использовать параметр длины, чтобы удалить сегмент маркера.

А.1.1 Типы маркеров и сегментов маркеров

Используют шесть типов маркеров и сегментов маркеров: разграничительные, неизменяемой (фиксированной) информации, функциональные, встроенные в поток битов, указательные и информационные. Разграничительные маркеры и сегменты маркеров используют для отделения основного заголовка и заголовка части тайла от потока битов данных. Сегменты маркеров неизменяемой информации дают требуемую информацию об изображении. Расположение этих сегментов маркеров, также как и разграничительных маркеров и сегментов маркеров, определено. Сегменты функциональных маркеров используют для описания выбранной функции кодирования. Маркеры и сегменты маркеров, встроенные в поток битов, используют для защиты от ошибок. Сегменты маркеров указателей указывают на конкретные смещения в потоке битов. Сегменты информационных маркеров дают дополнительную информацию.

А.1.2 Синтаксис, аналогичный синтаксису Рекомендации МСЭ-Т Т.81 | ИСО/МЭК 10918-1

В синтаксисе маркеров и сегментов маркеров используют ту же самую конструкцию, которая определена в Рекомендации МСЭ-Т Т.81 | ИСО/МЭК 10918-1.

Диапазон маркера от 0xFF30 до 0xFF3F зарезервирован данной Рекомендацией | Международным стандартом для маркеров без параметров сегмента маркеров. В таблице А.1 показано, в какой спецификации описываются эти маркеры и сегменты маркеров.

Таблица А.1 – Определения маркеров

Диапазон кода маркера	Стандарт определения
от 0xFF00, 0xFF01, 0xFFFE, 0xFFC0 до 0xFFDF	Определен в Рекомендация МСЭ-Т Т.81 ИСО/МЭК 10918-1
от 0xFFFF0 до 0xFFFF6	Определен в Рекомендация МСЭ-Т Т.84 ИСО/МЭК 10918-3
от 0xFFFF7 до 0xFFFF8	Определен в Рекомендация МСЭ-Т Т.87 ИСО/МЭК 14495-1
от 0xFF4F до 0xFF6F, от 0xFF90 до 0xFF93	Определен в данной Рекомендации Международном стандарте
от 0xFF30 до 0xFF3F	Зарезервирован для определения только в качестве маркеров (не сегментов маркеров)
	Все прочие значения зарезервированы

А.1.3 Правила для маркера, сегмента маркера и кодированного потока

- Сегменты маркеров, а следовательно основной заголовок и заголовок части тайла, представляют множество из 8 битов (одного байта). Далее потоки битов данных от заголовка и до маркера ЕОС (см. А.4.4) также должны быть заполнены 8 битовыми знаками.
- Все сегменты маркеров в заголовке части тайла используют только для того тайла, к которому они принадлежат.
- Все сегменты маркеров в основном заголовке используют для всего изображения, если только это специально не отменено маркерами или сегментами маркеров в заголовке части тайла.
- Маркеры и сегменты маркеров разграничительных и неизменяемой информации обязательно должны располагаться на определенных позициях в кодированном потоке.
- Сегменты маркеров должны корректно описывать изображение, которое представляет кодированный поток. Если производится сокращение, изменение или редактирование кодированного потока, сегменты маркеров должны обновляться, если это необходимо.
- Все величины параметров в сегментах маркеров являются битами представления ("big endian").
- Сегменты маркеров в данном заголовке могут располагаться в любом порядке. Исключения составляют разграничительные маркеры и сегменты этих маркеров, а также сегменты маркеров неизменяемой информации.
- Все маркеры с кодом маркера от 0xFF30 до 0xFF3F не имеют параметров сегментов маркеров. Они должны пропускаться декодером.
- Некоторые сегменты маркеров имеют значения, приписанные к группам битов внутри параметра. В некоторых случаях существуют биты, обозначенные "х", для которых не предусмотрены значения ни в одном поле внутри параметра. Для всех таких битов кодированный поток должен содержать значение нуля. Декодер должен игнорировать такие биты.

ПРИМЕЧАНИЕ. – Маркеры в диапазоне от 0xFF30 до 0xFF3F могут быть использованы в будущих расширениях. Они могут пропускаться или не пропускаться декодером без ответвления.

А.1.4 Ключ к графическим описаниям (информативное сообщение)

Каждый сегмент маркера описан в понятиях его функции, применения и длины. Функция описывает информацию, которая содержится в сегменте маркера. Применение описывает логическое расположение и частоту этого сегмента маркера в кодированном потоке. Длина описывает, какие параметры определяют длину сегмента маркера.

Далее представлен рисунок, на котором показан порядок и взаимосвязь параметров сегмента маркера. Рисунок 1 дает пример такого типа. Сегменты маркера обозначены трехбуквенным кодом маркера, связанного с сегментом маркера. Символы параметра обозначены заглавными буквами, за которыми следуют символы маркера строчными буквами. Чтобы показать расположение параметров в сегменте маркера, используют прямоугольник. Ширина этого прямоугольника пропорциональна числу байтов параметра. Затененная часть прямоугольника (с диагональными полосами) обозначает параметр переменной величины. Два параметра с верхними индексами и серой площадкой между ними показывают диапазон некоторых из этих параметров.

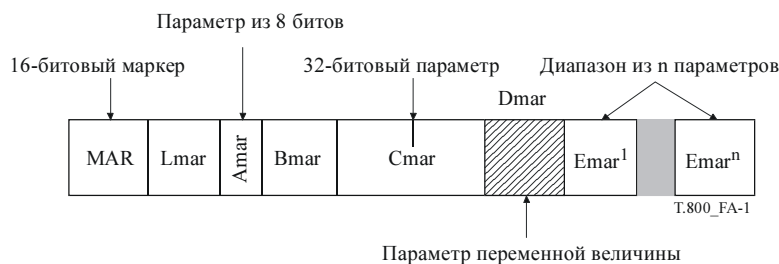


Рисунок А.1 – Пример описания составляющих сегмента маркера

ИСО/МЭК 15444-1:2002 (R)

За рисунком следует список, в котором приведено значение каждого параметра сегмента маркера. Если параметры повторяются, длина и диапазон параметров описываются. В качестве примера на рисунке А.1 первый прямоугольник представляет маркер с символом MAR. Второй прямоугольник представляет длину параметра. Параметры переменной длины Aмар, Bмар, Cмар и Dмар имеют соответственно длину 8, 16 и 32 бита. Чтобы показать существование *n* разных параметров последовательности Eмар¹, применяют систему счисления Eмар¹.

После списка следует таблица, в которой либо описывают разрешенные величины параметров, либо дают ссылки на другие таблицы, где описаны эти величины. Чтобы описать любой параметр без его численной величины, приведены таблицы индивидуальных параметров. В некоторых случаях эти параметры описаны величиной бита в поле битов. В этом случае для обозначения битов используется знак "x", который не включен в спецификацию этого параметра или субпараметра в соответствующей строке таблицы.

Некоторые параметры сегментов маркеров описаны с использованием системы счисления "Sxxx" и "SPxxx" (для символа маркера XXX). Параметр Sxxx выбирает параметр SPxxx среди множества возможных значений. В соответствии с этим выбором модифицируют параметр SPxxx или список параметров.

А.2 Информация в сегментах маркеров

В таблице А.2 перечислены маркеры, специфицированные в этой Рекомендации | Международном стандарте. В таблице А.3 указано, какой вид информации представлен маркером или сегментом маркера.

Таблица А.2 – Список маркеров и сегментов маркеров

	Символ	Код	Основной заголовок	Заголовок части тайла
Разграничительные маркеры и сегменты маркеров				
Запуск кодированного потока	SOC	0xFF4F	требуемый ^{а)}	не разрешенный
Запуск части тайла	SOT	0xFF90	не разрешенный	требуемый
Запуск данных	SOD	0xFF93	не разрешенный	последний маркер
Конец кодированного потока	EOC	0xFFD9	не разрешенный	не разрешенный
Сегменты маркера неизменяемой информации				
Размеры изображения и тайла	SIZ	0xFF51	требуемый	не разрешенный
Сегменты функционального маркера				
Тип кодирования по умолчанию	COD	0xFF52	требуемый	дополнительный
Тип кодирования компонента	COC	0xFF53	дополнительный	дополнительный
Область интересов	RGN	0xFF5E	дополнительный	дополнительный
Квантование по умолчанию	QCD	0xFF5C	требуемый	дополнительный
Квантование компонента	QCC	0xFF5D	дополнительный	дополнительный
Изменение порядка последовательности ^{б)}	POC	0xFF5F	дополнительный	дополнительный
Сегменты маркера указателя				
Длина частей тайла	PLM	0xFF57	дополнительный	не разрешенный
Длина пакета, основной заголовок	PLT	0xFF58	не разрешенный	дополнительный
Длина пакета, заголовок части тайла	PPM	0xFF60	дополнительный	не разрешенный
Заголовки упакованных пакетов, основной заголовок ^{с)}	PPT	0xFF61	не разрешенный	дополнительный
Заголовки упакованных пакетов, заголовок части тайла ^{с)}	TLM	0xFF55	дополнительный	не разрешенный
Маркеры внутри потока битов и сегменты маркера				
Запуск пакета	SOP	0xFF91	не разрешенный	не разрешенный в заголовке части тайла, дополнительный в потоке битов
Конец заголовка пакета	EPH	0xFF92	дополнительный внутри сегмента маркера PPM	дополнительный внутри сегмента маркера PPT или в потоке битов
Сегменты информационного маркера				
Регистрация компонента	CRG	0xFF63	дополнительный	не разрешенный
Комментарий	COM	0xFF64	дополнительный	дополнительный
а) "требуемый" означает, что маркер или сегмент маркера должен быть в этом заголовке, "дополнительный" означает, что эти параметры могут быть использованы.				
б) Если меняется порядок последовательности, требуется маркер POC.				
с) Если заголовки пакетов не распределены в потоке битов, то не требуется ни сегмент маркера PPM, ни PPT. Если используется сегмент маркера PPM, то сегмент маркера PPT не должен использоваться, и наоборот.				

Таблица А.3 – Информация в сегментах маркеров

Информация	Сегмент маркера
Возможности Область изображения или размер опорной сетки (высота и ширина) Размер тайла (высота и ширина) Число компонентов Точность компонентов Расположение компонентов по отношению к опорной сетке (расстановка субобразцов)	SIZ
Индекс тайла Данные длины части тайла	SOT, TLM
Порядок последовательности Число слоев Множества использованных преобразований компонента	COD
Тип кодирования Число уровней декомпозиции Размер блока кодирования Тип блока кодирования Частотное преобразование Размер граничной области	COD, COC
Сдвиг области интересов	RGN
Отсутствие квантования Вторичное квантование Интерпретация квантования	QCD, QCC
Точка начала последовательности Точка конца последовательности Порядок последовательности по умолчанию	POC
Защита от ошибок	SOP
Конец заголовка пакета	EPH
Заголовки пакетов	PPM, PPT
Длина пакетов	PLM, PLT
Регистрация компонентов Дополнительная информация	CRG COM

А.3 Структура кодированного потока

На рисунке А.2 показана структура кодированного потока. На рисунке А.3 показана структура основного заголовка пакета. Все сплошные линии указывают на требуемые сегменты маркеров. Следующие маркеры и сегменты маркеров требуют специального расположения: SOC, SIZ, SOT, SOD и EOC. Пунктирные линии показывают дополнительные или возможные, но не требуемые сегменты маркеров. На рисунке А.4 показана структура первого заголовка части данного тайла. На рисунке А.5 показана структура заголовка части тайла, который отличается от первого заголовка.

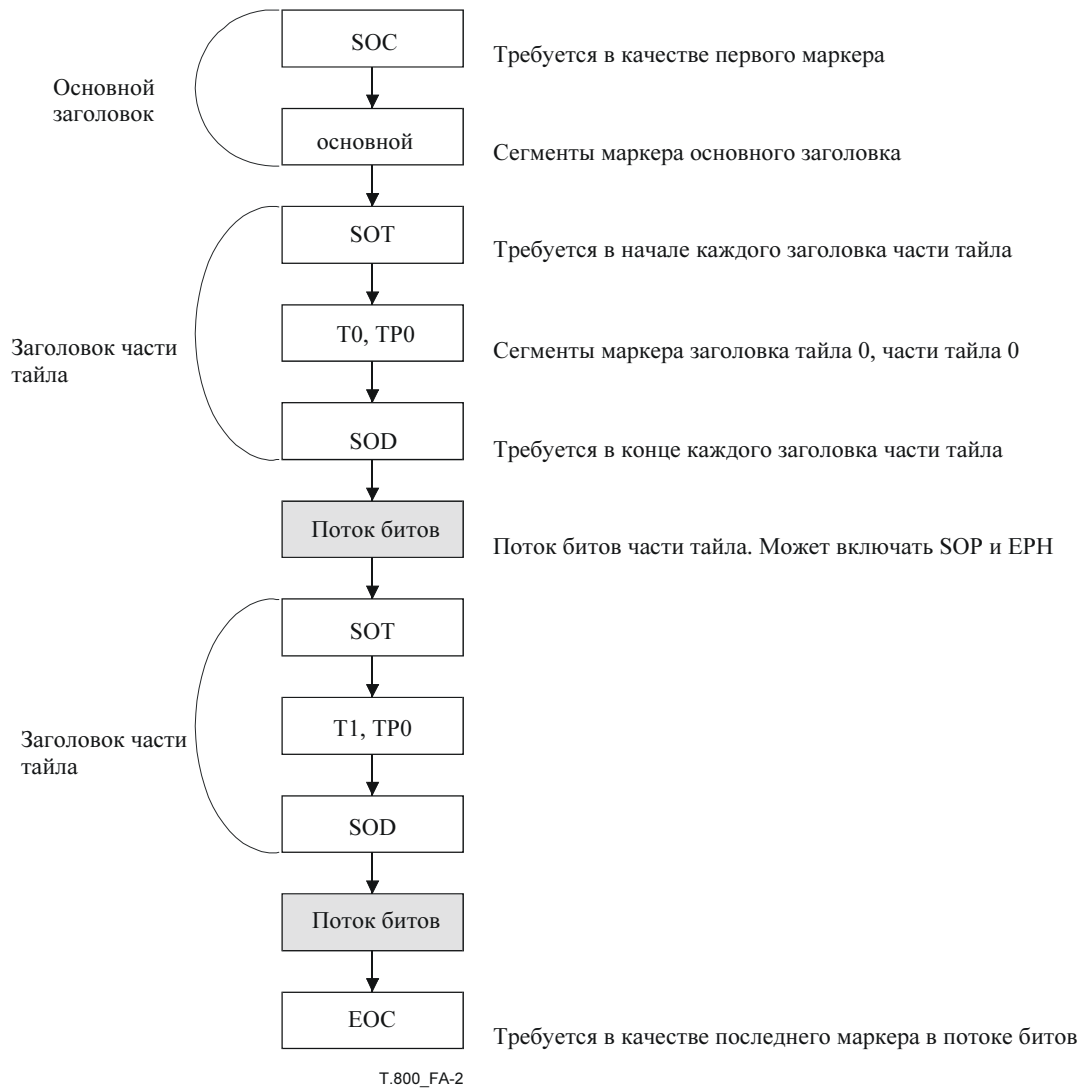


Рисунок А.2 – Структура кодированного потока

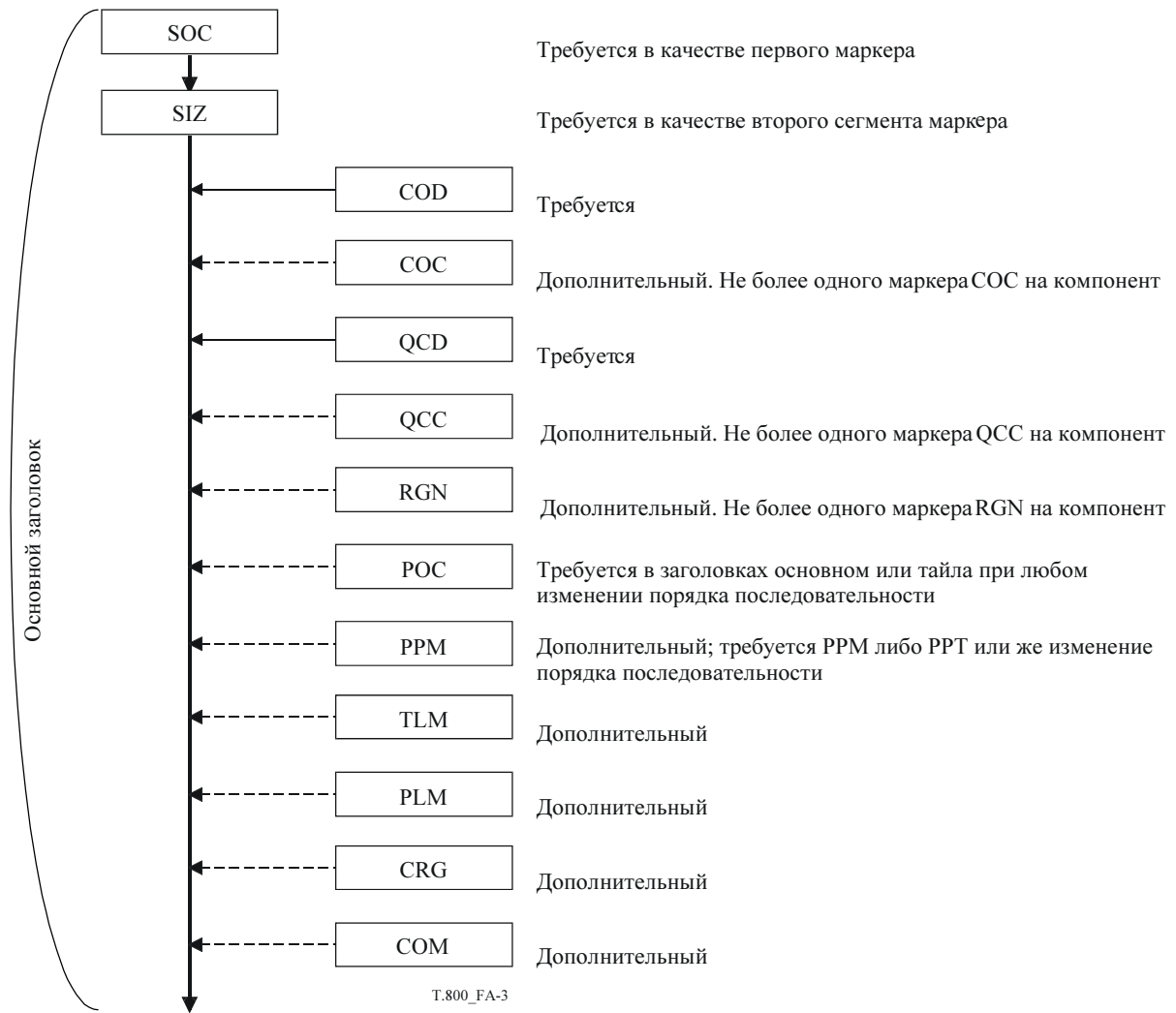


Рисунок А.3 – Структура основного заголовка

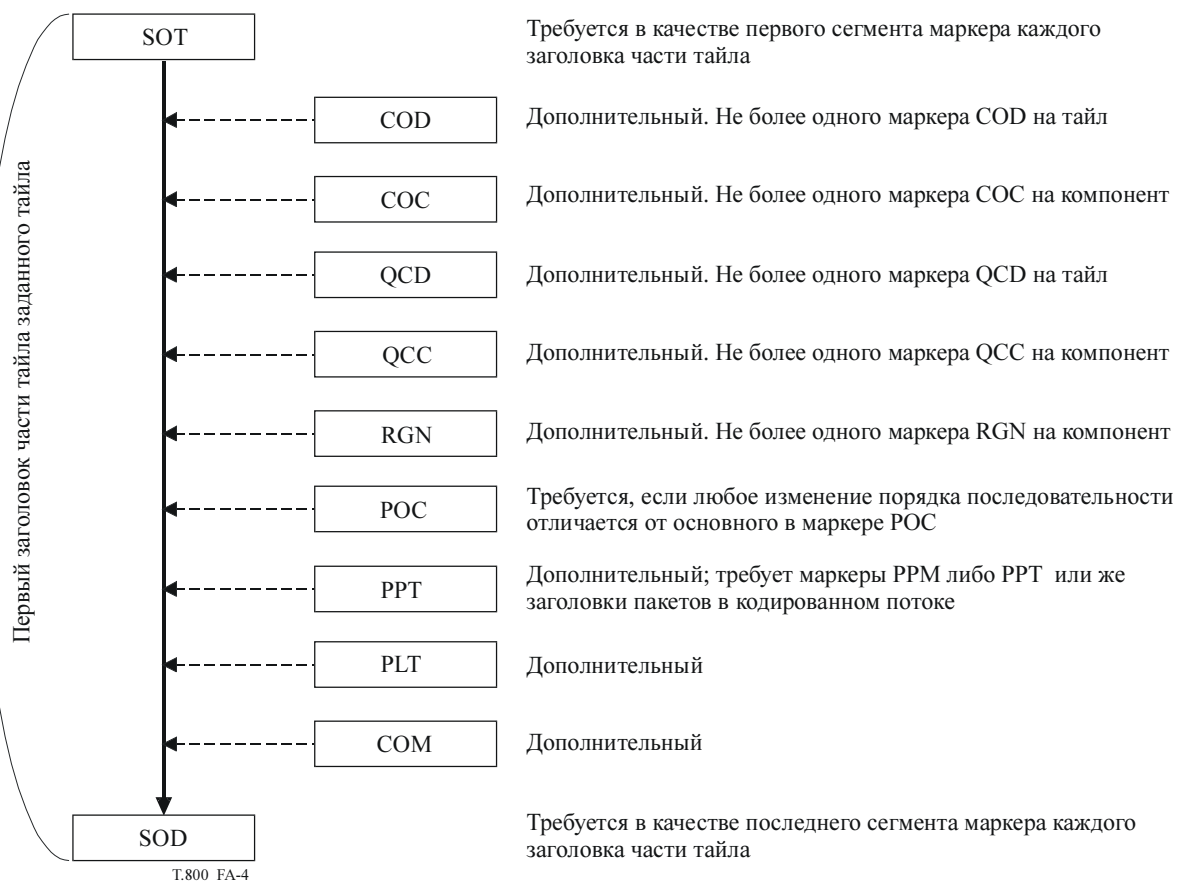


Рисунок А.4 – Структура первого заголовка части данного тайла

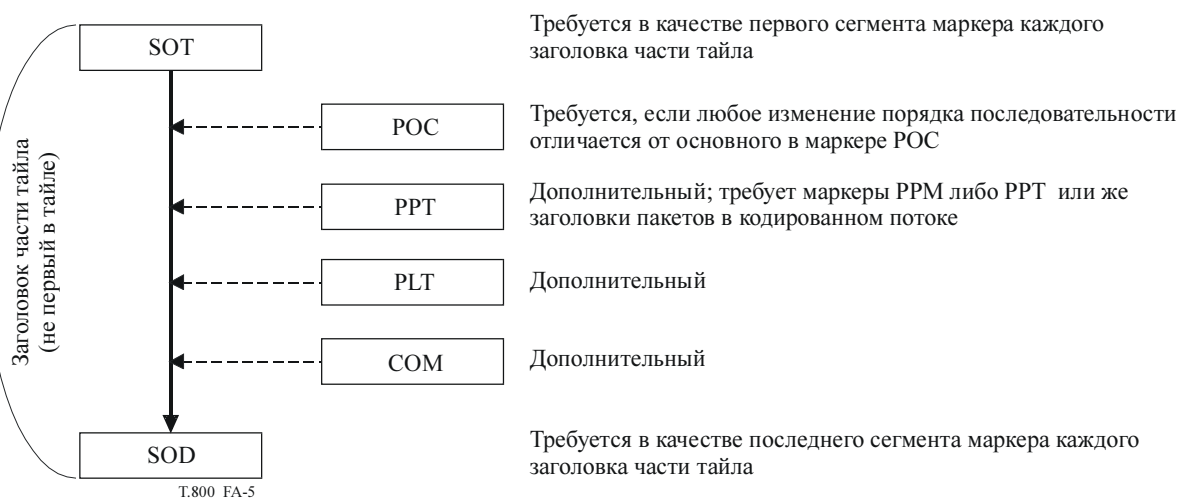


Рисунок А.5 – Структура не первого заголовка части тайла

Сегменты маркеров COD и COC, а также сегменты маркеров QCD и QCC обладают иерархией применения. Это сделано для того, чтобы компоненты тайла имели различные характеристики кодирования и квантования с минимумом сигнализации.

Например, для основного заголовка требуется сегмент маркера COD. Если все компоненты во всех тайлах кодируют одним и тем же способом, это – наилучший вариант того, что требуется. Если же имеется один компонент, который кодируют отлично от прочих (например, компонент яркости изображения, составленный из компонентов яркости и цветности), то маркер COC может показать это в основном заголовке. Если один или более компонентов кодируют различными способами в различных тайлах, то маркеры COD и COC используют аналогичным образом, чтобы показать эту информацию в заголовках частей тайла.

Сегмент маркера POC, который присутствует в основном заголовке, используют во всех тайлах, если только в заголовке части тайла не появляется другой маркер POC.

За исключением маркеров и сегментов маркеров SOC, SOT, SOD, EOC и SIZ сегменты маркера могут появляться в любом порядке соответствующих заголовков.

A.4 Разграничительные маркеры и сегменты маркера

Разграничительный маркер и сегменты маркера должны быть представлены во всех кодированных потоках, соответствующих данной Рекомендации | Международному стандарту. Каждый кодированный поток имеет только один маркер SOC, один маркер EOC и, по крайней мере, одну часть тайла. Каждая часть тайла имеет один маркер SOT и один SOD. Маркеры SOC, SOD и EOC являются разграничительными маркерами, а не сегментами маркера, и не имеют явно выраженную информационную длину или другие параметры.

A.4.1 Запуск кодированного потока (SOC)

Функция: Отмечает начало кодированного потока, описанного в данной Рекомендации | Международном стандарте.

Применение: Основной заголовок. Это первый маркер в кодированном потоке. Должен быть только один маркер SOC на каждый поток.

Длина: Неизменяемая.

SOC: Код маркера.

Таблица A.4 – Значение параметров запуска кодированного потока

Параметр	Размер (биты)	Значение
SOC	16	0xFF4F

A.4.2 Запуск части тайла (SOT)

Функция: Отмечает начало части тайла, индекс этого тайла и индекс этой части тайла. Части тайла заданного тайла должны появляться в кодированном потоке по порядку (см. TP_{sot}). Однако части тайлов от других тайлов могут перемежаться в потоке. Поэтому части данного тайла могут поступать в кодированный поток не непрерывно.

Применение: Заголовок каждой части тайла. В заголовке части тайла должен быть сегмент первого маркера. В кодированном потоке должен быть, по крайней мере, один маркер SOT. На каждую часть тайла должен быть только один маркер SOT.

Длина: Неизменяемая.

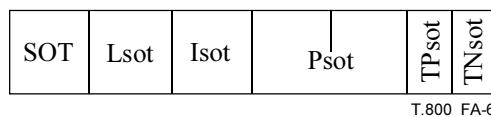


Рисунок A.6 – Синтаксис запуска части тайла

SOT: Код маркера. В таблице A.5 показаны размеры и значения символов и параметров для запуска сегмента маркера части тайла.

Lsot: Длина сегмента маркера в байтах (не включая маркер).

Isot: Индекс тайла. Это число относится к тайлу в порядке запуска раstra, начиная с номера 0.

P_{sot}: Длина в байтах от начала первого байта сегмента маркера SOT части тайла до конца данных этой части тайла. На рисунке A.16 показано расположение. Только последняя часть тайла в кодированном потоке может содержать 0 для P_{sot}. Если параметр P_{sot} равен 0, то предполагается, что эта часть тайла содержит все данные, вплоть до маркера EOC.

TP_{sot}: Индекс части тайла. Существует особый порядок, который требуется для декодирования частей тайла. Данный индекс обозначает порядок, начинающийся с 0. Если в тайле имеется только одна часть тайла, то это значение равно нулю. Части данного тайла должны появляться в кодированном потоке в установленном порядке, хотя и не обязательно последовательно.

TN_{sot}: Число частей данного тайла в кодированном потоке. Допустимы два значения: правильное число частей тайла заданного тайла и нуль. Значение нуля означает, что число частей тайла заданного тайла не определено для этой части тайла.

Таблица А.5 – Значения параметров запуска части тайла

Параметр	Размер (биты)	Значения
SOT	16	0xFF90
Lsot	16	10
Isot	16	от 0 до 65534
Psot	32	от 0 или 14 до $(2^{32} - 1)$
TPsot	8	от 0 до 254
TNsot	8	табл. А.6

Таблица А.6 – Число частей тайла, TNsot, значение параметра

Значение	Число частей тайла
0	В этом заголовке число частей данного тайла в кодированном потоке не определено
от 1 до 255	Число частей данного тайла в кодированном потоке

А.4.3 Запуск данных (SOD)

Функция: Показывает начало потока битов данных для текущей части тайла. Маркер SOD также обозначает конец заголовка части тайла.

Применение: В каждом заголовке части тайла. В заголовке части тайла должен быть последний маркер. Данные потока битов от SOD до следующего маркера SOT или EOC (конец изображения) должны быть кратным 8 битам. Если это необходимо, кодированный поток заполняется битами. В кодированном потоке должен быть, по крайней мере, один маркер SOD, причем один маркер SOD на каждую часть тайла.

Длина: Неизменяемая.

SOD: Код маркера

Таблица А.7 – Значения параметров запуска данных

Параметр	Размер (биты)	Значения
SOD	16	0xFF93

А.4.4 Конец кодированного потока (EOC)

Функция: Указывает на конец кодированного потока.

ПРИМЕЧАНИЕ 1. – Для этого маркера используется тот же самый код, что и для маркера EOI в Рекомендация МСЭ-Т.81 | ИСО/МЭК 10918-1.

Применение: В кодированном потоке должен быть последний маркер. На каждый кодированный поток должен быть один маркер EOC.

ПРИМЕЧАНИЕ 2. – В случае повреждения файла, возможно, что декодер сможет извлечь много полезных данных о сжатом изображении без маркера EOC.

Длина: Неизменяемая.

EOC: Код маркера

Таблица А.8 – Значения параметров конца кодированного потока

Параметр	Размер (биты)	Значения
EOC	16	0xFFD9

А.5 Сегмент маркера неизменяемой информации

Сегмент этого маркера описывает требуемую информацию об изображении. В основном заголовке маркер сегмента SIZ требуется сразу же после маркера SOC.

A.5.1 Размеры изображения и тайла (SIZ)

Функция: Предоставляет информацию о несжатом изображении, такую как ширина и высота опорной сетки, ширина и высота тайлов, число компонентов, глубина компонента в битах и разделение образцов компонента относительно опорной сетки (см. B.2).

Применение: Основной заголовок. В основном заголовке должен быть один и только один SIZ сразу же после сегмента маркера SOC. В кодированном потоке должен быть только один маркер SIZ.

Длина: Переменная, зависящая от числа компонентов.

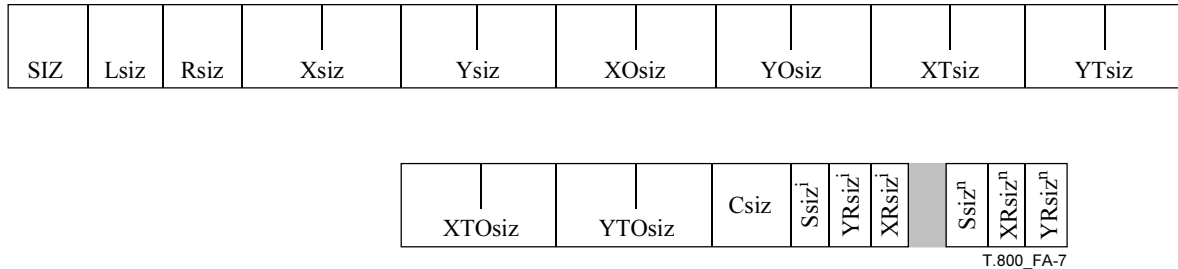


Рисунок A.7 – Синтаксис размеров изображения и тайла

SIZ: Код маркера. В таблице A.9 показаны параметры, размеры и значения для сегмента маркера размеров изображения и тайла.

Lsiz: Длина сегмента маркера в байтах (не включая сам маркер). Значение этого параметра определяется следующим равенством:

$$Lsiz = 38 + 3 \cdot Csiz . \tag{A-1}$$

Rsiz: Обозначает ресурсы, которые требуются декодеру для правильного декодирования кодированного потока.

Xsiz: Ширина опорной сетки.

Ysiz: Высота опорной сетки.

XOsiz: Горизонтальный сдвиг от начала опорной сетки влево от площади изображения.

YOsiz: Вертикальный сдвиг от начала опорной сетки вверх от площади изображения.

XTsiz: Ширина одного опорного тайла по отношению к опорной сетке.

YTtiz: Высота одного опорного тайла по отношению к опорной сетке.

XTOsiz: Горизонтальный сдвиг от начала опорной сетки влево от первого тайла.

YTOsiz: Вертикальный сдвиг от начала опорной сетки вверх от первого тайла.

Csiz: Число компонентов в изображении.

Ssizⁱ: Точность (глубина в битах) и знак образца i-го компонента. Под точностью понимают точность образцов компонентов перед тем, как производится сдвиг уровня постоянного тока (то есть точность исходных образцов компонента перед выполнением любой обработки). Если значения образцов компонента имеют знак, то диапазон значений образца: $2^{(Ssiz+1 \text{ AND } 0x7F)-1} \leq \text{значение образца компонента} \leq 2^{(Ssiz+1 \text{ AND } 0x7F)-1} - 1$. Существует одна реализация этого параметра для каждого компонента. Порядок соответствует индексу компонента, начиная с нуля.

XRsizⁱ: Горизонтальный разнос образца i-го компонента по отношению к опорной сетке. Имеется только одна реализация этого параметра для каждого компонента.

YRsizⁱ: Вертикальный разнос образца i-го компонента по отношению к опорной сетке. Имеется только одна реализация этого параметра для каждого компонента.

Таблица А.9 – Значения параметров размеров изображения и тайла

Параметр	Размер (биты)	Значения
SIZ	16	0xFF51
Lsiz	16	от 41 до 49190
Rsiz	16	Табл. А.10
Xsiz	32	от 1 до $(2^{32} - 1)$
Ysiz	32	от 1 до $(2^{32} - 1)$
XOsiz	32	от 0 до $(2^{32} - 2)$
YOsiz	32	от 0 до $(2^{32} - 2)$
XTsiz	32	от 1 до $(2^{32} - 1)$
YTsiz	32	от 1 до $(2^{32} - 1)$
XTOsiz	32	от 0 до $(2^{32} - 2)$
YTOsiz	32	от 0 до $(2^{32} - 2)$
Csiz	16	От 1 до 16384
Ssiz ⁱ	8	Табл. А.11
XRsiz ⁱ	8	от 1 до 255
YRsiz ⁱ	8	от 1 до 255

Таблица А.10 – Параметр ресурса Rsiz

Значение (биты)		Ресурс
MSB	LSB	
0000	0000 0000 0000	Ресурсы, определенные только в данной Рекомендации Международном стандарте
0000	0000 0000 0001	Кодированный поток, ограниченный так, как это описано в наборе характеристик 0 в табл. А.45
0000	0000 0000 0010	Кодированный поток, ограниченный так, как это описано в наборе характеристик 1 в табл. А.45
		Прочие значения зарезервированы

Таблица А.11 – Параметр компонента Ssiz

Значение (биты)		Точность образца компонента
MSB	LSB	
x000	0000 до x010 0101	Битовая глубина (глубина в битах) образца компонента = значение + 1. Для битовой глубины от 1 до 38 соответственно (учитывая знак бита там, где это следует) ^{a)} , R_I
0xxx	xxxx	Значение образца компонента для величин без знака
1xxx	xxxx	Значение образца компонента для величин со знаком
		Прочие значения зарезервированы
^{a)} Точность образца компонента ограничена числом защитных битов, квантованием, возрастанием коэффициентов в каждом уровне декомпозиции и числом кодирующих проходов, о которых поступил сигнал. Не все комбинации типов кодирования должны допускать кодирование 38-битовых образцов.		

А.6 Сегменты функционального маркера

Эти сегменты маркера описывают функции, которые используют для кодирования собственно тайлов, если они находятся в заголовке части тайла или изображения, а также если они находятся в основном заголовке.

А.6.1 Способ кодирования по умолчанию (COD)

Функция: Описывает способ кодирования, число уровней декомпозиции и размещение по слоям, которые используют по умолчанию для сжатия всех компонентов изображения (если они упомянуты в основном заголовке) или тайла (если они упомянуты в заголовке части тайла). Для отдельного компонента значения параметров могут быть пересмотрены с помощью сегмента СОС либо в основном заголовке, либо в заголовке части тайла.

Применение: Заголовки основной и первой части данного тайла. Дополнительно может быть не менее чем один для каждого тайла. В основном заголовке должен быть один и только один сегмент. Если в тайле содержится множество частей тайла и присутствует этот сегмент маркера, он должен находиться только в первой части тайла ($TP_{\text{pot}} = 0$).

Если маркер COD используют в основном заголовке, значения параметров используют для всех компонентов тайла, для которых нет соответствующего сегмента маркера ни в основном заголовке, ни в заголовке части тайла. Если сегмент маркера COD используют в заголовке части тайла, то он получает преимущество перед основными маркерами COD и COC и используется для всех компонентов в этом тайле без соответствующего сегмента маркера в этой части тайла. Таким образом, существует следующий порядок преимущества:

Часть тайла COC > Часть тайла COD > Основной COC > Основной COD,

где знак "больше", >, означает, что больший имеет преимущество перед меньшим сегментом маркера.

Длина: Переменная, зависит от значения Scod.



Рисунок А.8 – Синтаксис способа кодирования по умолчанию

COD: Код маркера. В таблице А.12 приведены размеры и значения символов, а также параметры способов кодирования и сегмент маркера по умолчанию.

Lcod: Длина сегмента маркера в байтах (не включая сам маркер). Значение этого параметра определяется следующим равенством:

$$Lcod = \begin{cases} 12 & \text{maximum_precincts} \\ 13 + \text{number_decomposition_levels} & \text{user-defined_precincts} \end{cases} \quad (\text{A-2})$$

где максимальная (maximum_precincts) и определенная пользователем (user-defined_precincts) границы указаны в параметре Scod, а число уровней декомпозиции – в параметре SPcod.

Scod: Тип кодирования всех компонентов. В таблице А.13 приведено значение параметра Scod.

SGcod: Параметры типов кодирования, определенных маркером Scod. Эти параметры не зависят от компонентов и обозначены в последовательности сверху вниз в таблице А.14. В поле SGcod параметры типов кодирования приведены в последовательности, показанной на рисунке А.9.

SPcod: Параметры типа кодирования, обозначенные в Scod. Эти параметры относятся ко всем компонентам и приведены в таблице А.15 в последовательности сверху вниз. В поле SPcod параметры типов кодирования приведены в последовательности, показанной на рисунке А.9.

Таблица А.12 – Значение параметров типа кодирования по умолчанию

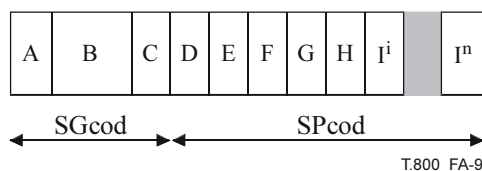
Параметр	Размер (биты)	Значения
COD	16	0xFF52
Lcod	16	от 12 до 45
Scod	8	Табл. А.13
SGcod	32	Табл. А.14
SPcod	переменный	Табл. А.15

Таблица А.13 – Значение параметров типа кодирования для маркера Scod

Значения (биты)		Тип кодирования
MSB	LSB	
xxxx	xxx0	Статистический кодер с границами PPx = 15 и PPy = 15
xxxx	xxx1	Статистический кодер с границами, определенными ниже
xxxx	xx0x	Не используются никакие сегменты маркера SOP
xxxx	xx1x	Могут использоваться сегменты маркера SOP
xxxx	x0xx	Не используется никакой маркер EPH
xxxx	x1xx	Должен использоваться маркер EPHd
		Все прочие значения зарезервированы

Таблица А.14 – Значение параметров типа кодирования для маркера SGcod

Параметры (по порядку)	Размер (биты)	Значения	Содержание значений SGcod
Порядок последовательности	8	Табл. А.16	Порядок последовательности
Число уровней	16	от 1 до 65535	Число уровней
Преобразование множества компонентов	8	Табл. А.17	Применение преобразования множества компонентов



- A Порядок последовательности
- B Число слоев
- C Преобразование множества компонентов
- D Число уровней декомпозиции
- E Ширина блока кодирования
- F Высота блока кодирования
- G Тип блока кодирования
- H Преобразование
- I¹ до Iⁿ Граничный размер

Рисунок А.9 – Диаграмма типов кодирования параметров маркеров SGcod и SPcod

Таблица А.15 – Значения типов кодирования параметров маркеров SPcod и SPcoc

Параметры (по порядку)	Размер (биты)	Значения	Содержание значений SPcod
Число уровней декомпозиции	8	0 – 32	Число уровней декомпозиции, N_L . Нуль означает отсутствие преобразования.
Ширина блока кодирования	8	Табл. А.18	Значение сдвига ширины блока кодирования, xcb
Высота блока кодирования	8	Табл. А.18	Значение сдвига высоты блока кодирования, ycb
Тип блока кодирования	8	Табл. А.19	Тип кодирующих проходов блока кодирования
Преобразование	8	Табл. А.20	Использованное частотное преобразование
Граничные размеры	переменные	Табл. А.21	Если Scod или Scoc = xxxx xxx0, то этот параметр не представлен, в противном случае он указывает на ширину и высоту. Первый параметр (8 битов) соответствует поддиапазону N_L LL. Каждый последующий параметр соответствует каждому последующему уровню разрешения по порядку.

Таблица А.16 – Порядок последовательности параметров SGcod, SPcoc и Ppoc

Значения (биты)		Порядок последовательности
MSB	LSB	
0000	0000	Последовательность: слой, уровень разрешения, компонент, положение
0000	0001	Последовательность разрешения: уровень, слой, компонент, положение
0000	0010	Последовательность разрешения: уровень, положение, компонент, слой
0000	0011	Последовательность: положение, компонент, уровень разрешения, слой
0000	0100	Последовательность: компонент, положение, уровень разрешения, слой
		Все прочие значения зарезервированы

Таблица А.17 – Преобразование множества компонентов параметров SGcod

Значения (биты)		Тип преобразования множества компонентов
MSB	LSB	
0000	0000	Не определено преобразование никакого множества компонентов
0000	0001	Преобразование компонентов, произведенное над компонентами 0, 1 и 2 (см. G.2), для кодирования эффективности. Необратимое преобразование компонентов используют с необратимым фильтром 9-7. Обратимое преобразование компонентов используют с обратимым фильтром 5-3.
		Все прочие значения зарезервированы

Таблица А.18 – Высота и ширина и блоков кодирования параметров SPcod и SPcoc

Значения (биты)		Высота и ширина блока кодирования
MSB	LSB	
xxxx	0000 до xxxx	Значение сдвига блока кодирования по высоте и ширине $xcb = value + 2$ or $ycb = value + 2$. Высота и ширина блока кодирования ограничены степенью два с минимальным размером 2^2 и максимальным размером 2^{10} . Кроме того, блок кодирования ограничен условием: $xcb + ycb \leq 12$.
		Все прочие значения зарезервированы

Таблица А.19 – Тип блока кодирования параметров SPcod и SPcoc

Значения (биты)		Тип блока кодирования
MSB	LSB	
xxxx	xxx0 xxxx	Не предусмотрен избирательный обход арифметического кодирования Избирательный обход арифметического кодирования
xxxx	xx0x xxxx	На границах кодирующего прохода отсутствуют возможности перезагрузки контекста На границах кодирующего прохода имеется возможность перезагрузки контекста
xxxx	x0xx xxxx	Нет завершения при каждом кодирующем проходе Имеется завершение при каждом кодирующим проходе
xxxx	0xxx xxxx	Отсутствует контекст вертикальной причинной связи Имеется контекст вертикальной причинной связи
xxx0	xxxx xxx1	Непредсказуемое завершение Предсказуемое завершение
xx0x	xxxx xx1x	Не используются символы сегментации Используются символы сегментации
		Все прочие значения зарезервированы

Таблица А.20 – Преобразование параметров SPcod и SPcoc

Значение (биты)		Тип преобразования
MSB	LSB	
0000	0000	необратимый фильтр 9-7
0000	0001	обратимый фильтр 5-3
		Все прочие значения зарезервированы

Таблица А.21 – Границы ширины и высоты параметров SPcod и SPcoc

Значение (биты)		Граничные размеры
MSB	LSB	
xxxx	0000 до xxxx 1111	4 бита LSB являются граничными пределами ширины, $PPx = value$. Это значение может быть нулем только при уровне разрешения, соответствующем диапазону N_{iLL} .
0000	xxxx до 1111 xxxx	4 бита MSB являются граничными пределами высоты, $PPy = value$. Это значение может быть нулем только при уровне разрешения, соответствующем диапазону N_{iLL} .

А.6.2 Тип кодирования компонента (COC)

Функция: Описывает тип кодирования и число уровней декомпозиции, использованных для сжатия отдельного компонента.

Применение: Основной заголовок и заголовок первой части тайла. Дополнительно используется как в основном заголовке, так и в заголовке части тайла. Ни в основном заголовке, ни в заголовке части тайла не может быть представлено больше одного для любого заданного компонента. Если в одном тайле имеется много частей тайла и представлен этот сегмент маркера, то он должен находиться только в первой части тайла ($TP_{sot} = 0$).

Если маркер COC используют в основном заголовке, то он получает преимущество перед основным сегментом маркера COD для отдельного компонента. Если COC используют в заголовке части тайла, то он получает преимущество перед основным маркером COD, основным маркером COC, маркером COD тайла для отдельного компонента. Таким образом, порядок преимущества следующий:

Часть тайла COC > Часть тайла COD > Основной COC > Основной COD,

где знак "больше", >, означает, что больший имеет преимущество перед меньшим сегментом маркера.

Длина: Переменная, зависит от Scoc.

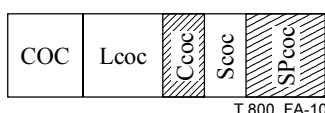


Рисунок А.10 – Синтаксис типа кодирования компонента

COC: Код маркера. В таблице А.22 приведены размеры и значения символов, а также параметры типов кодирования и сегмент маркера.

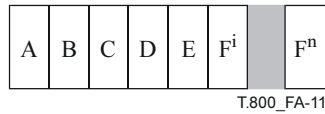
Lcoc: Длина сегмента маркера в байтах (не включая сам маркер). Значение этого параметра определяется следующим равенством:

$$Lcoc = \begin{cases} 9 & \text{maximum_precincts AND } C_{siz} < 257 \\ 10 & \text{maximum_precincts AND } C_{siz} \Rightarrow 257 \\ 10 + \text{number_decomposition_levels} & \text{user-defined_precincts AND } C_{siz} < 257 \\ 11 + \text{number_decomposition_levels} & \text{user-defined_precincts AND } C_{siz} \Rightarrow 257 \end{cases} \quad (A-3)$$

где максимальные (maximum_precincts) и определенные пользователем (user-defined_precincts) границы указаны в параметре Scoc, а число уровней декомпозиции – в параметре SPcoc.

Ccoc: Индекс компонента, к которому относится этот сегмент маркера. Индекс компонентов 0, 1, 2 и т. д.

- Scoc:** Тип кодирования этого компонента. В таблице A.23 показаны значения для каждого параметра Scoc.
- SPcoc:** Параметр типа кодирования, определенный в Scoc. Эти параметры указаны в таблице A.15 по порядку сверху вниз. В поле SPcoc параметры типа кодирования присутствуют в последовательности, показанной на рисунке A.11.



- A Число уровней декомпозиции
- B Ширина блока кодирования
- C Высота блока кодирования
- D Тип блока кодирования
- E Преобразование
- Fⁱ до Fⁿ Граничный размер

Рисунок A.11 – Диаграмма типа кодирования параметров SPcoc

Таблица A.22 – Значения параметров кодирования компонентов

Параметр	Размер (биты)	Значения
COC	16	0xFF53
Lcoc	16	от 9 до 43
Ccoc	8 16	от 0 до 255; если Csiz < 257 от 0 до 16383; Csiz ≥ 257
Scoc	8	Табл. A.23
SPcoc ⁱ	Переменный	Табл. A.15

Таблица A.23 – Тип кодирования значений параметров для параметра Scoc

Значение (биты)		Тип кодирования
MSB	LSB	
0000	0000	Статистический кодер с максимальными граничными значениями PPx = PPy = 15
0000	0001	Статистический кодер со значениями границ, определенными ниже
		Все прочие значения зарезервированы

A.6.3 Область интересов (RGN)

Функция: Сигналы представления области ROI в кодированном потоке.

Применение: В основном заголовке или заголовке первой части тайла. При использовании в основном заголовке маркер относится к значению масштабирования ROI для одного компонента всего изображения, действующего для всех тайлов, за исключением тех, которые связаны с сегментом маркера RGN.

При использовании в заголовке части тайла значение масштабирования действительно только для одного компонента в этом тайле. Может существовать, по крайней мере, один сегмент маркера RGN для каждого компонента либо в основном заголовке, либо в заголовке первой части тайла. Сегмент маркера RGN для отдельного компонента, отмеченного в заголовке части тайла, получает преимущество перед любым маркером для этого компонента в основном заголовке для этого тайла. Если в тайле существует множество частей, то этот сегмент маркера должен находиться только в первом заголовке части тайла.

Длина: Переменная.

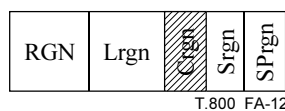


Рисунок A.12 – Синтаксис области интересов

- RGN:** Код маркера. В таблице A.24 приведены размеры и значения символов, а также параметры для сегмента маркера области интересов.
- Lrgn:** Длина сегмента маркера в байтах (не включая сам маркер).
- Crgn:** Индекс компонента, к которому относится сегмент маркера. Индекс компонента 0, 1, 2 и т. д.
- Srgn:** Тип маркера ROI для текущей области ROI. В таблице A.25 показаны значения параметра Srgn.
- SPrgn:** Параметр для типа ROI, описанного в Srgn.

Таблица A.24 – Значения параметров области интересов

Параметр	Размер (биты)	Значения
RGN	16	0xFF5E
Lrgn	16	5 – 6
Crgn	8 16	0–255; если Csiz < 257 0–16383; Csiz ≥ 257
Srgn	8	Табл. A.25
SPrgn	8	Табл. A.26

Таблица A.25 – Значения параметров области интересов для параметра Srgn

Значения	Тип ROI (Srgn)
0	Неявная область ROI (максимальный сдвиг)
	Все прочие значения зарезервированы

Таблица A.26 – Значения области интересов, полученные из параметра SPrgn (Srgn = 0)

Параметры (по порядку)	Размер (биты)	Значения	Содержание значения SPrgn
Неявный сдвиг ROI	8	0 – 255	Бинарный сдвиг коэффициентов области ROI по сравнению с исходными коэффициентами

A.6.4 Квантование по умолчанию (QCD)

Функция: Описывает процесс квантования по умолчанию, который используют для сжатия всех компонентов, не определенных в сегменте маркера QCC. С помощью сегмента маркера QCC значения параметров для отдельного компонента могут получать преимущество в основном заголовке или заголовке части тайла.

Применение: В основном или в первом заголовке части данного тайла. Маркер должен присутствовать в основном заголовке один и только один раз. Может быть, по крайней мере, один для всех заголовков частей тайла. Если в тайле существует множество частей и присутствует этот сегмент маркера, то он должен находиться только в первой части тайла (TPsot = 0).

При использовании в заголовке части тайла этот маркер имеет преимущество над основными маркерами QCD и QCC для некоторых компонентов. Таким образом, порядок преимущества следующий:

Часть тайла QCC > Часть тайла QCD > Основной QCC > Основной QCD,

где знак "больше", >, означает, что больший имеет преимущество перед меньшим сегментом маркера.

Длина: Переменная, зависит от числа элементов квантования.

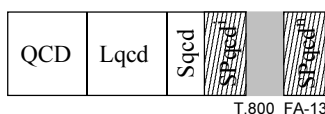


Рисунок A.13 – Синтаксис квантования по умолчанию

QCD: Код маркера. В таблице A.27 приведены размеры и значения символов, а также параметры для сегмента маркера квантования по умолчанию.

Lqcd: Длина сегмента маркера в байтах (не включая сам маркер). Значение этого параметра определяется следующим равенством:

$$Lqcd = \begin{cases} 4 + 3 \cdot \text{number_decomposition_levels} & \text{no_quantization} \\ 5 & \text{scalar_quantization_derived} \\ 5 + 6 \cdot \text{number_decomposition_levels} & \text{scalar_quantization_expounded} \end{cases}, \quad (\text{A-4})$$

где число `number_decomposition_levels` определяется в сегментах маркеров COD и COC, а значения `no_quantization`, `scalar_quantization_derived` и `scalar_quantization_expounded` приведены в параметре `Sqcd`.

ПРИМЕЧАНИЕ. – Длину `Lqcd` можно использовать, чтобы определить количество ступеней квантования в сегменте маркера. Однако при этом нет необходимости соотносить это с числом действительных поддиапазонов, поскольку поддиапазоны могут быть сокращены без необходимости коррекции этого сегмента маркера.

Sqcd: Тип квантования для всех компонентов.

SPqcdⁱ: Величина шага квантования для *i*-го поддиапазона в определенном порядке (см. F.3.1). Число параметров соответствует числу поддиапазонов в компоненте тайла с наибольшим числом уровней декомпозиции.

Таблица A.27 – Значения параметров квантования по умолчанию

Параметр	Размер (биты)	Значения
QCD	16	0xFF5C
Lqcd	16	4–197
Sqcd	8	Табл. A.28
SPqcd ⁱ	переменный	Табл. A.28

Таблица A.28 – Значения квантования по умолчанию для параметров Sqcd и Sqcc

Значение (биты) MSB LSB	Тип квантования	Размер SPqcd или SPqcc (биты)	Применение SPqcd или SPqcc
xxx0 0000	Нет квантования	8	Табл. A.29
xxx0 0001	Полученный скаляр (значения сигнализации только для поддиапазона N_{iLL}). Используют равенство (E-5)	16	Табл. A.30
xxx0 0010	Интерпретированный скаляр (значения сигнализации для каждого поддиапазона). Имеется столько размеров шагов сигнализации, сколько и поддиапазонов	16	Табл. A.30
000x xxxx до 111x xxxx	Число защитных битов: от 0 до 7		
	Все прочие значения зарезервированы		

Таблица A.29 – Значения размеров обратимых шагов для параметров SPqcd и SPqcc (только при обратимом преобразовании)

Значение (биты) MSB LSB	Значения размеров обратимых шагов
0000 0xxx до 1111 1xxx	Показатель ϵ_b обратимости динамического диапазона, о котором для каждого поддиапазона поступил сигнал (см. равенство (E-5))
	Все прочие значения зарезервированы

Таблица А.30 – Значения квантования для параметров SPqcd и SPqcc (только для необратимого преобразования)

Значение (биты)		Значения шагов квантования
MSB	LSB	
xxxx	x000 0000 0000 до xxxx x111 1111 1111	Мантисса μ_b значения шага квантования (см. равенство (E-5))
0000	0xxx xxxx xxxx до 1111 1xxx xxxx xxxx	Показатель ϵ_b значения шага квантования (см. равенство (E-5))

А.6.5 Квантование компонента (QCC)

Функция: Описывает квантование, которое используют для сжатия отдельного компонента.

Применение: В основном и первом заголовке данного тайла. Дополнительно можно использовать в обоих этих заголовках. Для каждого компонента можно использовать не более одного маркера либо в основном, либо в заголовке части тайла. Если в тайле существует множество частей и присутствует этот сегмент маркера, то он должен находиться только в первой части тайла (TPsot = 0).

Дополнительно может находиться в обоих заголовках: основном и части тайла. При использовании в основном заголовке этот маркер имеет преимущество над основным сегментом маркера QCD для некоторых компонентов. При использовании в заголовке части тайла этот заголовок для некоторых компонентов имеет преимущество над основными маркерами QCD и QCC, а также маркером тайла QCD. Таким образом, порядок преимущества следующий:

Часть тайла QCC > Часть тайла QCD > Основной QCC > Основной QCD,

где знак "больше", >, означает, что больший имеет преимущество перед меньшим сегментом маркера.

Длина: Переменная, зависит от числа элементов квантования.

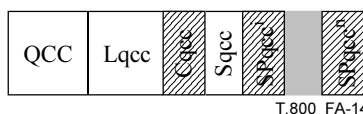


Рис. А.14 – Синтаксис квантования компонента

QCC: Код маркера. В таблице А.31 приведены размеры и значения символов, а также параметры для сегмента маркера квантования компонента.

Lqcc: Длина сегмента маркера в байтах (не включая сам маркер). Значение этого параметра определяется следующим равенством:

$$Lqcc = \begin{cases} 5 + 3 \cdot \text{number_decomposition_levels} & \text{no_quantization AND Csiz} < 257 \\ 6 & \text{scalar_quantization_derived AND Csiz} < 257 \\ 6 + 6 \cdot \text{number_decomposition_levels} & \text{scalar_quantization_expounded AND Csiz} < 257 \\ 6 + 3 \cdot \text{number_decomposition_levels} & \text{no_quantization AND Csiz} \Rightarrow 257 \\ 7 & \text{scalar_quantization_derived AND Csiz} \Rightarrow 257 \\ 7 + 6 \cdot \text{number_decomposition_levels} & \text{scalar_quantization_expounded AND Csiz} \Rightarrow 257 \end{cases}, \quad (\text{A-5})$$

где число number_decomposition_levels определено в сегментах маркеров COD и COC, а значения no_quantization, scalar_quantization_derived или scalar_quantization_expounded обозначены в параметре Sqcc.

ПРИМЕЧАНИЕ. – Длину Lqcc можно использовать, чтобы определить количество ступеней квантования в сегменте маркера. Однако при этом нет необходимости соотносить это с числом действительных поддиапазонов, поскольку поддиапазоны могут быть сокращены без необходимости коррекции этого сегмента маркера.

Cqcc: Индекс компонента, к которому относится этот сегмент маркера. Компоненты индексируются как 0, 1, 2 и т. д. (либо 8, либо 16 битов в зависимости от значения Csiz).

Sqcc: Тип квантования для этого компонента.

SPqccⁱ: Значение квантования для каждого диапазона в определенном порядке (см. F.3.1). Число параметров соответствует числу поддиапазонов в компоненте тайла с наибольшим числом уровней декомпозиции.

Таблица А.31 – Значения параметров квантования компонентов

Параметр	Размер (биты)	Значения
QCC	16	0xFF5D
Lqcc	16	5 – 199
Cqcc	8 16	0–255; если Csiz < 257 0–16383; Csiz ≥ 257
Sqcc	8	Табл. А.28
SPqcc ⁱ	Переменный	Табл. А.28

А.6.6 Изменение порядка последовательности (РОС)

Функция: Описывает границы и порядок последовательности для любых последовательностей, кроме тех, которые оговорены в сегментах маркеров COD в кодированном потоке.

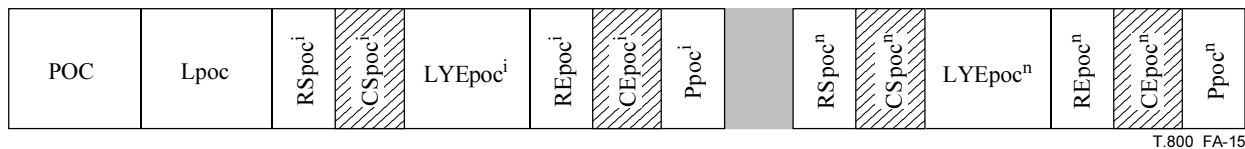
Применение: В основном заголовке и заголовке части тайла. В любом заголовке должен быть, по крайней мере, один сегмент маркера РОС. Однако несколько последовательностей могут описываться одним сегментом маркера РОС. Если сегмент маркера РОС используют в основном заголовке, он получает преимущество в порядке последовательности в основном заголовке и сегментах маркера тайла COD. Если для описания последовательности отдельного тайла используется РОС, сегмент маркера РОС должен появиться в заголовке первой части тайла. Таким образом, порядок последовательности заданного тайла определяется присутствием маркера РОС или значений COD со следующим порядком преимущества:

Часть тайла РОС > Основной РОС > Часть тайла COD > Основной COD,

где знак "больше", >, означает, что больший имеет преимущество перед меньшим сегментом маркера.

В случаях использования сегмента маркера РОС, последовательность каждого пакета в кодированном потоке (или для этого тайла кодированного потока) должна быть описана в одном или более сегментах маркера РОС. Каждый порядок последовательности описывается только в одном сегменте маркера РОС и должен быть описан в любом заголовке части тайла, прежде чем появится какой-нибудь пакет этой последовательности.

Длина: Переменная, зависит от числа различных последовательностей.



T.800_FA-15

Рисунок А.15 – Синтаксис тайла для изменения порядка последовательности

РОС: Код маркера. В таблице А.32 приведены размеры и значения символов, а также параметры сегмента маркера для изменения порядка последовательности.

Lpoc: Длина сегмента маркера в байтах (не включая сам маркер). Значение этого параметра определяется следующим равенством:

$$Lpoc = \begin{cases} 2 + 7 \cdot \text{number_progression_order_change} & \text{Csiz} < 257 \\ 2 + 9 \cdot \text{number_progression_order_change} & \text{Csiz} \Rightarrow 257 \end{cases} \quad (\text{A-6})$$

где число number_progression_order_changes определено декодером.

RSposⁱ: Индекс уровня разрешения (включающий) для запуска последовательности. В этом тайле или части тайла одно значение для каждого изменения последовательности. Число изменений последовательностей может быть получено из длины сегмента маркера.

CSposⁱ: Индекс компонента (включающий) для запуска последовательности. Компоненты индексируются как 0, 1, 2 и т. д. (либо 8, либо 16 битов в зависимости от значения Csiz). В этом тайле или части тайла одно значение для каждого изменения последовательности. Число изменений последовательностей может быть получено из длины сегмента маркера.

- LYE_{pos}ⁱ:** Индекс уровня (исключающий) для конца последовательности. Индекс уровня для каждой последовательности всегда начинается с нуля. Пакеты, которые уже были включены в кодирующий поток, более не включаются. В этом тайле или части тайла одно значение для каждого изменения последовательности. Число изменений последовательностей может быть получено из длины сегмента маркера.
- RE_{pos}ⁱ:** Индекс уровня разрешения (исключающий) для конца последовательности. В этом тайле или части тайла одно значение для каждого изменения последовательности. Число изменений последовательностей может быть получено из длины сегмента маркера.
- CE_{pos}ⁱ:** Индекс компонента (исключающий) для конца последовательности. Компоненты индексируются как 0, 1, 2 и т. д. (либо 8, либо 16 битов в зависимости от значения Csiz) В этом тайле или части тайла одно значение для каждого изменения последовательности. Число изменений последовательностей может быть получено из длины сегмента маркера.
- P_{pos}ⁱ:** Порядок последовательности. В этом тайле или части тайла одно значение для каждого изменения последовательности. Число изменений последовательностей может быть получено из длины сегмента маркера.

Таблица А.32 – Изменение порядка последовательности, значения параметров тайла

Параметр	Размер (биты)	Значения
POC	16	0xFF5F
L _{pos}	16	от 9 до 65535
R _{Spoc} ⁱ	8	от 0 до 32
C _{Spoc} ⁱ	8 16	от 0 до 255; если Csiz < 257 от 0 до 16383; Csiz ≥ 257
LYE _{pos} ⁱ	16	от 1 до 65535
RE _{pos} ⁱ	8	от (R _{Spoc} ⁱ + 1) до 33
CE _{pos} ⁱ	8 16	(C _{Spoc} ⁱ + 1) до 255, 0; если Csiz < 257 (C _{Spoc} ⁱ + 1) до 16384, 0; Csiz ≥ 257 (0 интерпретируется как 256)
P _{pos} ⁱ	8	Табл. А.16

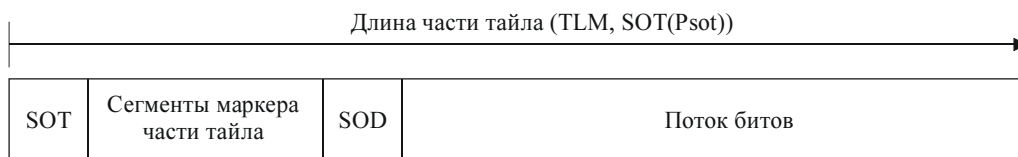
А.7 Сегменты маркера указателя

Сегменты маркера указателя либо поддерживают длину пакета, либо служат указателями в кодированном потоке. Сегмент маркера TLM описывает длину частей тайла. Он имеет такую же информационную длину, как сегмент маркера SOT. Сегменты маркеров PLM и PLT описывают длину пакетов.

ПРИМЕЧАНИЕ. – Присутствие в основном заголовке сегментов маркера указателя обеспечивает непосредственный доступ к потоку битов данных. Присутствие в заголовках частей тайла информации указателя снимает с кодировщика задачу поиска хранимой информации.

Параметры TLM (P_{tlm}) или SOT (P_{sot}) указывают на начало сегмента маркера SOT текущей части тайла и конец потока битов данных этой части тайла. Величину длины всегда выражают в байтах, поэтому части тайла требуется умножить на 8 битов. На рисунке А.16 показана длина части тайла.

Сегменты маркеров PLM или PLT являются дополнительными. Сегменты маркера PLM используют в основном заголовке, а сегменты маркера PLT – в заголовках частей тайла. Сегменты маркеров PLM и PLT описывают длину каждого пакета в кодированном потоке.



T.800_FA-16

Рисунок А.16 – Длины частей тайла

А.7.1 Длины частей тайла (TLM)

Функция: Описывает длину каждой части тайла в кодированном потоке. Длина каждой части тайла измеряется от первого байта сегмента маркера SOT до конца потока битов этой части тайла. Длина каждой отдельной части тайла в сегменте маркера TLM точно такая же, как длина в соответствующем параметре P_{sot} сегмента маркера SOT.

Применение: Основной заголовок. Дополнительно используют только в основном заголовке. В основном заголовке может быть множество сегментов маркера TLM.

Длина: Переменная, зависит от числа частей тайла в кодированном потоке.

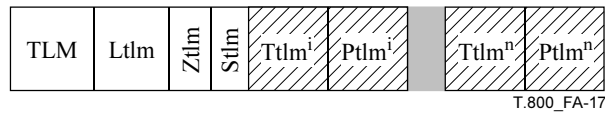


Рисунок А.17 – Синтаксис длины части тайла

TLM: Код маркера. В таблице А.33 приведены размеры и значения символов, а также параметры для длины сегмента маркера части тайла.

Ltlm: Длина сегмента маркера в байтах (не включая сам маркер). Значение этого параметра определяется следующим равенством:

$$Ltlm = \begin{cases} 4 + 2 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 0 \text{ AND } SP = 0 \\ 4 + 3 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 1 \text{ AND } SP = 0 \\ 4 + 4 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 2 \text{ AND } SP = 0 \\ 4 + 4 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 0 \text{ AND } SP = 1 \\ 4 + 5 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 1 \text{ AND } SP = 1 \\ 4 + 6 \cdot \text{number_of_tile_parts_in_marker_segment} & ST = 2 \text{ AND } SP = 1 \end{cases} \quad (A-7)$$

где число *number_of_tile-parts_in_marker_segment* – это число длин частей тайла, которые описаны в сегменте маркера. *ST* и *SP* – сигналы от параметра *Stlm*.

Ztlm: Индекс этого маркера относительно всех других сегментов маркера TLM представлен в текущем заголовке. Последовательность пар (*Ttlmⁱ*, *Ptlmⁱ*) этого сегмента маркера объединяется в порядке возрастания *Ztlm* с последовательностями пар от других сегментов маркера. *J*-ый вход в списке разрешений содержит индекс тайла и длину пары частей тайла для включения в кодированный поток *j*-ой части тайла.

Stlm: Величины параметров *Ttlm* и *Ptlm*.

Ttlmⁱ: Индекс тайла *i*-ой части тайла. Для каждой части тайла это значение либо не существует, либо оно единственное. Число частей тайла в каждом тайле может быть получено из этого сегмента маркера (или списка объединений всех таких маркеров) или из ненулевого параметра *TN_{not}*, если он представлен.

Ptlmⁱ: Длина в байтах от начала маркера SOT *i*-ой части тайла до конца потока битов данных для этой части тайла. Одно значение для каждой части тайла.

Таблица А.33 – Значения параметров длины части тайла

Параметр	Размер (биты)	Значения
TLM	16	0xFF55
Ltlm	16	6–65535
Ztlm	8	0–255
Stlm	8	Табл. А.34
Ttlm ⁱ	0, если ST = 0 8, если ST = 1 16, если ST = 2	тайлы по порядку 0–254 0–65534
Ptlm ⁱ	16, если SP = 0 32, если SP = 1	14–65535 14–(2 ³² –1)

Таблица А.34 – Размеры параметров *Stlm*

Значение (биты)		Размер параметра
MSB	LSB	
xx00	xxxx	ST = 0; 0 битов для параметра <i>Ttlm</i> . В индекс без пропусков и повторений включены все тайлы, а также по одной части тайла от каждого тайла
xx01	xxxx	ST = 1; параметр <i>Ttlm</i> 8 битов
xx10	xxxx	ST = 2; параметр <i>Ttlm</i> 16 битов
x0xx	xxxx	SP = 0; параметр <i>Ptlm</i> 16 битов
x1xx	xxxx	SP = 1; параметр <i>Ptlm</i> 32 бита
		Все прочие значения зарезервированы

А.7.2 Длина пакета, основной заголовок (PLM)

Функция: Перечень длин пакетов в частях тайла по порядку для каждой части.

Применение: Основной заголовок. Может существовать множество сегментов маркеров. Сегменты маркеров PLM и PLT являются дополнительными и могут использоваться вместе или по отдельности.

Длина: Переменная, зависит от числа частей тайла в изображении и числа пакетов в каждой части тайла.

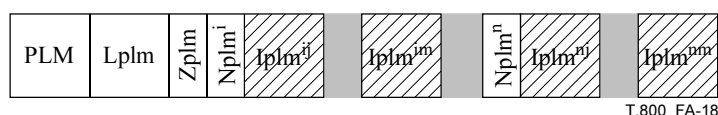


Рисунок А.18 – Длины пакетов, синтаксис основного заголовка

PLM: Код маркера. В таблице А.35 приведены размеры и значения символов, а также параметры для длины пакета, основного заголовка и сегмента маркера.

Lplm: Длина сегмента маркера в байтах (не включая сам маркер).

Zplm: Индекс этого сегмента маркера относительно всех других сегментов маркера PLM представлен в текущем заголовке. Последовательность параметров ($Nplm^i$, $Iplm^i$) этого сегмента маркера объединяется в порядке возрастания $Zplm$ с последовательностями параметров других сегментов маркера. К-ый вход в списке разрешений содержит число байтов и число пар заголовков пакетов для k-ой части тайла, присутствующих в кодированном потоке.

Каждый сегмент маркера в этой последовательности должен оканчиваться полной длиной заголовка пакета. Однако серия из параметров $Iplm$, описанных с помощью $Nplm$, не должна заканчиваться в данном сегменте маркера. Поэтому вполне возможно, что в следующем сегменте маркера PLM параметр $Nplm$ не будет следовать за $Zplm$, а продолжит последовательность $Iplm$ из последнего сегмента маркера PLM.

$Nplm^i$: Число байтов информации $Iplm$ для i-ой части тайла в порядке следования кодированного потока. На каждую часть тайла приходится одно значение. Если кодированный поток содержит одну или более частей тайла, превышающих ограничения маркеров PLM, эти маркеры не должны использоваться.

ПРИМЕЧАНИЕ. – Данное значение выражается 8-битовым числом, которое ограничивает число байтов $Iplm$ величиной 255, а число пакетов в части тайла – величиной 255 или менее. Это связано не с ограничением числа пакетов, которые могут находиться в части тайла, а с ограничением возможности сегмента маркера описать пакеты в части тайла.

$Iplm^{ij}$: Длина j-ого пакета в i-ой части тайла. Если заголовки пакетов запоминаются с этим пакетом, то длина включает и длину заголовка пакета. Если заголовки пакетов запоминаются в PPM или PPT, то эта длина не включает длину заголовка пакета. На каждую часть тайла приходится один диапазон значений и по одному значению для каждого пакета в тайле.

Таблица А.35 – Значение параметров длины пакетов и основного заголовка

Параметр	Размер (биты)	Значения
PLM	16	0xFF57
Lplm	16	4–65535
Zplm	8	0–255
Nplm ⁱ	8	0–255
Iplm ⁱⁱ	переменный	Табл. А.36

Таблица А.36 – Список длины пакетов Iplm, Iplt

Параметры (по порядку)	Размер (биты)	Значения	Содержание значений Iplm или Iplt
Длина пакета	При необходимости 8 битов повторяются	0xxx xxxx 1xxx xxxx x000 0000 до x111 1111	Последние 7 битов длины пакета, предельное число ^{a)} Продолжить считывание ^{b)} 7 битов длины пакета
<p>a) Это последние 7 битов, которые составляют длину пакета.</p> <p>b) Это не последние 7 битов, которые составляют длину пакета. Эти 7 битов составляют часть тех битов, которые составляют длину пакета. Длину пакета можно разбить на 7-битовые сегменты, которые передаются в порядке от наиболее значимого сегмента к наименее значимому сегменту. Кроме того, биты наиболее значимого сегмента составляют соответствующую границу байтов. Например, пакету длиной 128 соответствует сигнал 1000 0001 0000 0000, а длиной 512 – сигнал 1000 0100 0000 0000.</p>			

А.7.3 Длина пакета, заголовок части тайла (PLT)

Функция: Перечень длин пакетов в части тайла.

Применение: Заголовки частей тайла. В одном тайле может быть множество сегментов маркера PLT. Сегменты маркеров PLM и PLT являются дополнительными и могут использоваться вместе или по отдельности. Должны присутствовать в каждом заголовке части тайла перед пакетами, длина которых указана в этих сегментах.

Длина: Переменная, зависит от числа пакетов в каждой части тайла.

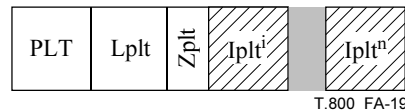


Рисунок А.19 – Длина пакета, синтаксис заголовка части тайла

- PLT:** Код маркера. В таблице А.37 приведены размеры и значения символов, а также параметры для длины пакета и сегмента маркера заголовка части тайла.
- Lplt:** Длина сегмента маркера в байтах (не включая сам маркер).
- Zplt:** Индекс этого сегмента маркера относительно всех других сегментов маркера PLT представлен в текущем заголовке. Последовательность параметров (Ipltⁱ) этого сегмента маркера объединяется в порядке возрастания Zplt с последовательностями параметров других сегментов маркера. Каждый сегмент маркера в этой последовательности должен оканчиваться полной длиной заголовка пакета.
- Ipltⁱ:** Длина i-ого пакета. Если заголовки пакетов запоминаются с этим пакетом, то эта длина включает и длину заголовка пакета. Если заголовки пакетов запоминаются в маркерах PPM или PPT, то в эту длину не включена длина заголовка пакета.

Таблица А.37 – Длина пакета, значения параметров заголовков частей тайла

Параметр	Размер (биты)	Значения
PLT	16	0xFF58
Lplt	16	4–65535
Zplt	8	0–255
Iplt ⁱ	переменный	Табл. А.36

А.7.4 Заголовки упакованных пакетов, основной заголовок (PPM)

Функция: Набор заголовков пакетов всех тайлов.

ПРИМЕЧАНИЕ. – Это полезно, поскольку для многих считываний не потребуется декодирование заголовков.

Применение: Основной заголовок. Может быть использован для всех частей тайла, если только в заголовке части тайла не использован сегмент маркера PPT.

В кодированном потоке заголовки пакетов должны находиться только в одном из трех мест.

Если присутствует сегмент маркера PPM, все заголовки пакетов должны находиться в основном заголовке. В этом случае запрещено использование сегмента маркера PPT и пакетов, которые находятся в потоке битов из частей тайлов.

Если отсутствует сегмент маркера PPM, заголовки пакетов могут размещаться либо в сегментах маркера PPT, либо в кодированном потоке, как описано в В.10. Заголовки пакетов не должны находиться в сегменте маркера PPT и в кодированном потоке того же самого тайла.

Если в сегменте маркера имеются заголовки пакетов, они должны появляться в заголовке части тайла до появления соответствующих данных пакета (то есть в том же самом заголовке части тайла или в заголовке с меньшим значением TPso_t). В заголовке части тайла может находиться много сегментов маркера PPT.

Длина: Переменная, зависит от числа пакетов в каждой части тайла и от размеров заголовков пакетов.

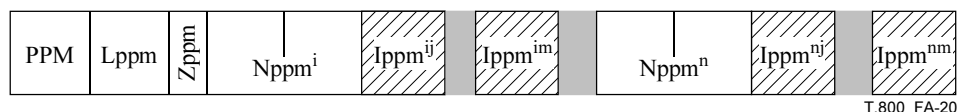


Рисунок А.20 – Заголовки упакованных пакетов, синтаксис основного заголовка

PPM: Код маркера. В таблице А.38 приведены размеры и значения символов, а также параметры для заголовков упакованных пакетов и сегмента маркера основного заголовка.

Lppm: Длина сегмента маркера в байтах (не включая сам маркер).

Zppm: Индекс этого сегмента маркера относительно всех других сегментов маркера PPM представлен в основном заголовке. Последовательность параметров ($Nppm^i$, $Ippm^i$) этого сегмента маркера объединяется в порядке возрастания $Zppm$ с последовательностями параметров других сегментов маркера. К-ый вход в списке разрешений содержит число байтов и пар заголовков пакетов для k-ой части тайла, появляющейся в кодированном потоке.

Каждый сегмент маркера в этой серии должен оканчиваться полной длиной заголовка пакета. Однако последовательность из параметров $Ippm$, описанных с помощью $Nppm$, не должна заканчиваться в данном сегменте маркера. Поэтому вполне возможно, что в следующем сегменте маркера PPM параметр $Nppm$ не будет следовать за $Zppm$, а продолжит последовательность $Ippm$ из последнего сегмента маркера PPM.

$Nppm^i$: Число байтов информации $Ippm$ для i-ой части тайла в порядке следования кодированного потока. На каждую часть тайла приходится одно значение (но не тайл).

$Ippm^{ij}$: Заголовок пакета для каждого пакета по порядку в части этого тайла. Содержанием заголовка является в точности заголовок пакета, который должен размещаться в кодированном потоке, согласно В.10.

Таблица А.38 – Заголовки упакованных пакетов, значения параметров основного заголовка

Параметр	Размер (биты)	Значения
PPM	16	0xFF60
Lppm	16	7–65535
Zppm	8	0–255
Nppm ⁱ	32	0–(2 ³² –1)
Ippm ⁱⁱ	переменный	заголовки пакетов

А.7.5 Заголовки упакованных пакетов, заголовок части тайла (PPT)

Функция: Набор заголовков пакетов от одного тайла или части тайла.

Применение: Заголовки части тайла. Должен появляться в любой части тайла до появления пакетов, заголовки которых описываются здесь.

В кодированном потоке заголовки пакетов должны находиться только в одном из трех мест.

Если присутствует сегмент маркера PPM, все заголовки пакетов должны располагаться в основном заголовке. В этом случае запрещено использование сегмента маркера PPM и пакетов, которые находятся в потоке битов частей тайлов.

Если отсутствует сегмент маркера PPM, заголовки пакетов могут размещаться либо в сегментах маркера PPT, либо в кодированном потоке, как описано в В.10. Заголовки пакетов не должны находиться в сегменте маркера PPT и в кодированном потоке того же самого тайла.

Если в сегменте маркера имеются заголовки пакетов, они должны появляться в заголовке части тайла до появления соответствующих данных пакета (то есть в том же самом заголовке части тайла или с меньшим значением TPstot). В заголовке части тайла может находиться много сегментов маркера PPT.

Длина: Переменная, зависит от числа пакетов в каждой части тайла и от размеров заголовков пакетов.

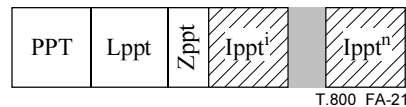


Рисунок А.21 – Заголовки упакованных пакетов, синтаксис заголовка части тайла

- PPT:** Код маркера. В таблице А.39 приведены размеры и значения символов, а также параметры заголовков упакованных пакетов и сегментов маркера в заголовке части тайла.
- Lppt:** Длина сегмента маркера в байтах (не включая сам маркер).
- Zppt:** Индекс этого сегмента маркера относительно всех других сегментов маркера PPT, представленного в текущем заголовке. Последовательность параметров (Ipptⁱ) этого сегмента маркера объединяется в порядке возрастания Zppt с последовательностями параметров других сегментов маркера. Каждый сегмент маркера в этой серии должен оканчиваться полной длиной заголовка пакета.
- Ipptⁱ:** Заголовок пакета для каждого пакета по порядку в части заданного тайла. Индекс компонента, слоя и уровня разрешения определяются типом последовательности или сегментами маркера POS. Содержанием заголовка является в точности заголовок пакета, который должен размещаться в кодированном потоке, согласно В.10.

Таблица А.39 – Значения параметров заголовка пакета и заголовков частей тайла

Параметр	Размер (биты)	Значения
PPT	16	0xFF61
Lppt	16	4–65535
Zppt	8	0–255
Ippt ⁱ	переменный	заголовки пакетов

А.8 Маркер в потоке битов и сегменты маркера

Эти маркеры и сегменты маркера используют для защиты от ошибок. Они должны находиться в потоке битов. (Маркер EPH может быть также использован в сегментах маркеров PPM и PPT).

А.8.1 Запуск пакета (SOP)

Функция: Отмечает начало пакета в кодированном потоке.

Применение: Дополнительное. Может быть использован в потоке битов перед каждым пакетом. Не может быть использован, если в соответствующем сегменте маркера COD нет указания на возможность его применения (см. А.6.1). Если используются сегменты маркеров PPM или PPT, то сегмент маркера SOP может появиться в потоке битов непосредственно перед данными пакета.

Если применение сегментов маркера SOP разрешено (сигнализацией в сегменте маркера COD, см. А.6.1), каждый пакет в любой заданной части тайла может быть присоединен или не присоединен к сегменту маркера SOP. Однако, независимо от того, используется или не используется сегмент маркера SOP, появление каждого пакета засчитывается в N_{sop}. Если заголовки пакетов помещены в сегменты маркера PM или PPT (см. А.7.4 и А.7.5), то сегменты маркера SOP могут появляться непосредственно перед телом пакета в данных сжатого изображения части тайла.

Длина: Неизменяемая.

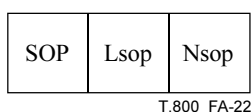


Рисунок А.22 – Синтаксис запуска пакета

SOP: Код маркера. В таблице А.40 приведены размеры и значения символов, а также параметры для запуска сегмента маркера пакета.

L_{sop}: Длина сегмента маркера в байтах (не включая сам маркер).

N_{sop}: Номер последовательности пакета. Первому пакету в кодированном тайле присваивают нулевое значение. Для каждого последующего пакета в этом тайле номер увеличивается на единицу. Когда достигается максимальное число, номер возвращается на нуль.

Таблица А.40 – Значения параметров запуска пакетов

Параметр	Размер (биты)	Значения
SOP	16	0xFF91
L _{sop}	16	4
N _{sop}	16	0–65535

А.8.2 Конец заголовка пакета (EPH)

Функция: Обозначает конец заголовка заданного пакета. Отделяет заголовок пакета в потоке битов или в сегментах маркеров PPM или PPT. Данный маркер не обозначает начало данных пакета. Если в потоке битов отсутствуют заголовки пакетов (то есть используются сегменты маркеров PPM или PPT), этот маркер не используют в потоке битов.

Применение: Должен использоваться только в том случае, если он обозначен в соответствующем сегменте маркера COD (см. А.6.1). Размещается непосредственно после заголовка пакета.

Если требуются маркеры EPH (с помощью сигнализации в сегменте маркера COD, см. А.6.1), каждый заголовок пакета в заданной части тайла должен быть размещен в конце сегмента маркера EPH. Если заголовки пакетов размещают в сегментах маркеров PPM или PPT (см. А.7.4 и А.7.5), то маркеры EPH должны располагаться после заголовков пакетов в сегментах маркеров PPM или PPT.

Длина: Неизменяемая.

EPH: Код маркера

Таблица А.41 – Значения параметров конца заголовка пакета

Параметр	Размер (биты)	Значения
EPH	16	0xFF92

А.9 Информационные сегменты маркера

Эти сегменты маркера являются строго информационными, и нет необходимости их декодировать. Однако эти сегменты маркера могут быть полезны при синтаксическом анализе или декодировании. При использовании файлового формата типа JP2 (см. Приложение I) можно получить дополнительную информацию об источнике и характеристиках изображения.

А.9.1 Регистрация компонента (CRG)

Функция: Производит специальную взаимную регистрацию компонентов. Для удобства кодирования образцы компонентов располагают на точках опорной сетки, которые являются целыми кратными от XR_{siz} и YR_{siz} (см. А.5.1). Однако это может не соответствовать представлению изображения. Сегмент маркера CRG описывает "центр массы" каждого образца компонента по отношению к разделению. Этот сегмент маркера не оказывает влияния на декодирование кодированного потока.

ПРИМЕЧАНИЕ. – Сдвиг регистрации компонента происходит по отношению к сдвигу изображения (XO_{siz} и YO_{siz}) и разделению компонентов (XR_{siz}^i и YR_{siz}^i). Например, точка на горизонтальной опорной сетке для крайних левых образцов компонента c : $XR_{siz}^c \lceil XO_{siz} \mid XR_{siz}^c \rceil$. (Аналогично для вертикального направления). Горизонтальный сдвиг в этом сегменте маркера является дополнением к этому сдвигу.

Применение: Только в основном заголовке. Может быть использована только одна регистрация CRG в основном заголовке и приложениях для всех тайлов.

Длина: Переменная, зависит от числа компонентов.

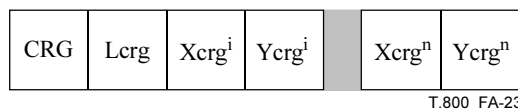


Рисунок А.23 – Синтаксис регистрации компонента

- CRG:** Код маркера. В таблице А.42 приведены размеры и значения символов, а также параметры для сегмента маркера регистрации компонента.
- Lcrg:** Длина сегмента маркера в байтах (не включая сам маркер).
- Xcrgⁱ:** Значение горизонтального сдвига в единицах 1/65536 от горизонтального разделителя XR_{siz}^i для i -го компонента. Таким образом, диапазон значений составляет от 0/65536 (образец занимает свою точку на опорной сетке) до $XR_{siz}^c 65535/65536$ (сразу перед точкой, которую занимает на опорной сетке следующий образец). Это значение повторяется для каждого компонента.
- Ycrgⁱ:** Значение вертикального сдвига в единицах 1/65536 от вертикального разделителя YR_{siz}^i для i -го компонента. Таким образом, диапазон значений составляет от 0/65536 (образец занимает свою точку на опорной сетке) до $YR_{siz}^c 65535/65536$ (сразу перед точкой, которую занимает на опорной сетке следующий образец). Это значение повторяется для каждого компонента.

Таблица А.42 – Значения параметров регистрации компонентов

Параметр	Размер (биты)	Значения
CRG	16	0xFF63
Lcrg	16	6–65534
Xcrg ⁱ	16	0–65535
Ycrg ⁱ	16	0–65535

А.9.2 Комментарии (COM)

Функция: Допускается в виде неструктурированных данных в основном заголовке и заголовке части тайла.

Применение: Основной заголовок и заголовок части тайла. Может повторяться столько раз, сколько требуется в этих заголовках. Сегменты маркера не влияют на процесс декодирования.

Длина: Переменная, зависит от длины сообщения.

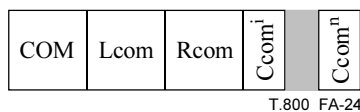


Рисунок А.24 – Синтаксис комментария

COM: Код маркера. В таблице А.43 приведены размеры и значения символов, а также параметры для сегмента маркера комментария.

Lcom: Длина сегмента маркера в байтах (не включая сам маркер).

Rcom: Значение регистрации сегмента маркера.

Ccomⁱ: Байт неструктурированных данных.

Таблица А.43 – Значения параметров комментария

Параметр	Размер (биты)	Значения
COM	16	0xFF64
Lcom	16	5–65535
Rcom	16	Табл. А.44
Ccom ⁱ	8	0–255

Таблица А.44 – Значения регистрации для параметра Rcom

Значения	Значения регистрации
0	Общего пользования (бинарные значения)
1	Общего пользования (значения ИСО/МЭК 8859-15 (Латинские))
	Все прочие значения зарезервированы

А.10 Ограничения по соответствию кодированного потока данной Рекомендации | Международному стандарту

Ограничения на кодированный поток наложены для того, чтобы обеспечить широкое взаимодействие кодированного потока JPEG 2000. "Ограничения кодированного потока" состоят из двух наборов характеристик: Profile-0 и Profile-1. Обозначение "Нет ограничений" означает, что эта Рекомендация | Международный стандарт могут считаться набором характеристик. Характеристики же Profile-0 и Profile-1 определяются следующим образом.

Максимальное взаимодействие может быть получено для кодированных потоков, соответствующих характеристикам Profile-0, а среднее взаимодействие – характеристикам, соответствующим Profile-1.

Таблица А.45 – Ограничения кодированных потоков

Ограничения	Характеристики Profile-0	Характеристики Profile-1
Сегмент маркера SIZ		
Указание характеристик	$Rsiz = 1$	$Rsiz = 2$
Размер изображения	$Xsiz, Ysiz < 2^{31}$	$Xsiz, Ysiz < 2^{31}$
Тайлы	Размер тайлов 128×128 : $YTsiz = XTsiz = 128$ или один тайл на все изображение: $YTsiz + YTOsiz \geq Ysiz$ $XTsiz + XTOsiz \geq Xsiz$	$XTsiz \min(XRsiz^i, YRsiz^i) \geq 1024$ $XTsiz = YTsiz$ или один тайл на все изображение: $YTsiz + YTOsiz \geq Ysiz$ $XTsiz + XTOsiz \geq Xsiz$
Изображение и тайл оригинала	$XOsiz = YOsiz = XTOsiz = YTOsiz = 0$	$XOsiz, YOsiz, XTOsiz, YTOsiz < 2^{31}$
Сегмент маркера RGN	$SPrgn \leq 37$	$SPrgn \leq 37$
Вторичный образец	$XRsiz^i = 1, 2, \text{ или } 4$ $YRsiz^i = 1, 2, \text{ или } 4$	Нет ограничений
Блоки кодирования		
Размер блока кодирования	$xcb = ycb = 5$ или $xcb = ycb = 6$	$xcb \leq 6, ycb \leq 6$
Тип блока кодирования	$SPcod, SPcoc = 00sp\ vtra$, где $a = r = v = 0, a\ t, p, s = 0$ или 1 ПРИМЕЧАНИЕ 1. – $t = 1$ для завершения каждого кодирующего прохода $p = 1$ для завершения предсказания $s = 1$ для знаков сегментации	Нет ограничений
Расположение маркеров		
Заголовки упакованных пакетов (PPM, PPT)	Не разрешены	Нет ограничений
COD, COC, QCD, QCC	Только в основном заголовке	Нет ограничений
Требования к подмножествам		
Разрешение LL	Если используют только один тайл для всего изображения, то $(Xsiz - XOsiz) D(I) \leq 128$ и $(Ysiz - YOsiz) D(I) \leq 128$, где $D(I) = 2^{\text{число_уровней_декомпозиции}}$ в SPcod или SPcoc для компонента $I = \text{от } 0 \text{ до } 3$	Для каждого тайла в изображении $\lfloor x1 D(i) \rfloor - \lfloor tx0/D(i) \rfloor \leq 128$ и $\lfloor ty1/D(i) \rfloor - \lfloor ty0/D(i) \rfloor \leq 128$, где $D(I) = 2^{\text{число_уровней_декомпозиции}}$ в SPcod или SPcoc для компонента $I = \text{от } 0 \text{ до } 3$. ПРИМЕЧАНИЕ 2. – значения $tx0, tx1, ty0$ и $ty1$ определены равенствами от (B-7) до (B-10).
Анализ	Если присутствует маркер POC, этот маркер должен удовлетворять условиям $RSPOC0 = 0$ и $CSPOC0 = 0$. ПРИМЕЧАНИЕ 3. – Некоторые совместимые декодеры должны декодировать только пакеты, связанные с первой последовательностью.	Нет ограничений
Части тайла	Части тайла с $TPsot = 0$ для каждого тайла перед частями тайла с $TPsot > 0$, Части тайла от $Isot = 0$ до $Isot = \text{число тайлов} - 1$, в порядке последовательности для всех частей тайла с $TPsot = 0$	Нет ограничений
Предельный размер	"Предельный размер", определенный SPcod или SPcoc (таблицы А.15 и А.21), должен быть достаточно большим, чтобы существовала только одна граница для всех уровней разрешения с размерами меньшими или равными 128 на 128 . ПРИМЕЧАНИЕ 4. – Предельный размер $PPx \geq 7$ и $PPy \geq 7$, достаточный, чтобы гарантировать единый размер на поддиапазон, если $XOsiz = 0$ и $YOsiz = 0$.	Нет ограничений

Приложение В

Изображение и упорядочение данных сжатого изображения

(Это Приложение составляет неотъемлемую часть данной Рекомендации | Международного стандарта)

В настоящем Приложении и всех его подразделах блок-схемы и таблицы являются нормативными (обязательными) только в том смысле, что они определяют выходные данные, которые при альтернативных применениях должны повторяться.

В этом приложении описаны объекты различной структуры и их организация в кодированном потоке: компоненты, тайлы, поддиапазоны и производные этих объектов.

В.1 Введение в концепцию структуры данных изображения

Опорная сетка создает механизм для параллельной регистрации компонентов и для определения подмножеств опорной сетки, например, площади изображения и тайлов.

Компоненты состоят из двухразмерных массивов образцов. Каждому компоненту c присвоены параметры XR_{siz}^c , YR_{siz}^c (см. А.5.1), которые определяют размещение образцов компонентов в точках опорной сетки (но не наоборот). Такое размещение приводит к регистрации компонентов друг с другом, которое используют исключительно для целей кодирования.

Каждый компонент разделяют на тайлы. Это разделение согласуется с распределением тайлов на опорной сетке. Компоненты тайла кодируют независимо. Каждый компонент тайла частотно преобразуют в несколько уровней декомпозиции, которые связаны с уровнями разрешения (см. Приложение F). Каждый уровень разрешения состоит либо из поддиапазонов HL, LH и HH для одного уровня декомпозиции, либо из поддиапазона N_L LL. Уровней разрешения на единицу больше, чем уровней декомпозиции.

Каждый поддиапазон имеет свое происхождение. Граничные условия поддиапазона единственны для каждого из поддиапазонов HL, LH и HH.

ПРИМЕЧАНИЕ. – Это условие отличается от частотных (волновых) диаграмм, на которых все поддиапазоны одного компонента располагаются в едином пространстве.

Границы областей и блоки кодирования определяют уровнем разрешения и поддиапазоном. Следовательно, они могут изменяться в зависимости от компонента тайла. Границы областей определяют таким образом, чтобы блоки кодирования были тщательно подогнаны, то есть "выровнены" друг с другом.

На сопроводительных рисунках показаны границы и оси координат. В каждом случае образцы или соответствующие коэффициенты на левой и верхней границе включают в заданную область, а образцы и коэффициенты на правой и/или нижней границах в эту область не включают.

Кроме того, в соответствующих формулах многие из переменных принимают значения, которые могут изменяться явным образом (согласно синтаксису, описанному в Приложении А) или неявным (согласно восстановлению). Для удобства записи в последующих обсуждениях некоторые зависимости опущены.

В.2 Размещение компонентов на опорной сетке

Все компоненты (и многие другие структуры в этом Приложении) определены по отношению к опорной сетке. Различные параметры, определяющие опорную сетку, показаны на рисунке В.1. Опорная сетка является прямоугольной сеткой точек, индексированных от $(0, 0)$ до $(X_{siz} - 1, Y_{siz} - 1)$. "Площадь изображения" определена на опорной сетке параметрами размера (X_{siz}, Y_{siz}) и (X_{Osiz}, Y_{Osiz}) . Более точно, площадь изображения на опорной сетке определена верхней левой точкой опорной сетки с координатами (X_{Osiz}, Y_{Osiz}) и нижней правой точкой опорной сетки с координатами $(X_{siz} - 1, Y_{siz} - 1)$.

Образцы компонентов c на опорной сетке являются целыми кратными от (XR_{siz}^c, YR_{siz}^c) . Область каждого компонента является вариантом дополнительного образца на опорной сетке с координатами $(0, 0)$ в качестве общей точки для каждого компонента. Ряды образцов занимают точки опорной сетки, которые являются целыми от умножения на XR_{siz}^c , а столбцы образцов занимают аналогичные точки, полученные умножением на YR_{siz}^c . Только те образцы, которые расположены внутри площади изображения, действительно являются компонентами изображения. Таким образом, образцы компонента c размещены в прямоугольнике, в котором верхний левый образец имеет координаты (x_0, y_0) , а нижний правый – $(x_1 - 1, y_1 - 1)$, где:

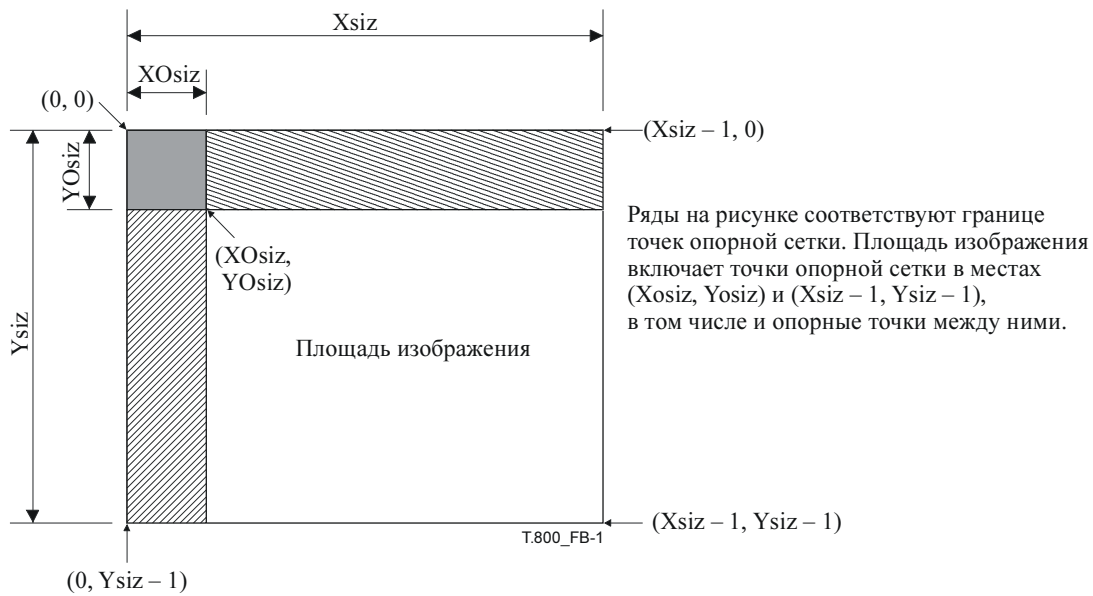


Рисунок В.1 – Диаграмма опорной сетки

$$x_0 = \left\lfloor \frac{X_{Osiz}}{XR_{siz}^c} \right\rfloor \quad x_1 = \left\lfloor \frac{X_{siz}}{XR_{siz}^c} \right\rfloor \quad y_0 = \left\lfloor \frac{Y_{Osiz}}{YR_{siz}^c} \right\rfloor \quad y_1 = \left\lfloor \frac{Y_{siz}}{YR_{siz}^c} \right\rfloor. \quad (B-1)$$

Следовательно, размеры компонента c :

$$(ширина, высота) = (x_1 - x_0, y_1 - y_0). \quad (B-2)$$

Все параметры X_{siz} , Y_{siz} , X_{Osiz} , Y_{Osiz} , XR_{siz}^c и YR_{siz}^c определены в сегменте маркера SIZ (см. А.5.1).

ПРИМЕЧАНИЕ 1. – Тот факт, что всем компонентам свойственен сдвиг изображения (X_{Osiz}, Y_{Osiz}) и размера (X_{siz}, Y_{siz}) , приводит к регистрации этих компонентов.

ПРИМЕЧАНИЕ 2. – На рисунке В.2 приведен пример из трех компонентов, размещенных на опорной сетке. На рисунке В.3 показана площадь изображения с конкретным сдвигом изображения на разные величины (XR_{siz}, YR_{siz}) . Показаны также верхние левые координаты образца в области изображения компонента, которые включены в площадь изображения.

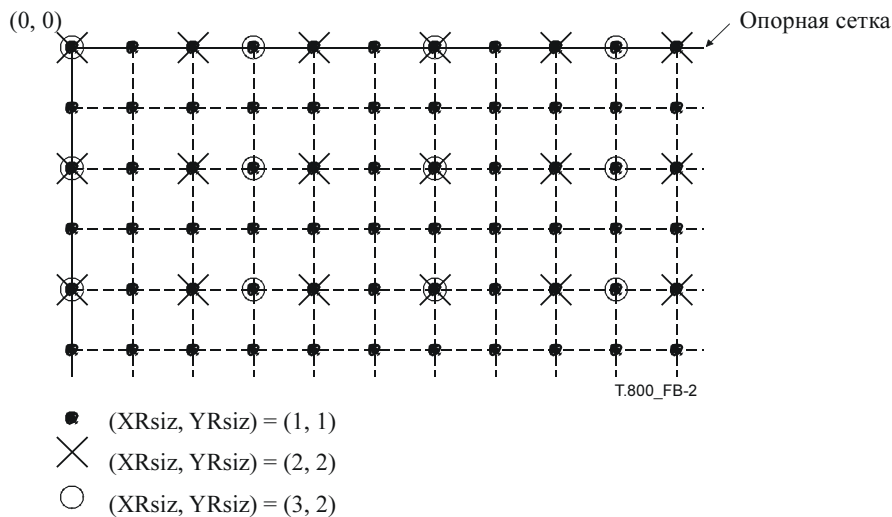


Рисунок В.2 – Расположение образца компонента на опорной сетке с разными величинами XR_{siz} и YR_{siz}

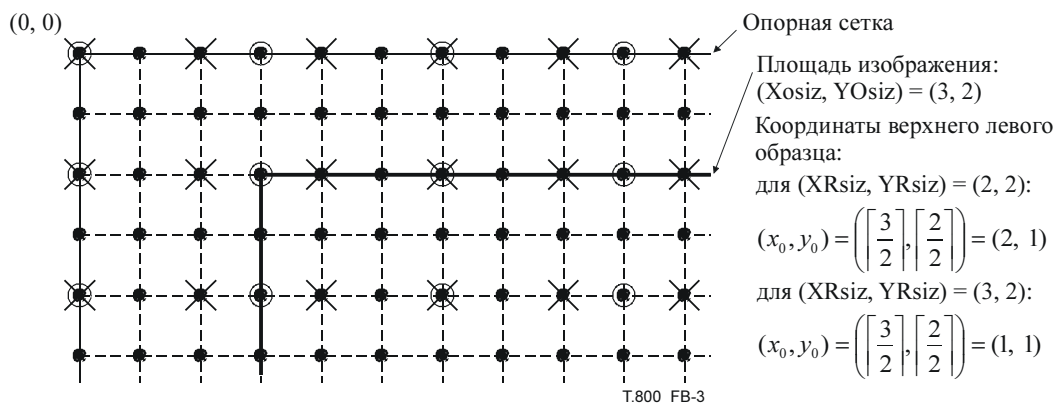


Рисунок В.3 – Пример расположения верхнего левого образца компонента

В.3 Разделение области изображения на тайлы и компоненты тайла

Опорная сетка разбивается на массивы тайлов правильной прямоугольной формы. Размер и сдвиг тайла заданы на опорной сетке парами размеров $(XTsiz, YTsiz)$ и $(XTOsiz, YTOsiz)$ соответственно. Все эти параметры находятся в сегменте маркера SIZ (см. А.5.1).

Каждый тайл на опорной сетке имеет ширину $XTsiz$ и высоту $YTsiz$. Верхний левый угол первого тайла (тайл 0) сдвинут от верхнего левого угла опорной сетки на $(XTOsiz, YTOsiz)$. Тайлы нумеруются в растровом порядке, то есть индексируются параметром $Isot$ сегмента маркера SOT согласно А.4.2. Таким образом, верхние левые координаты первого тайла относительно опорной сетки: $(XTOsiz, YTOsiz)$. Это соотношение показано на рисунке В.4.

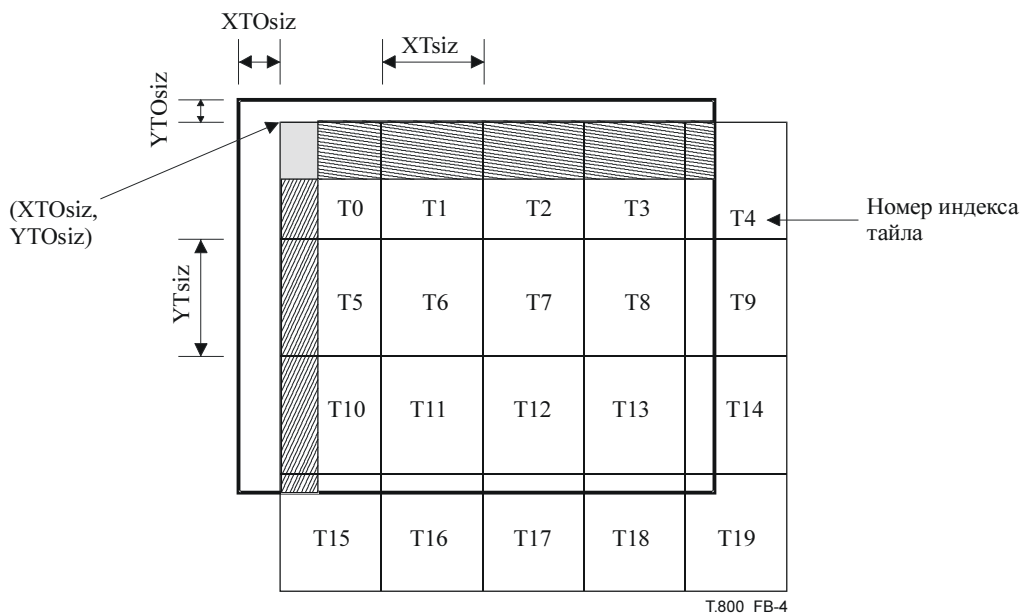


Рисунок В.4 – Диаграмма размещения тайлов на опорной сетке

Сдвиги сетки тайла $(XTOsiz, YTOsiz)$ должны быть не больше сдвигов площади изображения. Это выражается следующими соотношениями:

$$0 \leq XTOsiz \leq XOsiz \quad 0 \leq YTOsiz \leq YOsiz . \quad (B-3)$$

Кроме того, сумма размеров и сдвига тайла должна быть больше сдвига площади изображения. Это гарантирует, что в первом тайле (тайл 0) будет, по крайней мере, одна точка из площади изображения, что выражается следующими соотношениями:

$$XTsiz + XTOsiz > XOsiz \quad YTsiz + YTOsiz > YOsiz . \quad (B-4)$$

Число тайлов в направлениях X ($numXtiles$) и Y ($numYtiles$):

$$numXtiles = \left\lceil \frac{Xsiz - XTOsiz}{XTsiz} \right\rceil \quad numYtiles = \left\lceil \frac{Ysiz - YTOsiz}{YTsiz} \right\rceil. \quad (B-5)$$

Для целей данного описания полезно индексировать тайлы в обозначениях по горизонтали и вертикали. Положим, что p – горизонтальный индекс тайла в диапазоне от 0 до $numXtiles - 1$ и q – вертикальный индекс в диапазоне от 0 до $numYtiles - 1$ определены из индекса тайла следующим образом:

$$p = \text{mod}(t, numXtiles) \quad q = \left\lceil \frac{t}{numXtiles} \right\rceil, \quad (B-6)$$

где t – индекс тайла на рисунке В.4.

Тогда координаты тайла на опорной сетке выражаются следующими равенствами:

$$tx_0(p, q) = \max(XTOsiz + p \cdot XTsiz, XOsiz), \quad (B-7)$$

$$ty_0(p, q) = \max(YTOsiz + q \cdot YTsiz, YOsiz), \quad (B-8)$$

$$tx_1(p, q) = \min(XTOsiz + (p + 1) \cdot XTsiz, Xsiz), \quad (B-9)$$

$$ty_1(p, q) = \min(YTOsiz + (q + 1) \cdot YTsiz, Ysiz), \quad (B-10)$$

где $tx_0(p, q)$ и $ty_0(p, q)$ – координаты верхнего левого угла тайла, а $tx_1(p, q) - 1$ и $ty_1(p, q) - 1$ – координаты нижнего правого угла тайла. Часто мы опускаем координаты тайла, которые относятся к частному тайлу, и считаем координатами (tx_0, ty_0) и (tx_1, ty_1) .

Таким образом, размеры тайла на опорной сетке:

$$(tx_1 - tx_0, ty_1 - ty_0). \quad (B-11)$$

Внутри области компонента изображения i координаты верхнего левого образца – (tcx_0, tcy_0) , а нижнего правого – $(tcx_1 - 1, tcy_1 - 1)$, где:

$$tcx_0 = \left\lceil \frac{tx_0}{XRsiz^i} \right\rceil \quad tcx_1 = \left\lceil \frac{tx_1}{XRsiz^i} \right\rceil \quad tcy_0 = \left\lceil \frac{ty_0}{YRsiz^i} \right\rceil \quad tcy_1 = \left\lceil \frac{ty_1}{YRsiz^i} \right\rceil, \quad (B-12)$$

так что размеры компонента тайла:

$$(tcx_1 - tcx_0, tcy_1 - tcy_0). \quad (B-13)$$

В.4 Пример размещения компонентов на опорной сетке (информативный)

Следующий пример приведен для иллюстрации размещения компонентов изображения на опорной сетке и площади, которое вытекает из разбиения тайла на компоненты с различными показателями дополнительных образцов. Пример подразумевает приложение, в котором предполагается сжать оригинал изображения с соотношением 16:9, в соответствии с данной Рекомендацией | Международным стандартом. Выбор размера и сдвига изображения, а также размера и сдвига тайла сделан таким образом, чтобы изображение с размерами 4:3 могло быть сдвинуто от центра оригинала изображения. На рисунке В.5 показаны площади опорной сетки и изображения вместе со структурой тайлов, которая должна присутствовать в этом примере.

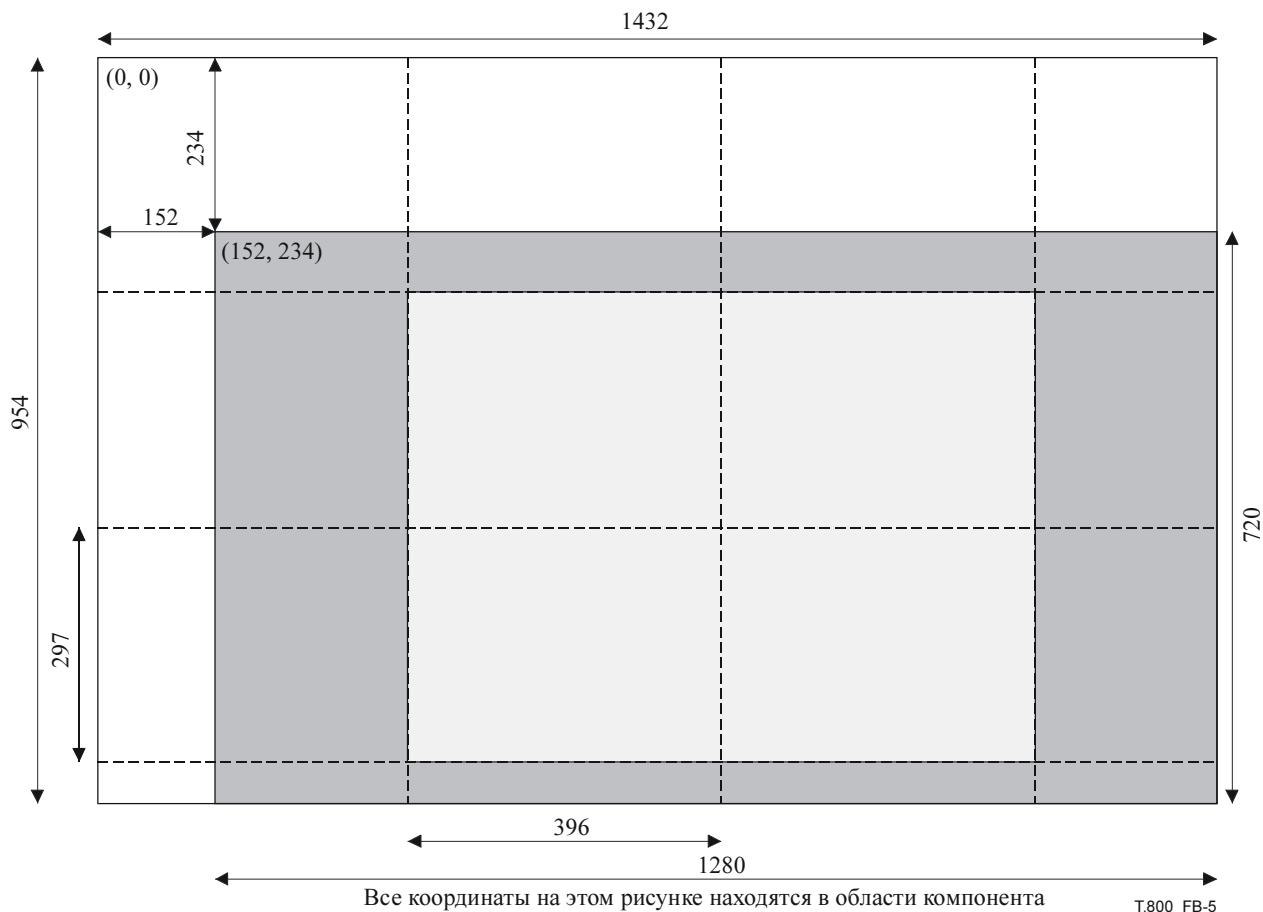
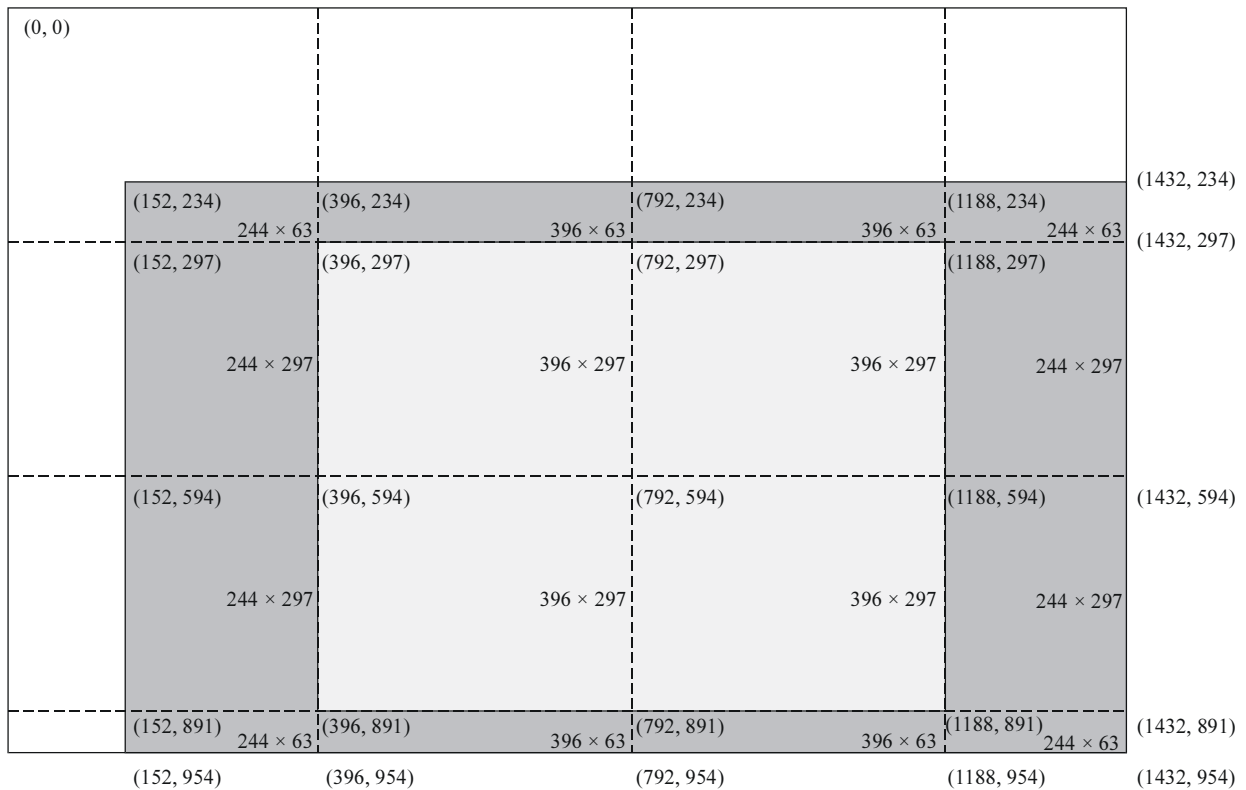


Рисунок В.5 – Пример опорной сетки

Положим размеры опорной сетки (X_{siz} , Y_{siz}) равными (1432, 954). В этом случае в изображении должны содержаться два компонента (индексы компонентов будут представлены как $i = 0, 1$). Показатели дополнительных образцов этих двух компонентов, XR_{siz}^i и YR_{siz}^i , по отношению к опорной сетке: $XR_{siz}^0 = YR_{siz}^0 = 1$ и $XR_{siz}^1 = YR_{siz}^1 = 2$. Сдвиг изображения составит: $(XO_{siz}, YO_{siz}) = (152, 234)$. Получив эти параметры, из равенства (B-1) можно определить размеры этих двух компонентов изображения. Верхний левый угол компонента 0 будет находиться в точке $(\lceil 152/1 \rceil, \lceil 234/1 \rceil) = (152, 234)$, а нижний правый угол компонента 0 – в точке $(\lceil 1432/1 \rceil - 1, \lceil 954/1 \rceil - 1) = (1431, 953)$. Действительный размер компонента 0 составит 1280 образцов в ширину и 720 в высоту. Верхний левый угол компонента 1 будет находиться в точке $(\lceil 152/2 \rceil, \lceil 234/2 \rceil) = (76, 117)$, а нижний правый угол компонента 1 – в точке $(\lceil 1432/2 \rceil - 1, \lceil 954/2 \rceil - 1) = (715, 476)$. Действительный размер компонента 1 составит 640 образцов в ширину и 360 в высоту.

Тайлы были выбраны исходя из соотношения размеров 4:3. В этом примере параметры (XT_{siz}, YT_{siz}) составят (396, 297), а сдвиги тайла (XTO_{siz}, YTO_{siz}) составят (0, 0). Число тайлов в направлениях x и y находится из равенства (B-5): $numX_{tiles} = \lceil 1432/396 \rceil = 4$, $numY_{tiles} = \lceil 954/297 \rceil = 4$. Разбитые на тайлы компоненты изображения будут содержать в целом $t = 16$ тайлов с индексами сетки тайлов, p и q , в диапазоне $0 \leq p, q < 4$. Теперь можно вычислить расположение тайлов в каждом компоненте изображения. Для этого следует определить значения tx_0 , tx_1 , ty_0 и ty_1 из равенств (B-7), (B-8), (B-9) и (B-10). Поскольку p и q находятся в одном множестве допустимых значений, будут использованы обозначения '0:3', которые относятся к последовательности значений $\{0, 1, 2, 3\}$, и обозначение '*' для указания того, что результат действителен для всех допустимых значений. Значения tx_0 находят как $tx_0(0:3, *) = \{152, 396, 792, 1188\}$, а значения tx_1 – как $tx_1(0:3, *) = \{396, 792, 1188, 1432\}$. Значения ty_0 и ty_1 равны: $ty_0(*, 0:3) = \{234, 297, 594, 891\}$, $ty_1(*, 0:3) = \{297, 594, 891, 954\}$.

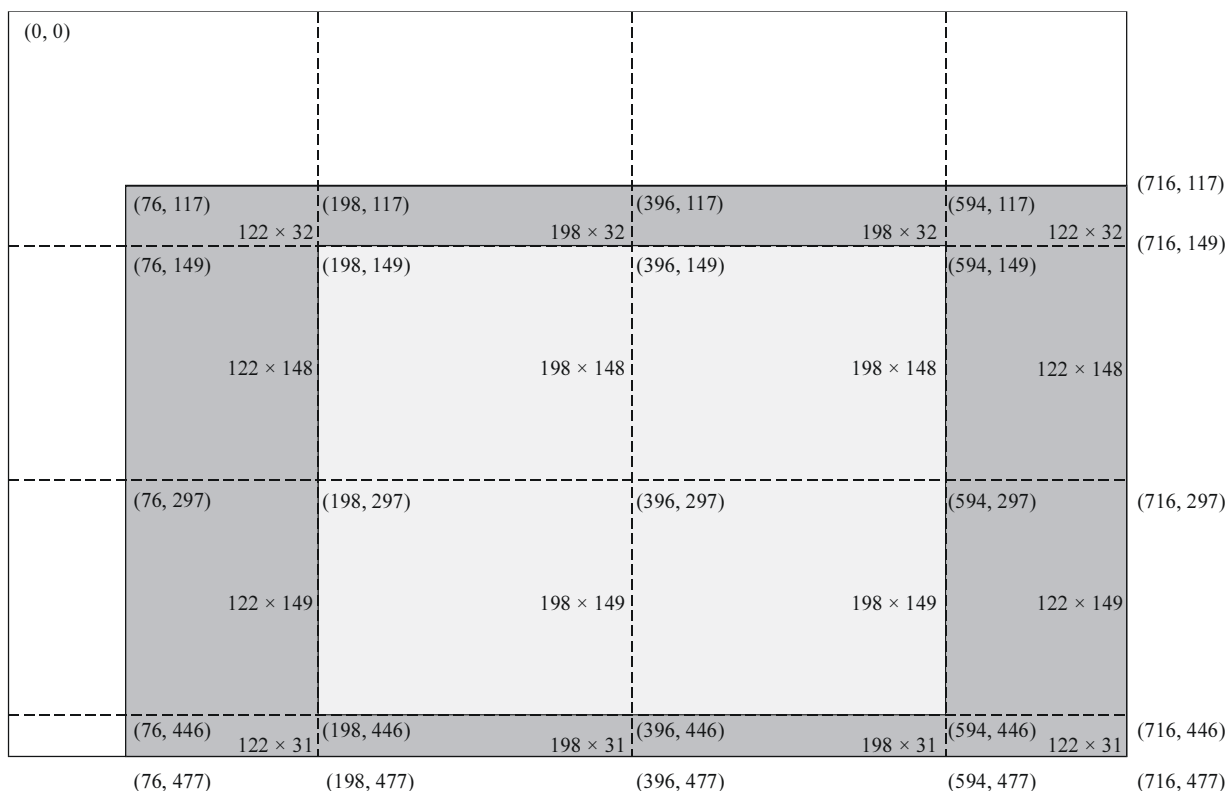


Распределение тайла для компонента 0. Все координаты на этом рисунке находятся в области компонента. T.800_FB-6

Рисунок В.6 – Пример размеров и расположений тайлов для компонента 0

После того, как найдены значения tx_0 , tx_1 , ty_0 и ty_1 , можно определить положения и размеры всех тайлов для каждого компонента. Для этого используем равенство (В-12). Соответствующие расположения и размеры для компонента 0 показаны на рисунке В.6, а для компонента 1 – на рисунке В.7. Представляют интерес "внутренние" тайлы на рисунках: (тайлы (1, 1), (1, 2), (2, 1) и (2, 2)). Эти тайлы не ограничены протяженностью площади изображения. В компоненте 0 все такие тайлы имеют одинаковый размер. Такая закономерность – результат того, что показатели дополнительных образцов для этого компонента $(XR_{siz}^0, YR_{siz}^0) = (1, 1)$. Однако в компоненте 1 такие тайлы имеют неодинаковые размеры, так как $(XR_{siz}^1, YR_{siz}^1) = (2, 2)$. Заметим, что тайлы (1, 1) и (2, 1) оба имеют размеры 198 x 148, а тайлы (1, 2) и (2, 2) – размеры 198 x 149. Из этого следует, что число образцов во внутренних тайлах компонента может варьироваться в зависимости от конкретного сочетания размеров тайла и параметров дополнительных образцов.

При таком выборе опорной сетки, сдвига изображения, размера тайла и сдвига тайла, кодированное изображение может быть непосредственно вставлено в требуемую внутреннюю область. Для каждого компонента могут быть выбраны четыре тайла, которые будут представлять усеченное изображение с размерами на опорной сетке (792, 594). Если выполнено такое усечение, нет необходимости вновь кодировать эти тайлы, но значения некоторых параметров опорной сетки должны измениться. Сдвиги изображения должны быть учтены в координатах усеченных расположений, так что $(XOsiz', YOsiz') = (396, 297)$, где $(XOsiz', YOsiz')$ – это сдвиги изображений усеченного изображения. Аналогичным образом размер изображения должен быть подстроен так, чтобы отражать усеченный размер: $(Xsiz', Ysiz') = (1188, 891)$, где $(Xsiz', Ysiz')$ – размеры усеченной опорной точки. Наконец, сдвиги тайлов теперь не равны нулю, а должны составлять $(XTOsiz', YTOsiz') = (396, 297)$, где $(XTOsiz', YTOsiz')$ – сдвиги тайлов на усеченной опорной сетке.



Распределение тайла для компонента 1. Все координаты на этом рисунке находятся в области компонента. T.800_FB-7

Рисунок В.7 – Пример размеров и расположений тайлов для компонента 1

В.5 Разделение преобразованного компонента тайла на уровни разрешения и поддиапазоны

Как объясняется в Приложении F, каждый компонент тайла частотно преобразуется с N_L уровнями декомпозиции. Таким образом, получают $N_L + 1$ различных уровней разрешения, обозначенных как $r = 0, 1, \dots, N_L$. Самый нижний уровень разрешения, $r = 0$, представлен диапазоном $N_L LL$. В общем, сокращенным вариантом разрешения компонента тайла с уровнем разрешения r является поддиапазон nLL , где $n = N_L - r$. В этом разделе описаны величины такого сокращенного разрешения.

Координатами заданного компонента тайла по отношению к опорной сетке для конкретного уровня разрешения r являются координаты верхнего левого образца (trx_0, try_0) и нижнего правого образца ($trx_1 - 1, try_1 - 1$), где:

$$trx_0 = \left\lceil \frac{tcx_0}{2^{N_L - r}} \right\rceil \quad try_0 = \left\lceil \frac{tcy_0}{2^{N_L - r}} \right\rceil \quad trx_1 = \left\lceil \frac{tcx_1}{2^{N_L - r}} \right\rceil \quad try_1 = \left\lceil \frac{tcy_1}{2^{N_L - r}} \right\rceil. \quad (B-14)$$

Таким же образом координаты тайла можно поместить в любой конкретный поддиапазон b , имеющий координаты верхнего левого образца (tbx_0, tby_0) и нижнего правого образца ($tbx_1 - 1, tby_1 - 1$), где:

$$tbx_0 = \left\lceil \frac{tcx_0 - (2^{n_b} - 1) \cdot xo_b}{2^{n_b}} \right\rceil \quad tby_0 = \left\lceil \frac{tcy_0 - (2^{n_b} - 1) \cdot yo_b}{2^{n_b}} \right\rceil$$

$$tbx_1 = \left\lceil \frac{tcx_1 - (2^{n_b} - 1) \cdot xo_b}{2^{n_b}} \right\rceil \quad tby_1 = \left\lceil \frac{tcy_1 - (2^{n_b} - 1) \cdot yo_b}{2^{n_b}} \right\rceil, \quad (B-15)$$

где n_b – уровень декомпозиции, связанный с поддиапазоном b , указан в Приложении F, а величины (xo_b, yo_b) приведены в таблице В.1.

Таблица В.1 – Величины (x_{0b}, y_{0b}) для поддиапазона **b**

Поддиапазон	x_{0b}	y_{0b}
n_bLL	0	0
n_bHL (горизонтальный высокочастотный)	1	0
n_bLH (вертикальный высокочастотный)	0	1
n_bHH	1	1

ПРИМЕЧАНИЕ. – Как указано в В.1, поддиапазоны отличаются друг от друга.

Для каждого поддиапазона эти координаты определяют границы тайла в различных областях поддиапазонов. Кроме того, ширина каждого поддиапазона внутри своей области (для текущего уровня разрешения) задана как $tbx_1 - tbx_0$, а высота как $tby_1 - tby_0$.

В.6 Разделение уровней разрешения на граничные области

Рассмотрим конкретный компонент тайла и уровень разрешения, для которых, как уже указывалось, граничные координаты образца в области изображения с сокращенным уровнем разрешения составляют (trx_0, try_0) и $(trx_1 - 1, try_1 - 1)$. На рисунке В.8 показано разделение уровня разрешения этого компонента тайла на области. Область привязана к точке $(0, 0)$, так что верхний левый угол каждой области разделения расположен в целочисленных кратных точках $(2^{PPx}, 2^{PPy})$, где PPx и PPy – отмечены в сегментах маркеров COD или СОС (см. А.6.1 и А.6.2). Значения PPx и PPy могут отличаться для компонента тайла и уровня разрешения. Значения PPx и PPy должны быть, по крайней мере, 1 для всех уровней разрешения, за исключением $r = 0$, где допускается нулевое значение.

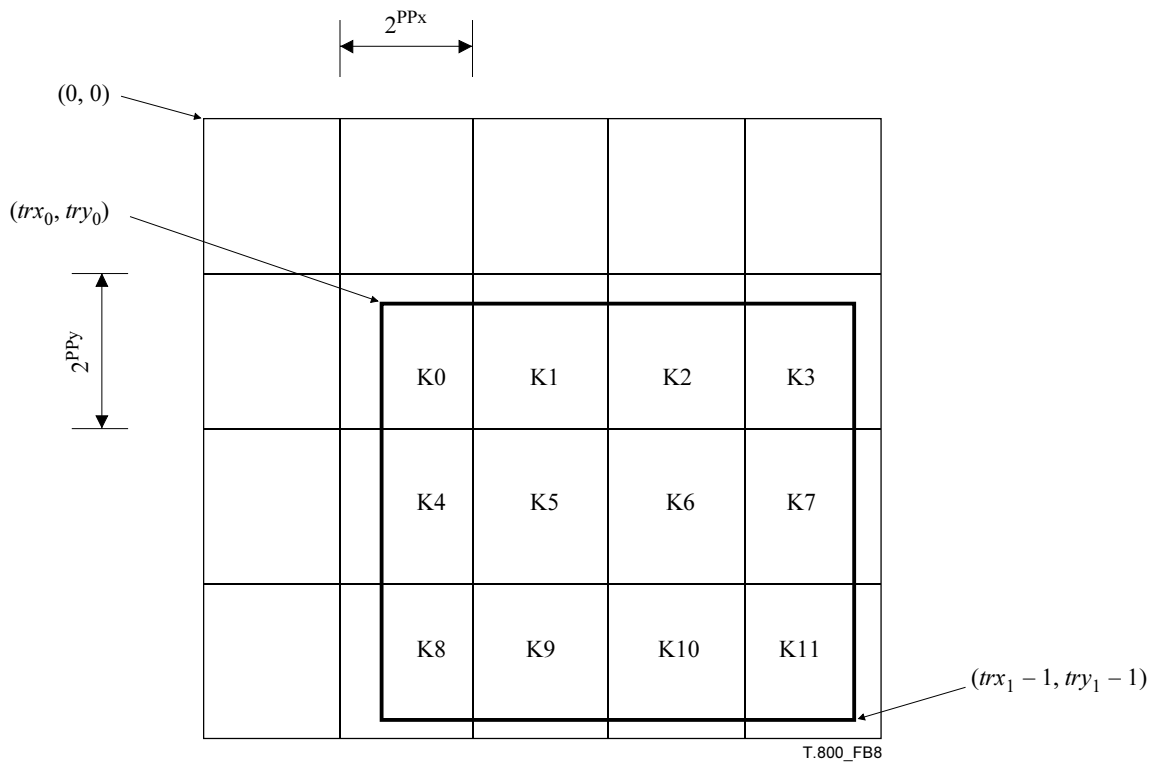


Рисунок В.8 – Граничные области одного сокращенного разрешения

Число граничных областей, которые занимает компонент тайла с уровнем разрешения r , задано как:

$$\begin{aligned}
 numprecinctswide &= \begin{cases} \left\lceil \frac{trx_1}{2^{PPx}} \right\rceil - \left\lfloor \frac{trx_0}{2^{PPx}} \right\rfloor & trx_1 > trx_0 \\ 0 & trx_1 = trx_0 \end{cases} &
 numprecinctshigh &= \begin{cases} \left\lceil \frac{try_1}{2^{PPy}} \right\rceil - \left\lfloor \frac{try_0}{2^{PPy}} \right\rfloor & try_1 > try_0 \\ 0 & try_1 = try_0 \end{cases}
 \end{aligned} \quad (B-16)$$

Как объясняется ниже, даже если равенство (B-16) указывает, что значения $numprecinctswide$ и $numprecinctshigh$ являются ненулевыми, некоторые или все граничные области все же могут оставаться пустыми. Индекс граничной области начинается с 0 и оканчивается на $numprecincts - 1$, где $numprecincts = numprecinctswide * numprecinctshigh$ в растровом порядке (см. рисунок В.8). Этот индекс используют для определения порядка следования в кодированном потоке пакетов, соответствующих каждой граничной области, как это поясняется в В.12.

Для частного компонента тайла и уровня разрешения возможен случай, когда значение $numprecincts$ является нулем. Тогда для этого компонента тайла не существует никаких пакетов и уровней разрешения.

Возможен случай, когда граничная область пустая. Это значит, что от соответствующего уровня разрешения отсутствуют коэффициенты поддиапазона, которые действительно вносят вклад в эту граничную область. Так, например, может получиться для нижнего правого компонента тайла вследствие распределения образцов относительно опорной сетки. В этом случае все же каждый пакет, соответствующий этой области, должен присутствовать в кодированном потоке (см. В.9).

В.7 Разделение поддиапазонов на блоки кодирования

Поддиапазоны разделяют на прямоугольные блоки кодирования с целью моделирования и кодирования коэффициентов. Размер каждого блока кодирования определяют двумя параметрами, xcb и ycb , о которых сообщается в сегментах маркеров COD и COC (см. А.6.1 и А.6.2). Размер блока кодирования одинаков для всех уровней разрешения. Однако для каждого уровня разрешения размер блока кодирования лимитирован размером граничной области. Размер блока кодирования для каждого поддиапазона при конкретном уровне разрешения определяют как $2^{xcb'} \times 2^{ycb'}$, где:

$$xcb' = \begin{cases} \min(xcb, PPx - 1), & \text{для } r > 0 \\ \min(xcb, PPx), & \text{для } r = 0 \end{cases} \quad (\text{B-17})$$

а:

$$ycb' = \begin{cases} \min(ycb, PPy - 1), & \text{для } r > 0 \\ \min(ycb, PPy), & \text{для } r = 0 \end{cases} \quad (\text{B-18})$$

Эти равенства отражают тот факт, что размеры блока кодирования ограничены размерами области и самого блока кодирования, параметры которого, xcb и ycb , идентичны для всех поддиапазонов компонента тайла. Подобно граничной области, разделение блока кодирования привязано к точке (0, 0), как показано на рисунке В.9. Таким образом, при разделении блока кодирования все первые ряды блоков расположены в точках $y = m2^{ycb'}$, а первые столбцы – в точках $x = n2^{xcb'}$, где m и n – целые числа.

ПРИМЕЧАНИЕ. – При разделении блоки кодирования могут выходить за границы коэффициентов поддиапазонов. В этом случае кодируются только коэффициенты, лежащие внутри поддиапазона. При этом используют методы кодирования, описанные в Приложении D. Кодруемая этим методом первая полоса соответствует первым четырем рядам коэффициентов поддиапазона в блоке кодирования или такому количеству рядов, которое имеется.

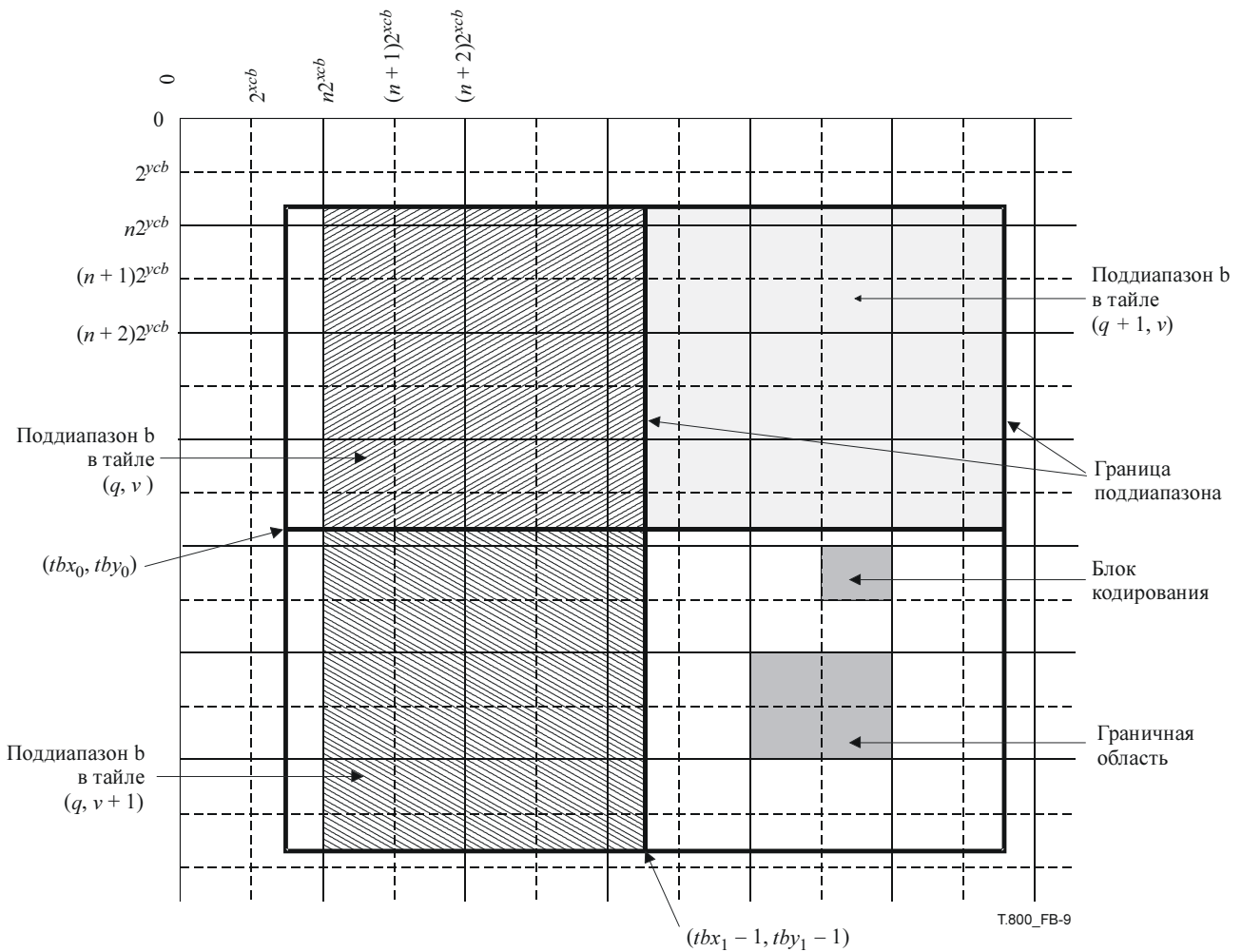


Рисунок В.9 – Блоки кодирования и граничные области в поддиапазоне b из четырех различных тайлов

В.8 Слои

Данные сжатого изображения каждого блока кодирования распределяют в одном или более слоев кодированного потока. Каждый слой состоит из нескольких последовательных кодирующих проходов битовых матриц от каждого блока кодирования в тайле, включая все поддиапазоны всех компонентов этого тайла. Число кодирующих проходов в слое варьируется от одного блока кодирования к другому и может быть нулем для любого или даже всех блоков. О числе слоев для тайла сообщается в сегменте маркера COD (см. А.6.1).

Для данного блока кодирования первый кодирующий проход (если это имеет место) в слое n выполняется сразу после кодирующего прохода блока кодирования в слое $n - 1$ (если это имеет место).

ПРИМЕЧАНИЕ 1. – Каждый слой последовательно и монотонно улучшает качество изображения.

Слои индексируются от 0 до $L - 1$, где L – номер слоя в каждом компоненте тайла.

ПРИМЕЧАНИЕ 2. – На рисунке В.10 показан пример девяти граничных областей с уровнем разрешения m . В таблице В.2 показано формирование слоя.

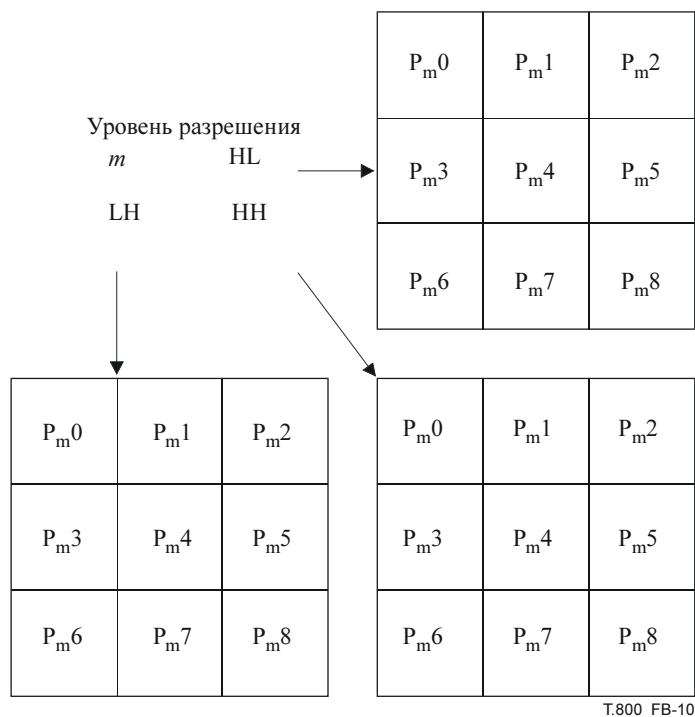


Рисунок В.10 – Диаграмма граничных областей с одним уровнем разрешения для одного компонента

Таблица В.2 – Пример формирования слоя (показан только один компонент)

Уровень разрешения	0			...	m				...	N_L		
	P_{m0}	P_{m1}	P_{m2}	...	P_{m0}	P_{m1}	...	P_{m8}	...	P_{Nl0}	P_{Nl1}	...
Область	Пакет 0	Пакет 0	Пакет 0	Пакет 0	...	Пакет 0	...	Пакет 0	Пакет 0	...
Слой 0	Пакет 1	Пакет 1	Пакет 1	Пакет 1	...	Пакет 1	...	Пакет 1	Пакет 1	...
Слой 1

Основными строительными блоками слоев являются пакеты. Пакеты создают из блоков кодирования данных сжатого изображения от граничных областей с разными уровнями разрешения (для данного компонента тайла).

В.9 Пакеты

Все данные сжатого изображения, представляющие отдельный тайл, слой, компонент, уровень разрешения и граничную область, появляются в кодированном потоке в виде смежных сегментов, которые называются пакетами. Данные пакета выравнивают по 8-битовым (однобайтовым) границам.

Как определено в F.3.1, уровень разрешения $r = 0$ содержит коэффициенты из диапазона $N_L LL$, где N_L – число уровней декомпозиции. Каждый последующий уровень разрешения, $r > 0$, содержит коэффициенты из поддиапазонов nHL , nLH и nHH , как это описано в Приложении F, где $n = N_L - r + 1$. На один компонент тайла с N_L уровнями декомпозиции приходится $N_L + 1$ уровней разрешения.

Данные сжатого изображения в пакете располагают таким образом, чтобы вклады от поддиапазонов LL , HL , LH и HH появлялись в том же порядке. Этот порядок для поддиапазонов идентичен порядку, описанному в F.3.1. Внутри каждого поддиапазона вклады блоков кодирования появляются в растровом порядке, заключенным в границы, которые установлены для соответствующих областей. Уровень разрешения $r = 0$ содержит только диапазон $N_L LL$, а уровни разрешения $r > 0$ – только диапазоны HL , LH и HH . В пакете представлены только те блоки кодирования, которые содержат образцы из соответствующих поддиапазонов, принадлежащих к граничной области.

ПРИМЕЧАНИЕ 1. – На рисунке В.11 показана организация блоков кодирования, из которых формируется пакет, внутри граничной области. В таблице В.3 показан пример кодирующих проходов блока, которые формируют пакет. В таблице В.3 переменные a , b и c – кодирующие проходы блока, где a = проход значимости воспроизведения, b = проход настройки величины, a c = проход очистки (см. Приложение D).

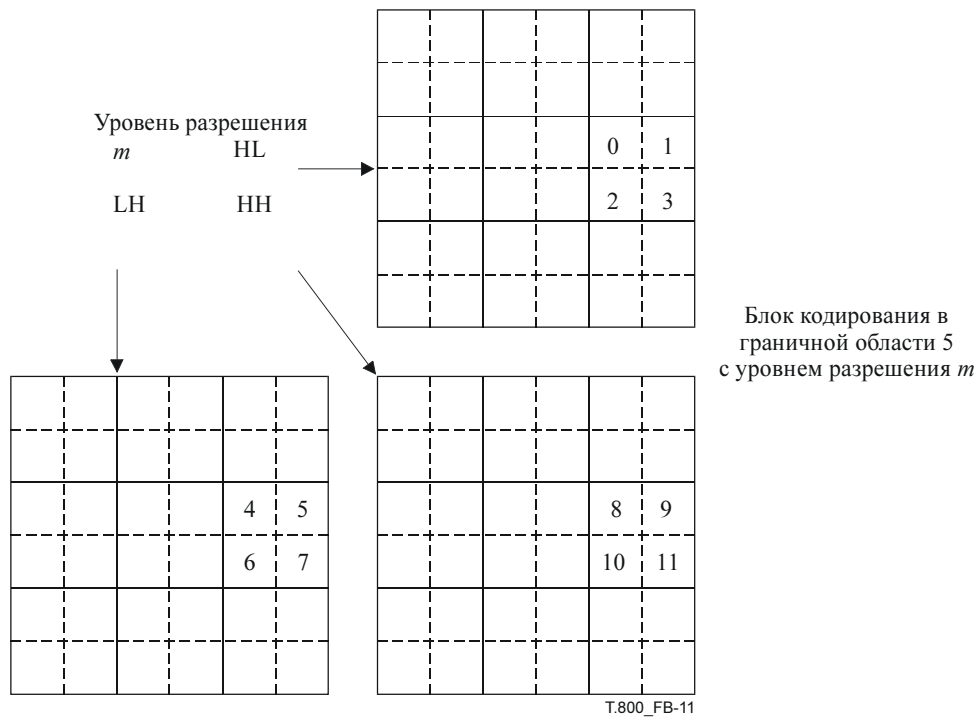


Рисунок В.11 – Диаграмма блоков кодирования внутри граничной области с одним уровнем разрешения

Таблица В.3 – Пример формирования пакета

	блок кодирования 0	блок кодирования 1	блок кодирования 2	...	блок кодирования 10	блок кодирования 11	
MSB	c	0	0	...	c	0	пакет 0
	a	0	0	...	a	0	
	b	0	0	...	b	0	
	c	c	0	...	c	0	
...	a	a	0	...	a	0	пакет 1
	b	b	0	...	b	0	
	c	c	c	...	c	c	
LSB	и т. д.
	a	a	a	...	a	a	
	b	b	b	...	b	b	
	c	c	c	...	c	c	

Данные пакета вводят заголовком пакета (синтаксис описан в В.10), за которым следует тело пакета, содержащее действительные байты кодирования, поступающие от каждого из соответствующих блоков кодирования. Вышеприведенный порядок соблюдается и далее при конструировании заголовка и тела пакета.

Как указано в В.6, возможен случай, когда в граничной области нет блоков кодирования от каких-либо поддиапазонов с некоторыми уровнями разрешения. Тогда все пакеты, соответствующие такой граничной области должны присутствовать в кодирующем потоке как пустые пакеты в соответствии с заголовком пакета, который описан в В.10.

ПРИМЕЧАНИЕ 2. – Даже если граничная область содержит соответствующие блоки кодирования, кодировщик имеет выбор включать ли некодированные проходы в любой соответствующий пакет заданного слоя. В этом случае в кодированном потоке все же должен присутствовать пустой пакет.

В.10 Информация о кодировании в заголовке пакета

Пакеты содержат заголовки со следующей информацией:

- пакеты с нулевой длиной;
- включение блока кодирования;
- информацию о нулевой битовой матрице;
- число кодирующих проходов;
- длина блока кодирования с данными сжатого изображения из заданного блока кодирования.

Два пункта в заголовке кодируют по схеме, которая описана ниже и называется деревом меток. Биты заголовка пакета упаковывают в целый ряд байтов с обычными битами стаффинга (вставки), которые описаны в В.10.1.

Заголовки пакетов находятся в кодированном потоке непосредственно перед данными пакета, за исключением пакетов, в которых использованы сегменты маркеров PPM или PPT. Если используют сегменты маркера PPM, все заголовки пакетов перемещают в основной заголовок (см. А.7.4). Если маркер PPM не используют, то может быть использован сегмент маркера PPT. В этом случае все заголовки пакетов этого тайла перемещают в заголовки частей тайла (см. А.7.5).

В.10.1 Процедура стаффинга битов

Биты упаковывают в байты в порядке от наиболее значимого, MSB, до наименее значимого, LSB. Как только сформирован законченный байт, его присоединяют к заголовку пакета. Если значение байта 0xFF, следующий байт включает дополнительный нулевой бит, вставленный как MSB. Как только сформированы все биты заголовка пакета, на границе байтов последний байт пакуют и передают. Последний байт в заголовке пакета не должен иметь значение 0xFF (потому должен быть включен простой нулевой бит, который вставляют после байта со значением 0xFF, даже если байт 0xFF при других обстоятельствах был бы последним).

В.10.2 Дерево меток

Дерево меток является способом представления двухразмерного массива целых неотрицательных чисел иерархическим способом. Этот способ последовательно создает сокращенные уровни разрешения двухразмерного массива, формируя дерево. В каждом узле дерева записывают минимальное целое одного из узлов (до четырех). На рисунке В.12 показан пример такого представления. Обозначение $q_i(m, n)$ – это значение в узле, который является m -ым от левого и n -ым от верху в i -ом уровне. Уровень 0 является наименьшим уровнем дерева меток. В него входит верхний узел.

1 $q_3(0, 0)$	3 $q_3(1, 0)$	2 $q_3(2, 0)$	3	2	3
2	2	1	4	3	2
2	2	2	2	1	2

а) Исходный массив чисел, уровень 3

1 $q_2(0, 0)$	1 $q_2(1, 0)$	2
2	2	1

б) Минимум из четырех (или менее) узлов, уровень 2

1 $q_1(0, 0)$	1
------------------	---

с) Минимум из четырех (или менее) узлов, уровень 1

1 $q_0(0, 0)$

д) Минимум из четырех (или менее) узлов, уровень 0

T.800_FB-12

Рисунок В.12 – Пример представления дерева меток

Для кодирования элементы массива располагают в растровом порядке. Кодирование – это ответ на ряд вопросов. Каждый узел привязан к текущему значению, которое инициализируют до нуля (минимально). Бит 0 в дереве меток означает, что минимум (или значение в случае наивысшего уровня) больше, чем текущее значение, а бит 1 означает, что минимум (или значение в случае наивысшего уровня) равен текущему значению. Текущее значение увеличивается на единицу для каждого последующего бита 0 в дереве меток. Узлы высших уровней нельзя кодировать до тех пор, пока значения узлов низших уровней остаются постоянными (то есть кодируются битом 1). Верхний по расположению узел уровня 0 (самый низкий уровень) опрашивается первым. Затем опрашивают соответствующий узел уровня 1 и т. д.

В текущем пункте заголовка пакета хранится информация, необходимая только для текущего блока кодирования. Когда накапливается достаточно информации, биты декодирования перестают поступать. Иерархическая природа дерева меток также означает, что в то время как кодируются соседние блоки или слои, формируются ответы на многие вопросы. Эту информацию повторно не кодируют, и поэтому существует ее причинная связь с заголовками пакетов.

ПРИМЕЧАНИЕ. – В качестве примера на рисунке В.12 показано, что кодирование числа в точке $q_3(0, 0)$ должно быть 01111. Два бита 01 означают, что верхний узел в точке $q_0(0, 0)$ больше нуля и в действительности является единицей. Третий бит 1 означает, что узел в точке $q_1(0, 0)$ – также единица. Четвертый бит 1 означает, что узел в точке $q_2(0, 0)$ – также единица. Наконец, последний бит 1 означает, что целевой узел в точке $q_3(0, 0)$ – также единица. Для декодирования следующего узла $q_3(1, 0)$ уже известны узлы в точках $q_0(0, 0)$, $q_1(0, 0)$ и $q_2(0, 0)$. Таким образом, биты кодирования – это 001, а ноль указывает, что узел в точке $q_3(1, 0)$ больше 1, второй ноль указывает, что он больше 2, а 1 означает, что это значение равно 3. Теперь, когда известны $q_3(0, 0)$ и $q_3(1, 0)$, биты кодирования для $q_3(2, 0)$ составят 101. Первый знак 1 показывает параметр $q_2(1, 0)$. Следующие знаки 01 показывают параметр $q_3(2, 0)$. Этот процесс продолжается для всего массива, изображенного на рисунке В.12а.

В.10.3 Пакет нулевой длины

Первый бит в заголовке пакета показывает, не является ли длина пакета нулевой (то есть пустой пакет). Значение 0 указывает на нулевую длину, и в этом случае блоки кодирования отсутствуют. Значение 1 указывает на ненулевую длину, и этот случай будет рассмотрен ниже.

ПРИМЕЧАНИЕ. – Если пакет отмечен как пустой, никакие блоки кодирования не вносят вклад в соответствующий слой. Если следующий пакет не отмечен как пустой, то в него должна быть включена информация блока кодирования (определенная в В.10.4) для предыдущего слоя (с битом пустого пакета). Информацию включения блока кодирования для блоков, которые еще не были включены ни в один пакет, кодируют с использованием дерева меток. Входы этого дерева инициализируют номером слоя, начиная от первого и до того, к которому относится данный блок кодирования. Таким образом, дерево меток получит дополнительную информацию о том, вносит или не вносит вклад блок кодирования в текущий слой и в слой, в котором пакет был отмечен как пустой.

В.10.4 Включение блока кодирования

Информация о том, включены ли в пакет данные сжатого изображения от каждого блока кодирования, сообщается одним из двух способов в зависимости от того, был ли этот же блок уже включен в предыдущий пакет (то есть находится ли он в предыдущем слое).

Для представления информации о блоках кодирования, которые уже были включены в предыдущий пакет, используют одиночный бит, причем 1 означает, что блок кодирования уже включен, а 0 – не включен.

Для блоков кодирования, которые не были включены ни в один пакет, эту информацию отмечают с помощью отдельного кода дерева меток для каждой граничной области, в которой находится поддиапазон. Значениями в этом дереве меток являются номера слоев, в которых первоначально заключены текущие блоки кодирования. Хотя в потоке битов находится полная последовательность битов, представляющих включения дерева меток, в заголовке пакета помещают только те биты, которые необходимы для определения, включен ли данный блок кодирования. Если какое-то дерево меток уже известно из предыдущих блоков или предыдущих слоев, оно более не повторяется. Точно так же включают столько деревьев меток, сколько необходимо для определения включения текущего слоя. Если какой-нибудь блок кодирования не включен до формирования следующего слоя, в поток битов включают только неполное дерево меток.

В.10.5 Нулевая информационная битовая матрица

Если блок кодирования включают первый раз, заголовок пакета содержит информацию, идентифицирующую действительное число битовых матриц, которое использовано для представления коэффициентов блока кодирования. Максимальное число битовых матриц, доступное для представления коэффициентов в любом поддиапазоне b , задано величиной M_b , как это определено равенством (Е-2). Однако, в общем, действительное число битовых матриц, которые создаются кодирующими проходами, составляет $M_b - P$, где число пропущенных наиболее значимых битовых матриц P , может меняться от одного блока кодирования к другому. Все эти пропущенные битовые матрицы считаются нулями. Величина P закодирована в заголовке пакета с особым деревом меток для каждой области таким же образом, как кодируют информацию включения блока кодирования.

В.10.6 Количество проходов кодирования

Число проходов кодирования, включенное в пакет от каждого блока кодирования, идентифицировано в заголовке пакета с использованием кодовых слов, показанных в таблице В.4. Эта таблица дает возможность сообщать о количестве проходов кодирования до 164.

Таблица В.4 – Кодовые слова и количества проходов кодирования для блока кодирования

Количество проходов кодирования	Кодовое слово в заголовке пакета
1	0
2	10
3	1100
4	1101
5	1110
6 – 36	1111 0000 0 до 1111 1111 0
37 – 164	1111 11111 0000 000 до 1111 11111 1111 111

ПРИМЕЧАНИЕ. – Поскольку параметр M_b лимитирован по максимуму величиной 37 из-за ограничений, наложенных синтаксисом сегментов маркеров QCD и QCC (см. А.6.4, А.6.5 и равенство (E-4)), кодирующий алгоритм блока кодирования (описанный в Приложении D) не позволяет выполнить более 109 проходов кодирования.

В.10.7 Длина данных сжатого изображения заданного блока кодирования

Заголовок пакета идентифицирует число байтов, которое поступает от каждого блока кодирования. Последовательность байтов, которые действительно поступили от блока кодирования, не должна заканчиваться на 0xFF. Поэтому, если позиция 0xFF появится в конце вклада блока кодирования в какой-нибудь пакет, 0xFF можно безопасно передвинуть в последующий пакет, в котором есть вклады от блоков кодирования, или опустить, если таких пакетов нет. Например, описанный в Приложении D алгоритм вычисления длины кодирующего прохода гарантирует, что кодирующий проход никогда не закончится на 0xFF.

ПРИМЕЧАНИЕ. – В действительности это не является обременяющим требованием, поскольку байты 0xFF всегда правильно синтезированы арифметическим кодером, описанным в Приложении С.

При сигнализации о числе байтов, вносимых блоком кодирования, возможны два случая: вклад блока содержит единственный сегмент кодового слова, или вклад блока содержит многочисленные сегменты кодового слова. Многочисленные сегменты кодового слова появляются в промежутке между проходами кодирования, которые включены в пакет, как показано в таблице D.8 и D.9.

В.10.7.1 Единственный сегмент кодового слова

Сегмент кодового слова – это число байтов, вносимых в пакет блоком кодирования. Длина сегмента кодового слова выражается длиной бинарного числа:

$$bits = Lblock + [\text{добавленное количество проходов кодирования}], \quad (B-19)$$

где $Lblock$ – переменная состояния блока кодирования. В граничной области для каждого блока кодирования используют различные переменные $Lblock$.

Величину $Lblock$ первоначально устанавливают равной трем. Числу байтов, вносимых каждым блоком, предшествует сигнализация битами, которые увеличивают величину $Lblock$ до необходимой. Сигнальный бит нуль указывает, что величина $Lblock$ достаточная. Если за нулем следует k сигнальных битов, величина $Lblock$ возрастает на k . В то время как величина $Lblock$ может только возрастать, число битов, которые используют для сигнализации о длине вклада блока кодирования, может возрастать или убывать в зависимости от числа включенных проходов кодирования.

ПРИМЕЧАНИЕ 1. – Предположим, например, что в следующих один за другим слоях блоки кодирования содержат 6 байтов, 31 байт, 44 байта и 134 байта соответственно. Предположим далее, что число проходов кодирования: 1, 9, 2 и 5. Кодом для каждого будет 0 110 (0 разделительный, а 110 = 6), 0011111 (0 разделительный, $\log_2 9 = 3$ бита для 9 проходов кодирования, 011111 = 31), 11 0 101100 (110 добавит 2 бита к $Lblock$, $\log_2 2 = 1$, 101100 = 44) и 1 0 1000110 (10 добавит 1 бит к $Lblock$, $\log_2 5 = 2$, 1000110 = 134).

ПРИМЕЧАНИЕ 2. – Не существует требования на минимальное число битов, которые используют для сигнализации о длине (действительно любое число).

В.10.7.2 Многочисленные сегменты кодового слова

Положим T – набор индексов законченных проходов кодирования, включенных в пакет за счет блока кодирования, как показано в таблице D.8 и D.9. Если индекс последнего включенного в пакет прохода кодирования не принадлежит множеству T , то его прибавляют к T . Положим $n_1 < \dots < n_k$ – индексы T . О длинах K непрерывно сообщают с использованием механизма, описанного в В.10.7.1. Первая длина – это число байтов от начала вклада блока кодирования до конца прохода кодирования n_1 . Число добавленных проходов кодирования для вычислений по равенству (B-19) – это число проходов в пакете вплоть до n_1 . Вторая длина – это число байтов от конца прохода

кодирования n_1 до конца прохода кодирования n_2 . Число добавленных проходов кодирования для вычислений по равенству (B-19): $n_2 - n_1$. Эту процедуру повторяют для всех K длин.

ПРИМЕЧАНИЕ. – Рассмотрим избирательный обход арифметического кодирования (см. D.6). Предположим, что проходы, включенные в пакет для заданного блока кодирования, являются проходами очистки битовой матрицы номер 4 до прохода значимости воспроизведения битовой матрицы номер 6 (см. табл. D.9). Эти проходы индексируют как {0, 1, 2, 3, 4}, а их длины заданы как {6, 31, 44, 134, 192} соответственно. Тогда $T = \{0, 2, 3, 4\}$, а $K = 4$ длинам, о которых сообщено. Набор длин, о которых сообщается – {6, 75, 134, 192}, а соответствующее число добавленных проходов кодирования – {1, 2, 1, 1}. Действующая последовательность кодовых битов: 11 1110 (длина $Lblock$ увеличена до 8), 0000 0110 ($\log_2 1 = 0$, 8 битов использовано для кодирования длины 6), 0 0100 1011 ($\log_2 2 = 1$, 9 битов использовано для кодирования длины 75), 1000 0110 ($\log_2 1 = 0$, 8 битов использовано для кодирования длины 134) и 1100 0000 ($\log_2 1 = 0$, 8 битов использовано для кодирования длины 192). Отметим, что величина $Lblock$ возросла только при запуске последовательности.

В.10.8 Порядок информации внутри заголовка пакета

Далее показан порядок информации в заголовке пакета для одного пакета из отдельного слоя, одного компонента тайла, одного уровня разрешения и граничной области.

бит для нулевой или ненулевой длины пакета

для каждого поддиапазона (LL или HL, LH и HH)

для всех блоков кодирования в этом поддиапазоне, заключенных в соответствующую граничную область в растровом порядке

биты включения блока кодирования (если ранее не включался, то дерево меток, в противном случае – один бит)

включен ли блок кодирования

первый ли случай для блока кодирования

информация нулевой битовой матрицы

число включенных кодирующих проходов

возрастание индикатора длины блока кодирования ($Lblock$)

для каждого сегмента кодового слова

длина сегмента кодового слова

Заголовок пакета может сразу следовать за маркером EPH, как описано в A.8.2. Маркер EPH может присутствовать независимо от того, содержатся ли в пакете какие-нибудь вклады от блоков кодирования. В случае появления сегментов маркеров PPM или PPT, маркер EPH (если он используется) должен появиться вместе с заголовком пакета.

ПРИМЕЧАНИЕ. – На рисунке В.13 и в таблице В.5 дан небольшой пример конструкции заголовка пакета. На рисунке В.13 показана информация, которая требуется для кодировщика. В частности в "информации включения" показан слой, в котором каждый блок кодирования появляется в пакете впервые. Декодер получит эту информацию с помощью дерева меток в некоторых заголовках пакетов. В таблице В.5 показан (частично) результирующий поток битов, полученный с помощью этой информации.



T.800_FB-13

Рисунок В.13 – Пример информации, поступившей в кодировщик

Таблица В.5 – Пример потока битов заголовка пакета

Поток битов (порядок)	Полученное значение
1	Пакет ненулевой длины
111	Блок кодирования 0, 0 включен первый раз (частичное включение дерева меток)
000111	Блок кодирования 0, 0 незначительный для 3-х битовых матриц
1100	Блок кодирования 0, 0 включены 3 кодирующие прохода
0	Блок кодирования 0, 0 длина индикатора не изменилась
0100	Блок кодирования 0, 0 имеется 4 байта, использовано 4 бита, $3 + (\log_2 3)$
1	Блок кодирования 1, 0 включен первый раз (частичное включение дерева меток)
01	Блок кодирования 1, 0 незначительный для 4-х битовых матриц
10	Блок кодирования 1, 0 включено 2 кодирующие прохода
10	Блок кодирования 1, 0 длина индикатора возросла на 1 бит (с 3 до 4)
00100	Блок кодирования 1, 0 имеется 4 байта, использовано 5 битов, $4 + (\log_2 2)$, (Отметим, что поскольку это – правильный вход, длина кода не минимальная)
0	Блок кодирования 2, 0 еще не включен (частичное включение дерева меток)
0	Блок кодирования 0, 1 еще не включен
0	Блок кодирования 1, 1 еще не включен
	Блок кодирования 2, 1 еще не включен (никаких данных не требуется; уже направлены частным деревом меток для блоков 2, 0)
...	Данные заголовка пакета для других поддиапазонов, данные пакета
Пакет для следующего слоя	
1	Пакет ненулевой длины
1	Блок кодирования 0, 0 включен еще раз
1100	Блок кодирования 0, 0 включены 3 кодирующие прохода
0	Блок кодирования 0, 0 длина индикатора не изменилась
1010	Блок кодирования 0, 0 имеется 10 байтов, использовано $3 + \log_2 (3)$ битов
0	Блок кодирования 1, 0 не включено в этом слое
10	Блок кодирования 2, 0 еще не включен
0	Блок кодирования 0, 1 еще не включен
1	Блок кодирования 1, 1 включен первый раз
1	Блок кодирования 1, 1 незначительный для 3-х битовых матриц
0	Блок кодирования 1, 1 включен 1 кодирующий проход
0	Блок кодирования 1, 1 информационная длина не изменилась
001	Блок кодирования 1, 1 имеется 1 байт, использовано $3 + \log_2 (1)$ бита
1	Блок кодирования 2, 1 включен первый раз
00011	Блок кодирования 2, 1 незначительный для 6-ти битовых матриц
0	Блок кодирования 2, 1 включен 1 кодирующий проход
0	Блок кодирования 2, 1 длина индикатора не изменилась
010	Блок кодирования 2, 1 имеется 2 байта, использовано $3 + \log_2 1$ бита
...	Данные заголовка пакета для других поддиапазонов, данные пакета

В.11 Тайлы и части тайла

Каждый кодированный тайл представляет последовательность пакетов. Правила, управляющие порядком последовательности пакетов тайла, помещаются внутри кодированного потока, как описано в В.12. Тайл может не содержать пакетов, если образцы компонента изображения не размещены в области, которую занимает тайл на опорной сетке.

Любое представление тайла может быть сокращено пропуском одного или более замыкающих пакетов. Кроме того, может быть пропущено любое число целых пакетов (по порядку), а представляющий тайл последний пакет может быть частично сокращен. Это будет отражено в параметре сегмента маркера.

Последовательность пакетов, представляющих любой конкретный тайл, можно разделить на смежные сегменты, известные как части тайла. В части тайла может содержаться любое число пакетов (включая нуль). Каждый тайл должен содержать по крайней мере одну часть тайла. Разделение на части тайла должно происходить на границах пакетов. Поскольку тайлы представляют связанные геометрические области, части тайла могут распределяться в кодированном потоке любым желательным образом при условии, что части тайла от одного и того же тайла будут появляться в порядке, который сохранит первоначальную последовательность. Каждая часть тайла начинается с сегмента маркера (см. А.4.2), содержащего индекс тайла, к которому принадлежит данная часть.

ПРИМЕЧАНИЕ. – До тех пор, пока сохраняется порядок частей тайла, допускается их чередование из разных тайлов. Например, правильный порядок кодированного потока может быть следующим:

- Тайл номер 0, часть тайла номер 0;
- Тайл номер 1, часть тайла номер 0;
- Тайл номер 0, часть тайла номер 1;
- Тайл номер 1, часть тайла номер 1;
- и т. д.

Если разрешены сегменты маркера SOP (о чем сообщено в сегменте маркера COD, см. А.6.1), каждый пакет от любой части тайла может быть присоединен к сегменту маркера SOP (см. А.8.1). Однако, независимо от использования сегмента маркера SOP, показания счетчика N_{sop} будут увеличиваться с каждым пакетом. Если заголовки пакетов перемещают в сегменты маркеров PPM или PPT (см. А.7.4 и А.7.5), то сегмент маркера SOP может занять место непосредственно перед телом пакета в данных части тайла сжатого изображения.

Если требуются маркеры EPH (сообщением в сегменте маркера COD, см. А.6.1), каждый заголовок пакета в каждой части тайла должен быть отмечен в сегменте маркера EPH (см. А.8.2). Если заголовки пакетов перемещены в сегменты маркеров PPM или PPT (см. А.7.4 и А.7.5), то маркеры EPH должны занять место в сегментах маркеров PPM или PPT после заголовков пакетов.

В.12 Порядок последовательности

Для заданной части тайла в пакетах содержатся все данные сжатого изображения: определенный слой, определенный компонент, определенный уровень разрешения и определенная граничная область. Порядок, в котором эти пакеты располагают в кодированном потоке, называется порядком последовательности. Организация упорядочения пакетов может осуществляться по четырем осям: слой, компонент, уровень разрешения и граничная область.

Возможен случай, когда компоненты имеют разное число уровней разрешения. В этом случае первым ($r = 0$) для всех компонентов считается уровень разрешения, который соответствует поддиапазону N_{iLL} . Индексы синхронизированы, начиная с этой позиции.

ПРИМЕЧАНИЕ. – Например, рассмотрим случай с последовательностью: уровень разрешения – позиция – компонент – слой, с двумя компонентами с 7 уровнями разрешения (6 уровней декомпозиции) и с 3 уровнями разрешения (2 уровня декомпозиции) соответственно. Для обоих компонентов $r = 0$ соответствует поддиапазону N_{iLL} . От $r = 0$ до $r = 2$ компоненты будут чередоваться, как показано ниже. От $r = 3$ до $r = 6$ пакеты будут иметь только компонент 0.

В.12.1 Определение порядка последовательности

Для пяти порядков последовательности используют сигнал сегмента маркера COD (см. А.6.1). В любом заголовке части тайла порядок последовательности может быть также пересмотрен сегментом маркера POS (см. А.6.6). Для каждого возможного порядка последовательности ниже описан механизм определения порядка, в который включены пакеты.

В.12.1.1 Последовательность слой – уровень разрешения – компонент – позиция

Последовательность слой – уровень разрешения – компонент – позиция определена как чередование пакетов в следующем порядке:

для каждого $l = 0, \dots, L - 1$

для каждого $r = 0, \dots, N_{max}$

для каждого $i = 0, \dots, Csiz - 1$

для каждого $k = 0, \dots, numprecincts - 1$

пакет для компонента i , уровня разрешения r , слоя l и граничной области k .

Здесь L – число слоев, а N_{max} – максимальное число уровней декомпозиции N_L , использованных в любом компоненте тайла. Последовательность такого типа может быть полезна, если наиболее желательна низкая точность образца, но необходима информация для всех компонентов.

В.12.1.2 Последовательность уровень разрешения – слой – компонент – позиция

Последовательность уровень разрешения – слой – компонент – позиция определена как чередование пакетов в следующем порядке:

для каждого $r = 0, \dots, N_{max}$

для каждого $l = 0, \dots, L - 1$

для каждого $i = 0, \dots, Csiz - 1$

для каждого $k = 0, \dots, numprecincts - 1$

пакет для компонента i , уровня разрешения r , слоя l и граничной области k .

Последовательность такого типа может быть полезна для получения варианта низкого уровня разрешения для всех компонентов изображения.

В.12.1.3 Последовательность уровень – позиция – компонент – слой

Последовательность уровень – позиция – компонент – слой определена как чередование пакетов в следующем порядке:

для каждого $r = 0, \dots, N_{max}$

для каждого $y = ty_0, \dots, ty_1 - 1$,

для каждого $x = tx_0, \dots, tx_l - 1$,

для каждого $i = 0, \dots, Csiz - 1$

если ((y кратно $YRsiz(i) \cdot 2^{PPy(r, i) + N_L(i) - r}$) ИЛИ (($y = ty_0$) И ($try_0 \cdot 2^{N_L(i) - r}$ НЕ кратно $2^{PPy(r, i) + N_L(i) - r}$))

если ((x кратно $XRsiz(i) \cdot 2^{PPx(r, i) + N_L(i) - r}$) ИЛИ (($x = tx_0$) И ($trx_0 \cdot 2^{N_L(i) - r}$ НЕ кратно $2^{PPx(r, i) + N_L(i) - r}$))

для следующей граничной области k , если таковая существует,

для каждого $l = 0, \dots, L - 1$

пакет для компонента i , уровня разрешения r , слоя l и граничной области k .

Учитывая изложенное, k можно получить из следующего равенства:

$$k = \left\lfloor \left\lceil \frac{x}{\frac{XRsiz(i) \cdot 2^{N_L - r}}{2^{PPx(r, i)}}} \right\rceil - \left\lfloor \frac{trx_0}{2^{PPx(r, i)}} \right\rfloor + numprecinctswide(r, i) \cdot \left(\left\lfloor \left\lceil \frac{y}{\frac{YRsiz(i) \cdot 2^{N_L - r}}{2^{PPy(r, i)}}} \right\rceil - \left\lfloor \frac{try_0}{2^{PPy(r, i)}} \right\rfloor \right) \right\rfloor \quad (B-20)$$

Чтобы воспользоваться этой последовательностью, для каждого компонента величины XR_{siz} и YR_{siz} должны быть в степени два. Последовательность такого типа может быть полезна для получения вариантов с низким уровнем разрешения для всех компонентов изображения в определенном месте пространства.

ПРИМЕЧАНИЕ. – Исключительно для упрощения (но не применения) в вышеприведенных формулах сделана итерация переменных x и y . Наличие большинства пар (x, y) , полученных с помощью такого замкнутого цикла, обычно приводит к отсутствию включения пакетов. Более эффективная итерация может быть основана на минимуме размеров разных граничных областей, размещенных на опорной сетке. Это замечание относится также к циклам в двух следующих последовательностях.

В.12.1.4 Последовательность позиция – компонент – уровень разрешения – слой

Последовательность позиция – компонент – уровень разрешения – слой определена как чередование пакетов в следующем порядке:

для каждого $y = ty_0, \dots, ty_1 - 1$,

для каждого $x = tx_0, \dots, tx_1 - 1$,

для каждого $i = 0, \dots, Csiz - 1$

для каждого $r = 0, \dots, N_L$, где N_L – число уровней декомпозиции для компонента i ,

если $((y \text{ кратно } YR_{siz}(i) \cdot 2^{PPy(r, i) + N_L(i) - r}) \text{ ИЛИ } ((y = ty_0) \text{ И } (try_0 \cdot 2^{N_L(i) - r} \text{ НЕ кратно } 2^{PPy(r, i) + N_L(i) - r})))$

если $((x \text{ кратно } XR_{siz}(i) \cdot 2^{PPx(r, i) + N_L(i) - r}) \text{ ИЛИ } ((x = tx_0) \text{ И } (trx_0 \cdot 2^{N_L(i) - r} \text{ НЕ кратно } 2^{PPx(r, i) + N_L(i) - r})))$

для следующей граничной области k , если таковая существует, в последовательности, показанной на рисунке В.8

для каждого $l = 0, \dots, L - 1$

пакет для компонента i , уровня разрешения r , слоя l и граничной области k .

Учитывая вышеизложенное, k можно получить из равенства (В-20). Чтобы воспользоваться этой последовательностью, для каждого компонента величины XR_{siz} и YR_{siz} должны быть в степени два. Последовательность такого типа может быть полезна для получения высокой точности образца для всех компонентов изображения в определенном месте пространства.

В.12.1.5 Последовательность компонент – позиция – уровень разрешения – слой

Последовательность компонент – позиция – уровень разрешения – слой определена как чередование пакетов в следующем порядке:

для каждого $i = 0, \dots, Csiz - 1$

для каждого $y = ty_0, \dots, ty_1 - 1$,

для каждого $x = tx_0, \dots, tx_1 - 1$,

для каждого $r = 0, \dots, N_L$, где N_L – число уровней декомпозиции для компонента i ,

если $((y \text{ кратно } YR_{siz}(i) \cdot 2^{PPy(r, i) + N_L(i) - r}) \text{ ИЛИ } ((y = ty_0) \text{ И } (try_0 \cdot 2^{N_L(i) - r} \text{ НЕ кратно } 2^{PPy(r, i) + N_L(i) - r})))$

если $((x \text{ кратно } XR_{siz}(i) \cdot 2^{PPx(r, i) + N_L(i) - r}) \text{ ИЛИ } ((x = tx_0) \text{ И } (trx_0 \cdot 2^{N_L(i) - r} \text{ НЕ кратно } 2^{PPx(r, i) + N_L(i) - r})))$

для следующей граничной области k , если таковая существует, в последовательности, показанной на рисунке В.8

для каждого $l = 0, \dots, L - 1$

пакет для компонента i , уровня разрешения r , слоя l и граничной области k .

Учитывая изложенное выше, k можно получить из равенства (В-20). Последовательность такого типа может быть полезна для получения высокой точности в определенном месте пространства для определенного компонента изображения.

В.12.2 Значения порядка последовательности

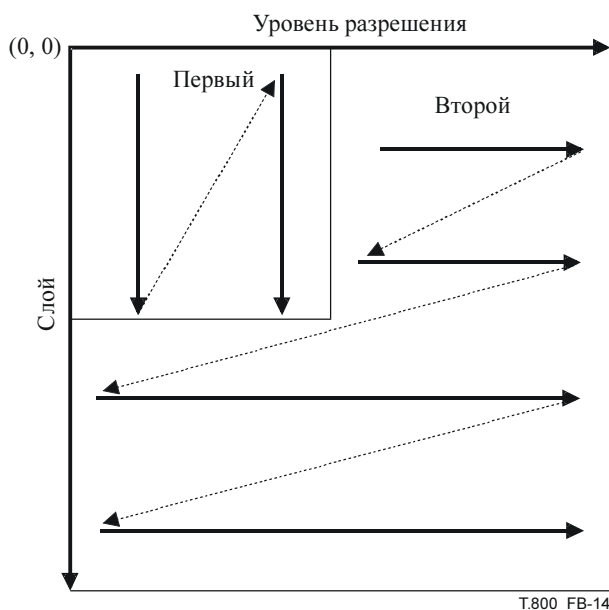
По умолчанию о порядке последовательности сообщается в сегменте маркера COD в основном заголовке или заголовках тайла (см. А.6.1). Циклы всех последовательностей В.12.1 начинаются с нуля и заканчиваются максимальным значением.

Если собираются изменить порядок последовательности, используют маркер РОС (см. А.6.6). В этом случае "циклы", описанные в В.12.1, ограничены пусковыми точками (включая CS_{pod}, RS_{pod}, Слой = 0,) и конечными точками (исключая SE_{pod}, RE_{pod} и LE_{pod}). Так образуют значения порядка последовательности пакетов. Все пакеты, включенные в значения порядка последовательности, находятся в кодированном потоке до тех пор, пока не произойдет следующее изменение порядка последовательности. В кодированном потоке ни один пакет никогда не повторяется. Поэтому для данного компонента тайла, уровня разрешения и граничной области слой всегда начинается со следующего слоя. От декодера требуется определить этот следующий слой.

Таким образом, переменные в указанных выше циклах ограничены значениями порядка последовательности, как это описано соотношениями (В-21).

$$\begin{aligned}
 CS_{pod} \leq i < CE_{pod} \\
 RS_{pod} \leq r < RE_{pod} . \\
 0 \leq l < LE_{pod}
 \end{aligned}
 \tag{B-21}$$

ПРИМЕЧАНИЕ. – На рисунке В.14 показан пример значений двух последовательностей для единственного компонента изображения. Первые пакеты отправляют в последовательности уровень – слой – компонент – позиция до тех пор, пока не будет завершен блок, обозначенный на этом рисунке как "Первый". Затем отправляют пакеты в последовательности слой – уровень разрешения – компонент – позиция для слоев со всеми уровнями разрешения, которые не были отправлены ранее.



T.800_FB-14

Рисунок В.14 – Пример значений порядка последовательности в двух измерениях

В.12.3 Сигнализация об изменении порядка последовательности

Если происходит изменение порядка последовательности, то в кодированном потоке должен быть использован, по крайней мере, один сегмент маркера РОС (см. А.6.6). В данном заголовке может быть только один сегмент маркера РОС, но этот сегмент маркера может описывать множество изменений порядка последовательности.

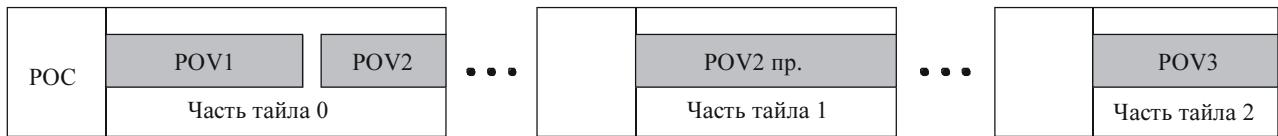
Если в основном заголовке находится сегмент маркера РОС, этот сегмент не учитывает последовательность, которая для всех тайлов записана в маркере COD. Для тех тайлов, в которых нет сегмента маркера РОС в заголовках частей тайла, этот сегмент используют в основном заголовке.

Если для отдельного тайла используют сегмент маркера РОС, то маркер РОС должен находиться в заголовке первой части этого тайла, а обо всех изменениях порядка последовательности должно быть сообщено в заголовках частей этого тайла. Порядок последовательности COD и сегмент маркера РОС (если таковой имеется) игнорируются.

Если об изменениях порядка последовательности сообщается в сегментах маркера РОС (либо в основном заголовке, либо в заголовках частей тайла), то весь порядок всех пакетов в кодированном потоке или затронутые части тайлов в этом потоке должны быть описаны в значениях порядка последовательности в сегментах маркера РОС. Не допускается случай, при котором записаны значения текущего порядка последовательности, а следующий порядок не записан. С другой стороны в сегментах маркера РОС можно описывать больше значений порядка последовательности, чем имеется в кодированном потоке. Кроме того, последнее значение порядка последовательности в каждом тайле может быть незаконченным.

В сегментах маркера РОС должны быть описаны значения порядка последовательности в любом заголовке части тайла перед тем, как будет включен первый пакет. Однако маркер РОС может присутствовать (но не требоваться) в заголовке части тайла непосредственно перед использованным значением порядка последовательности. Это возможно для описания многих значений порядков последовательности в заголовке части тайла, даже если эти значения не появлялись ранее в частях тайла.

ПРИМЕЧАНИЕ. – Например, все значения порядка последовательности могут быть описаны в одном сегменте маркера в заголовке первой части тайла. Этот сценарий показан на рисунке В.15а. В этом случае также допустимо описание двух значений порядков последовательности в заголовке первой части тайла и одного – в третьей, как показано на рисунке В.15б.



а) Все значения порядка последовательности описаны в сегментах маркера РОС в заголовке первой части тайла



T.800_FB-15

б) Значения порядка последовательностей 1 и 2 описаны в сегментах маркера РОС в заголовке первой части тайла, значение порядка последовательности 3 описано в заголовке третьей части тайла

Рисунок В.15 – Пример размещения сегментов маркера РОС

Приложение С

Арифметическое статистическое кодирование

(Это Приложение составляет неотъемлемую часть данной Рекомендации | Международного стандарта)

В настоящем Приложении и во всех его подразделах блок-схемы и таблицы являются нормативными (обязательными) только в том смысле, что они определяют выходные данные, которые при альтернативных применениях должны повторяться.

С.1 Бинарное кодирование (информативное)

На рисунке С.1 представлена элементарная блок-схема бинарного адаптивного арифметического кодировщика. Пару – решение (D) и контекст (CX) – обрабатывают вместе для вывода данных сжатого изображения (CD). Как D, так и CX предусмотрены устройством моделирования (не показано). Контекст CX выбирает вероятностную оценку для использования во время кодирования D. В этой Рекомендации | Международном стандарте CX является меткой для контекста.



Рисунок С.1 – Входы и выходы арифметического кодировщика

С.1.1 Рекурсивное промежуточное разбиение (информативное)

Промежуточное разбиение рекурсивной вероятности кодирования Elias является основой процесса бинарного арифметического кодирования. С каждым бинарным решением текущий вероятностный интервал разделяют на два интервала и кодовую строку модифицируют (если это необходимо) так, что она указывает на нижний предел вероятностного интервала, который присваивают возникающему символу.

В разбиении текущего интервала на два интервала интервал для наиболее вероятного символа (MPS) располагают выше интервала для наименее вероятного символа (LPS). Поэтому, если кодируют MPS, укороченный интервал для LPS добавляют к кодовой строке. Такое условие кодирования требует, чтобы символы различались или как MPS, или как LPS в большей степени, чем 0 или 1. Следовательно, чтобы кодировать это решение, размер интервала LPS и значение MPS для каждого решения должны быть известны.

Поскольку кодовая строка всегда указывает на нижний предел текущего интервала, процесс декодирования является средством определения для каждого решения, какой интервал указан данными сжатого изображения. Это сделано рекурсивно с использованием того же процесса деления на интервалы, что и в кодировщике. Каждый раз, когда решение декодируют, декодер вычитает интервал, добавленный кодировщиком к кодовой строке. Поэтому кодовая строка в декодере является указателем на текущий интервал относительно нижнего предела текущего интервала. Так как процесс кодирования включает в себя сложение двоичных дробей в большей степени, чем объединение целочисленных кодовых слов, более вероятные двоичные решения часто могут быть закодированы ценою гораздо меньшей, чем один бит на решение.

С.1.2 Принципы работы кодирования и аппроксимации (информативное)

Кодирующие операции выполняют с использованием целочисленной арифметики заданной точности и целочисленного представления дробных величин, в которых $0x8000$ эквивалентно десятичному числу $0,75$. Интервал A удерживают в диапазоне $0,75 \leq A < 1,5$ удвоением его всякий раз, когда целочисленное значение падает ниже $0x8000$.

Кодовый регистр C также удваивают каждый раз, когда удваивают A. Периодически – для удержания C от переполнения – байт данных сжатого изображения удаляют из битов высокого порядка регистра C и помещают во внешний буфер данных сжатого изображения. Процедура вставки (стаффинга) битов предотвращает перенос во внешний буфер.

Удержание A в диапазоне $0,75 \leq A < 1,5$ позволяет использовать обычную арифметическую аппроксимацию в интервале разделения. Интервал A и текущая оценка вероятности LPS – это Q_e . Точный подсчет интервалов требует, чтобы:

$$A - (Q_e * A) = \text{интервал для MPS}, \quad (C-1)$$

$$Q_e * A = \text{интервал для LPS}. \quad (C-2)$$

Так как значение A имеет порядок единицы, эти выражения можно аппроксимировать как:

$$A - Q_e = \text{интервал для MPS}, \quad (C-3)$$

$$Q_e = \text{интервал для LPS}. \quad (C-4)$$

Каждый раз, когда кодируют MPS, к кодовому регистру добавляют значение Q_e , а интервал приводят к $A - Q_e$. Когда кодируют LPS, кодовый регистр остается неизменным, а интервал понижается до Q_e . Затем восстанавливают требуемый для A точный диапазон (если это необходимо) перенормировкой как A , так и C .

В проиллюстрированном выше процессе аппроксимации в процедуре разделения интервала интервал для LPS может иногда оказаться больше, чем для MPS. Например, если значение Q_e составляет 0,5, а значение A соответствует минимально допустимой величине 0,75, то приближенный пересчет составляет 1/3 интервала к MPS и 2/3 к LPS. Чтобы избежать такой инверсии размера, интервалы MPS и LPS меняют местами каждый раз, когда интервал LPS больше, чем MPS. Такой условный обмен MPS/LPS может совершаться только в том случае, если существует необходимость перенормировки.

Если происходит перенормировка, запускается процесс вероятностной оценки, который определяет новую вероятностную оценку для текущего кодируемого контекста. Для этой оценки не требуется явного подсчета символов. После кодирования LPS или MPS относительные вероятности перенормировки запускают механизм подсчета приближенного символа, который используют для непосредственной оценки вероятностей.

C.2 Описание арифметического кодировщика (информативное)

На рисунке C.2 ENCODER инициализирует кодировщик процедурой INITENC. Пары CX и D считывают и передают далее в ENCODE до тех пор, пока не будут считаны все пары. В процедуру ENCODE встроены процедуры вероятностной оценки, которые предусматривают адаптивные оценки вероятности для каждого контекста. При необходимости байты данных сжатого изображения являются выходными. Когда все пары CX и D считаны, процедура FLUSH устанавливает в содержимом регистра C столько единичных битов, сколько возможно, а затем выводит конечные данные. Процедура FLUSH также завершает кодирование и формирует требуемый маркер окончания.

ПРИМЕЧАНИЕ. – Хотя FLUSH является необходимой в Рекомендации МСЭ-Т Т.88 | ИСО/МЭК 14492, в данной Рекомендации | Международном стандарте эта функция является информативной. Допустимы и другие методы, например, описанный в D.4.2.

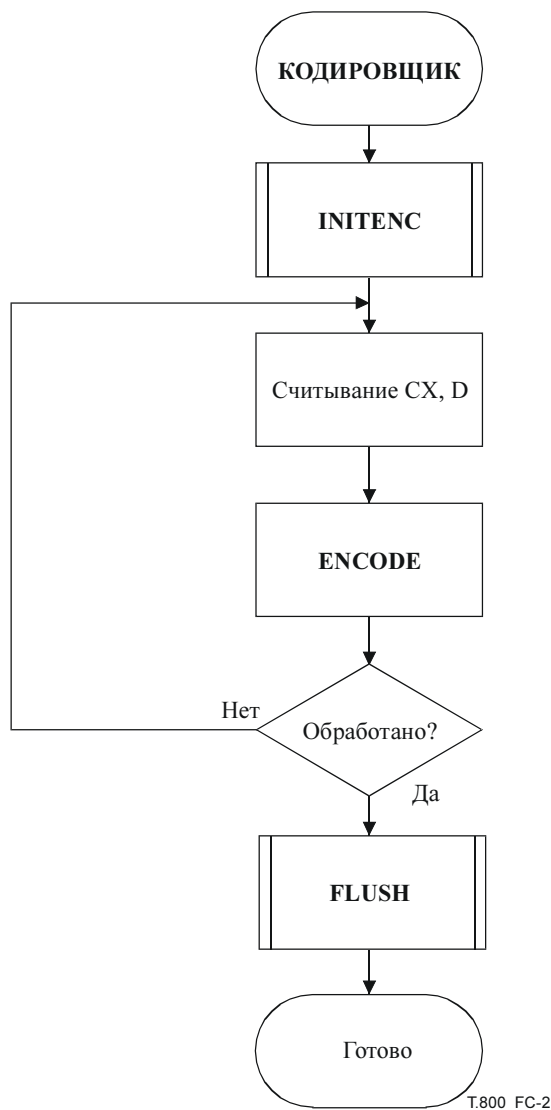


Рисунок С.2 – Кодировщик для кодера MQ

С.2.1 Принципы работы кодового регистра кодировщика (информативное)

Приведенные в этом Приложении блок-схемы допускают следующие структуры регистров для кодировщика:

Таблица С.1 – Структуры регистра кодировщика

	MSB			LSB
Регистр С	0000 cbbb	bbbb bsss	xxxx xxxx	xxxx xxxx
Регистр А	0000 0000	0000 0000	1aaa aaaa	aaaa aaaa

Биты "а" являются дробными в регистре А (значение текущего интервала), а биты "х" – дробными битами в кодовом регистре. Разделительные биты "s" обеспечивают полезные ограничения на перенос, а биты "b" указывают позиции битов, из которых в регистре С удаляют законченные байты данных сжатого изображения. Бит "с" является битом переноса. Более подробное описание стаффинга битов и управление переносом приведено в этом Приложении ниже.

С.2.2 Кодирование решения (информативное)

Процедура ENCODE определяет, является ли решение D равным 0. Далее вызывают соответствующую процедуру CODE0 или CODE1. Зачастую реализации не имеют процедуры ENCODE, а для кодирования решения 0 или 1 непосредственно вызывают процедуры CODE0 или CODE1. Процедура ENCODE представлена на рисунке С.3.

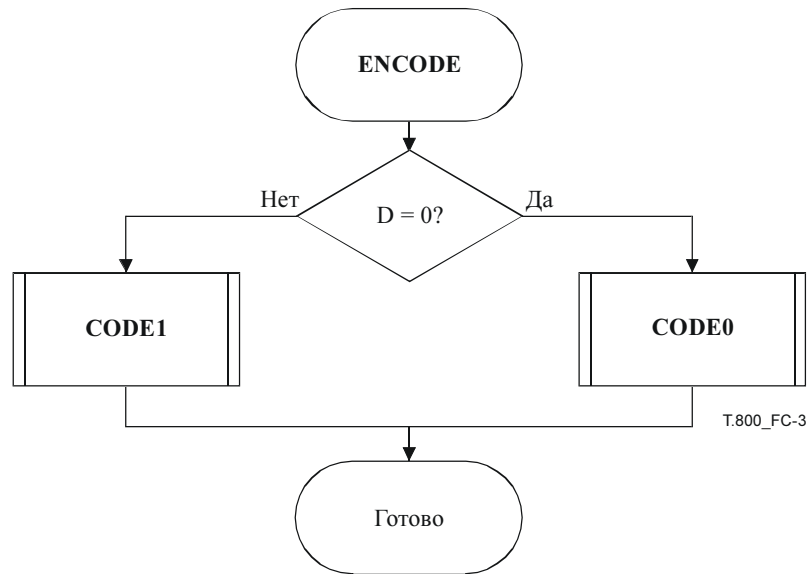


Рисунок С.3 – Процедура ENCODE

С.2.3 Кодирование 1 или 0 (CODE1 и CODE0) (информативное)

Возникающий при кодировании данного бинарного решения символ является либо наиболее вероятным, либо наименее вероятным. Процедуры CODE1 и CODE0 показаны на рисунках С.4 и С.5. На этих рисунках CX является контекстом. Для каждого контекста запоминают значение MPS и индекс вероятностной оценки, который используют в операциях кодирования. $MPS(CX)$ является значением MPS (0 или 1) для контекста CX .

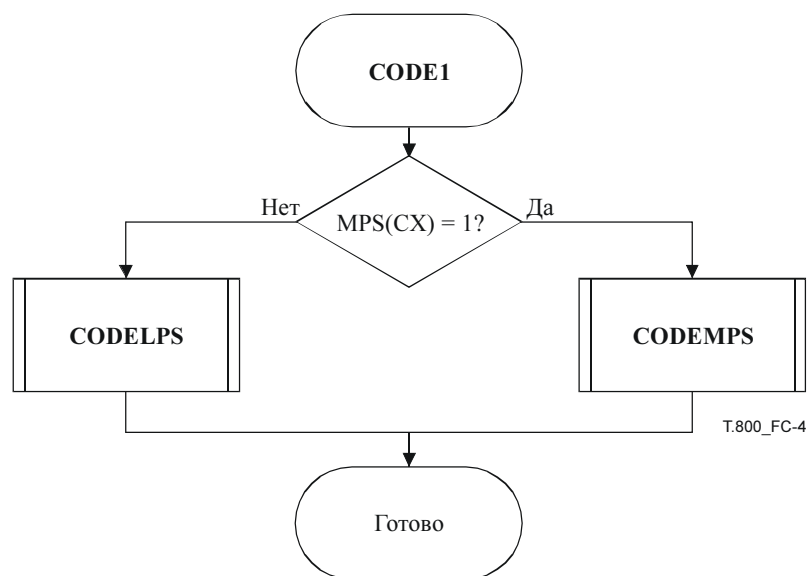


Рисунок С.4 – Процедура CODE1

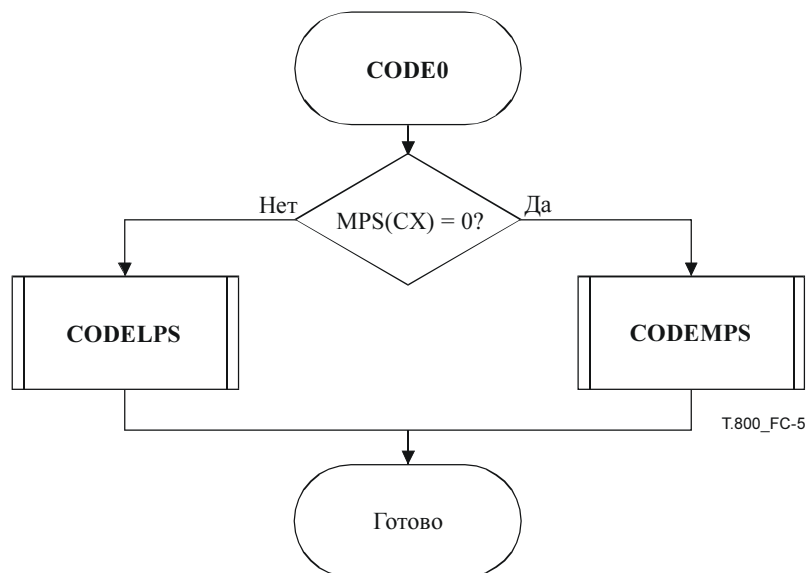


Рисунок С.5 – Процедура CODE0

С.2.4 Кодирование MPS или LPS (CODEMPS и CODELPS) (информативное)

Процедура CODELPS (рис. С.6), как правило, состоит из пересчета интервала к $Q_e(I(CX))$, а вероятностная оценка LPS определена по индексу I , который был сохранен для контекста CX . Верхний интервал является первым подсчитанным, поэтому его можно сравнить с нижним интервалом для подтверждения того, что Q_e имеет меньший размер. Далее всегда следует перенормировка (RENORME). В случае инвертирования размеров интервала происходит условный обмен MPS/LPS и кодируют верхний интервал. В любом случае вероятностная оценка обновляется. Если для индекса $I(CX)$ установлена метка SWITCH, то $MPS(CX)$ инвертируется. Новый индекс I сохраняется в CX как определенный по следующему индексу LPS (NLPS) в столбце таблицы С.2.

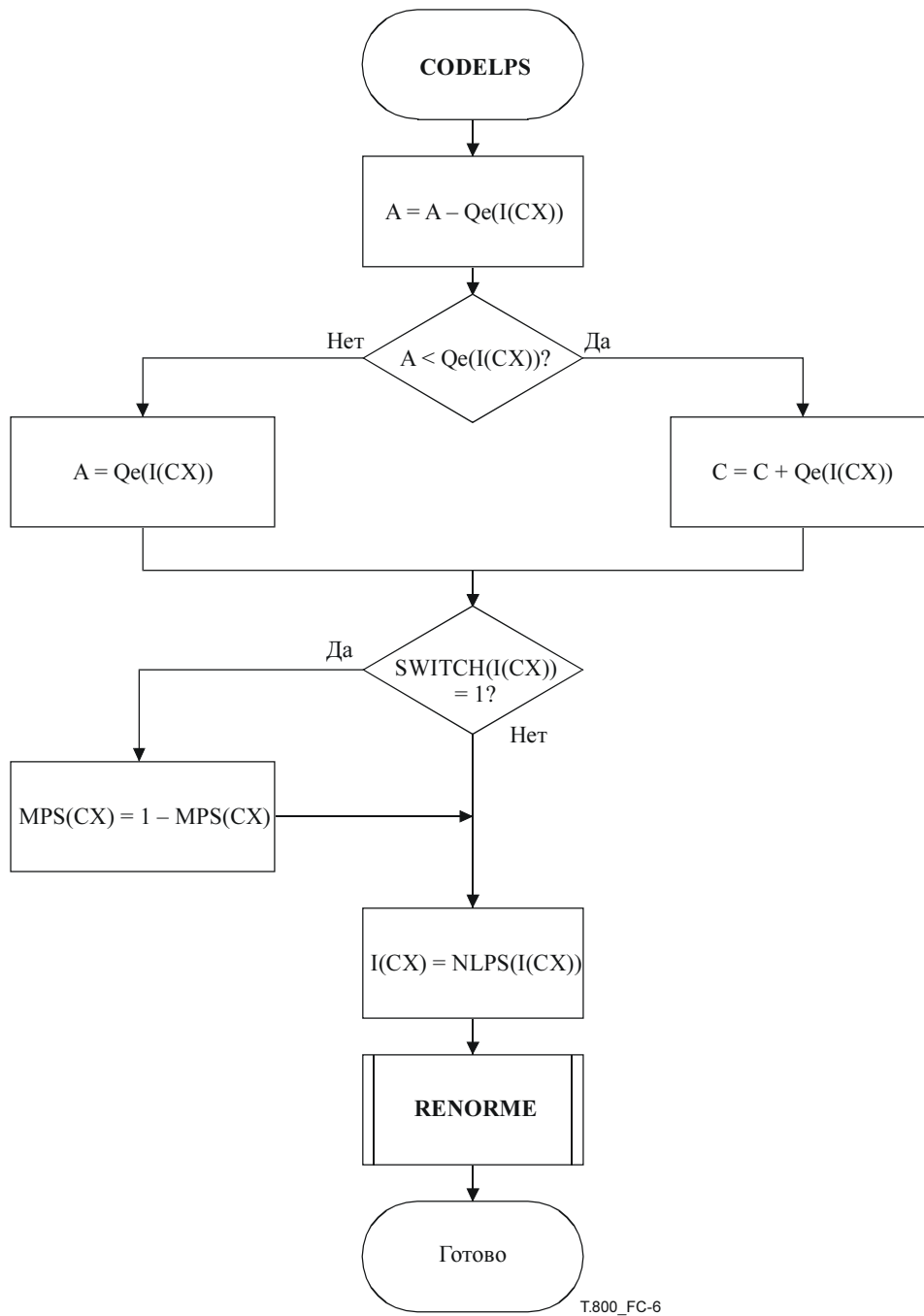


Рисунок С.6 – Процедура CODELPS с условным обменом MPS/LPS

Таблица С.2 – Значения Qe и вероятностная оценка

Индекс	Значения Qe			NMPS	NLPS	SWITCH
	(шестнадцатеричное)	(бинарное)	(десятичное)			
0	0x5601	0101 0110 0000 0001	0,503 937	1	1	1
1	0x3401	0011 0100 0000 0001	0,304 715	2	6	0
2	0x1801	0001 1000 0000 0001	0,140 650	3	9	0
3	0x0AC1	0000 1010 1100 0001	0,063 012	4	12	0
4	0x0521	0000 0101 0010 0001	0,030 053	5	29	0
5	0x0221	0000 0010 0010 0001	0,012 474	38	33	0
6	0x5601	0101 0110 0000 0001	0,503 937	7	6	1

Таблица С.2 – Значения Q_e и вероятностная оценка (продолжение)

Индекс	Значения Q_e			NMPS	NLPS	SWITCH
	(шестнадцатиричное)	(бинарное)	(десятичное)			
7	0x5401	0101 0100 0000 0001	0,492 218	8	14	0
8	0x4801	0100 1000 0000 0001	0,421 904	9	14	0
9	0x3801	0011 1000 0000 0001	0,328 153	10	14	0
10	0x3001	0011 0000 0000 0001	0,281 277	11	17	0
11	0x2401	0010 0100 0000 0001	0,210 964	12	18	0
12	0x1C01	0001 1100 0000 0001	0,164 088	13	20	0
13	0x1601	0001 0110 0000 0001	0,128 931	29	21	0
14	0x5601	0101 0110 0000 0001	0,503 937	15	14	1
15	0x5401	0101 0100 0000 0001	0,492 218	16	14	0
16	0x5101	0101 0001 0000 0001	0,474 640	17	15	0
17	0x4801	0100 1000 0000 0001	0,421 904	18	16	0
18	0x3801	0011 1000 0000 0001	0,328 153	19	17	0
19	0x3401	0011 0100 0000 0001	0,304 715	20	18	0
20	0x3001	0011 0000 0000 0001	0,281 277	21	19	0
21	0x2801	0010 1000 0000 0001	0,234 401	22	19	0
22	0x2401	0010 0100 0000 0001	0,210 964	23	20	0
23	0x2201	0010 0010 0000 0001	0,199 245	24	21	0
24	0x1C01	0001 1100 0000 0001	0,164 088	25	22	0
25	0x1801	0001 1000 0000 0001	0,140 650	26	23	0
26	0x1601	0001 0110 0000 0001	0,128 931	27	24	0
27	0x1401	0001 0100 0000 0001	0,117 212	28	25	0
28	0x1201	0001 0010 0000 0001	0,105 493	29	26	0
29	0x1101	0001 0001 0000 0001	0,099 634	30	27	0
30	0x0AC1	0000 1010 1100 0001	0,063 012	31	28	0
31	0x09C1	0000 1001 1100 0001	0,057 153	32	29	0
32	0x08A1	0000 1000 1010 0001	0,050 561	33	30	0
33	0x0521	0000 0101 0010 0001	0,030 053	34	31	0
34	0x0441	0000 0100 0100 0001	0,024 926	35	32	0
35	0x02A1	0000 0010 1010 0001	0,015 404	36	33	0
36	0x0221	0000 0010 0010 0001	0,012 474	37	34	0
37	0x0141	0000 0001 0100 0001	0,007 347	38	35	0
38	0x0111	0000 0001 0001 0001	0,006 249	39	36	0
39	0x0085	0000 0000 1000 0101	0,003 044	40	37	0
40	0x0049	0000 0000 0100 1001	0,001 671	41	38	0
41	0x0025	0000 0000 0010 0101	0,000 847	42	39	0
42	0x0015	0000 0000 0001 0101	0,000 481	43	40	0
43	0x0009	0000 0000 0000 1001	0,000 206	44	41	0
44	0x0005	0000 0000 0000 0101	0,000 114	45	42	0
45	0x0001	0000 0000 0000 0001	0,000 023	45	43	0
46	0x5601	0101 0110 0000 0001	0,503 937	46	46	0

С.2.5 Вероятностная оценка

В таблице С.2 представлены значения Q_e , которые соответствуют каждому индексу Q_e . Значения Q_e выражены в виде шестнадцатеричных целых чисел, двоичных целых чисел, а также как десятичные дроби. Для преобразования 15-битового целочисленного представления Q_e в десятичную вероятность, значения Q_e делят на $(4/3) * (0x8000)$.

Оценочная функция может быть определена как машина с конечным числом состояний – таблица индексов Q_e и связанные следующие состояния для каждого типа перенормировки (то есть новые табличные позиции), – как показано в таблице С.2. Изменение в состоянии происходит только тогда, когда перенормируют регистр интервала

арифметического кодера. Это выполняется каждый раз после кодирования LPS, а также, если после кодирования MPS регистр интервала меньше, чем 0x8000 (0,75 в десятичном представлении).

После перенормировки LPS NLPS устанавливает новый индекс для вероятностной оценки LPS. Если переключатель установлен на 1, то значение символа MPS реверсируется.

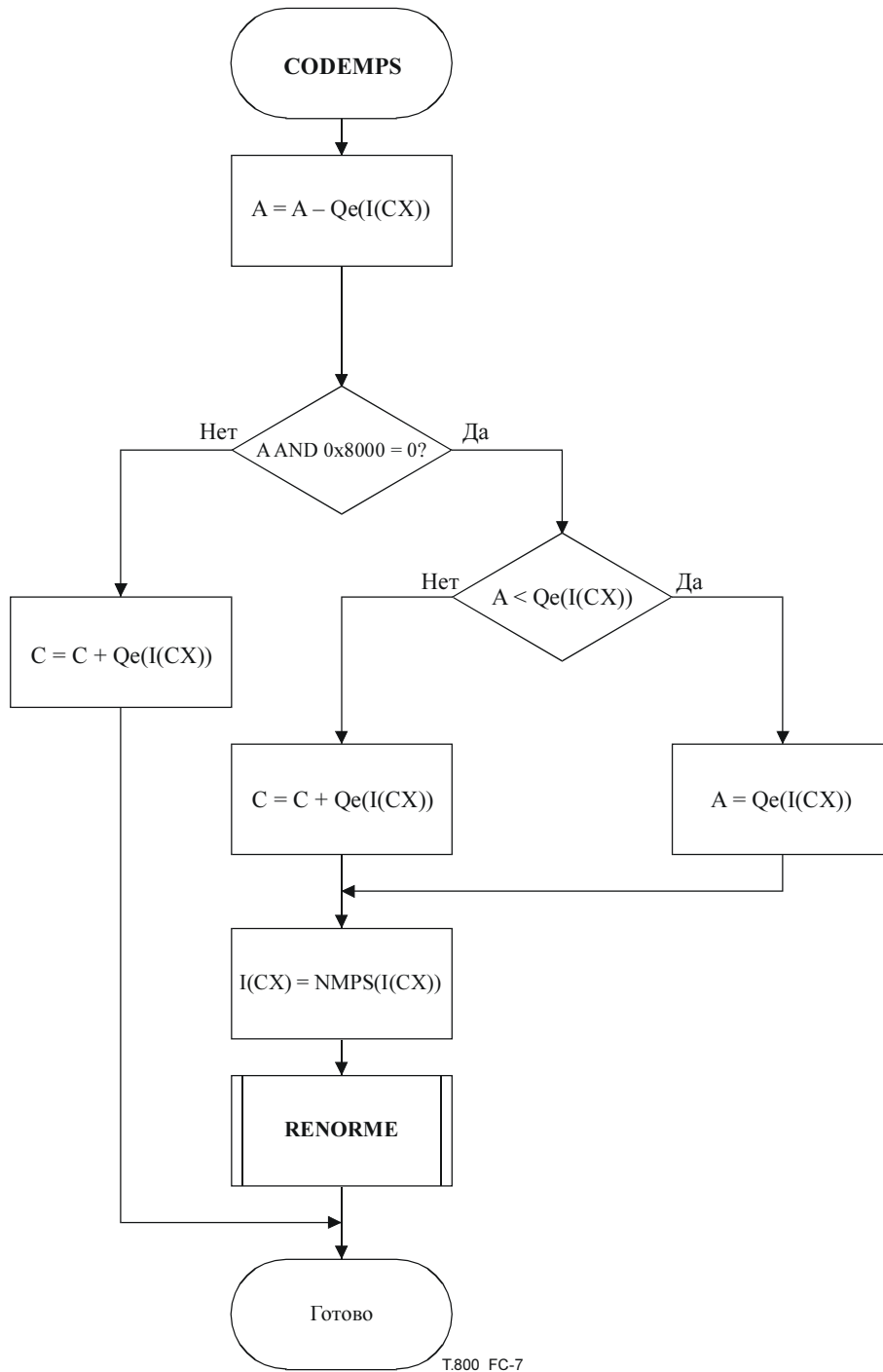


Рисунок С.7 – Процедура CODEMPS с условным обменом MPS/LPS

Индекс к текущей оценке является частью информации, сохраненной для контекста CX. Этот индекс используют как индекс к таблице значений NMPS, который дает следующий индекс для перенормировки MPS. Этот индекс сохраняют в контекстном хранилище в CX. Символ MPS(CX) не изменяется.

Процедура для оценки вероятности в процессе перенормировки LPS аналогична перенормировке MPS, за исключением того, что при SWITCH(I(CX)) равном 1 значение MPS(CX) инвертируется.

Конечное состояние индекса, 46, может быть использовано для установки фиксированной вероятностной оценки – 0,5.

С.2.6 Перенормировка в кодировщике (RENORME) (информативное)

Перенормировки в кодировщике и декодере очень похожи, за исключением того, что кодировщик формирует сжатые биты, а в декодере эти биты используются.

На рисунке С.8 представлена процедура RENORME для перенормировки кодировщика. Регистр интервала А и кодовый регистр С сдвигаются каждый раз на один бит. Количество сдвигов подсчитывают в счетчике СТ. Когда показания счетчика СТ падают до нуля, процедура BYTEOUT удаляет байт данных сжатого изображения из С. Перенормировка продолжается до тех пор, пока А не сделается меньше, чем 0x8000.

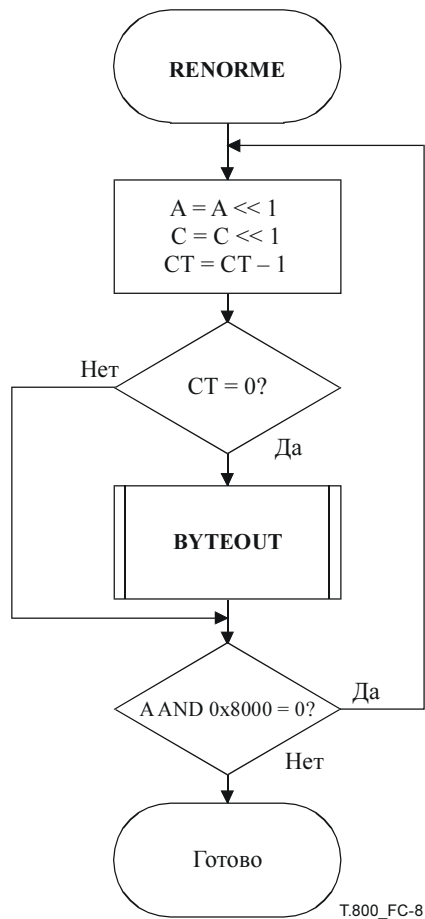


Рисунок С.8 – Процедура перенормировки кодировщика

С.2.7 Вывод данных сжатого изображения (BYTEOUT) (информативное)

На рисунке С.9 представлена подпрограмма BYTEOUT, которую вызывают процедурой RENORME. Эта подпрограмма содержит процедуры стаффинга битов, которые необходимы, чтобы ограничить распространение переноса в завершенные байты данных сжатого изображения. Правила не позволяют передавать содержимое больше, чем одним байтом, который записан в буфер данных сжатого изображения последним по времени.

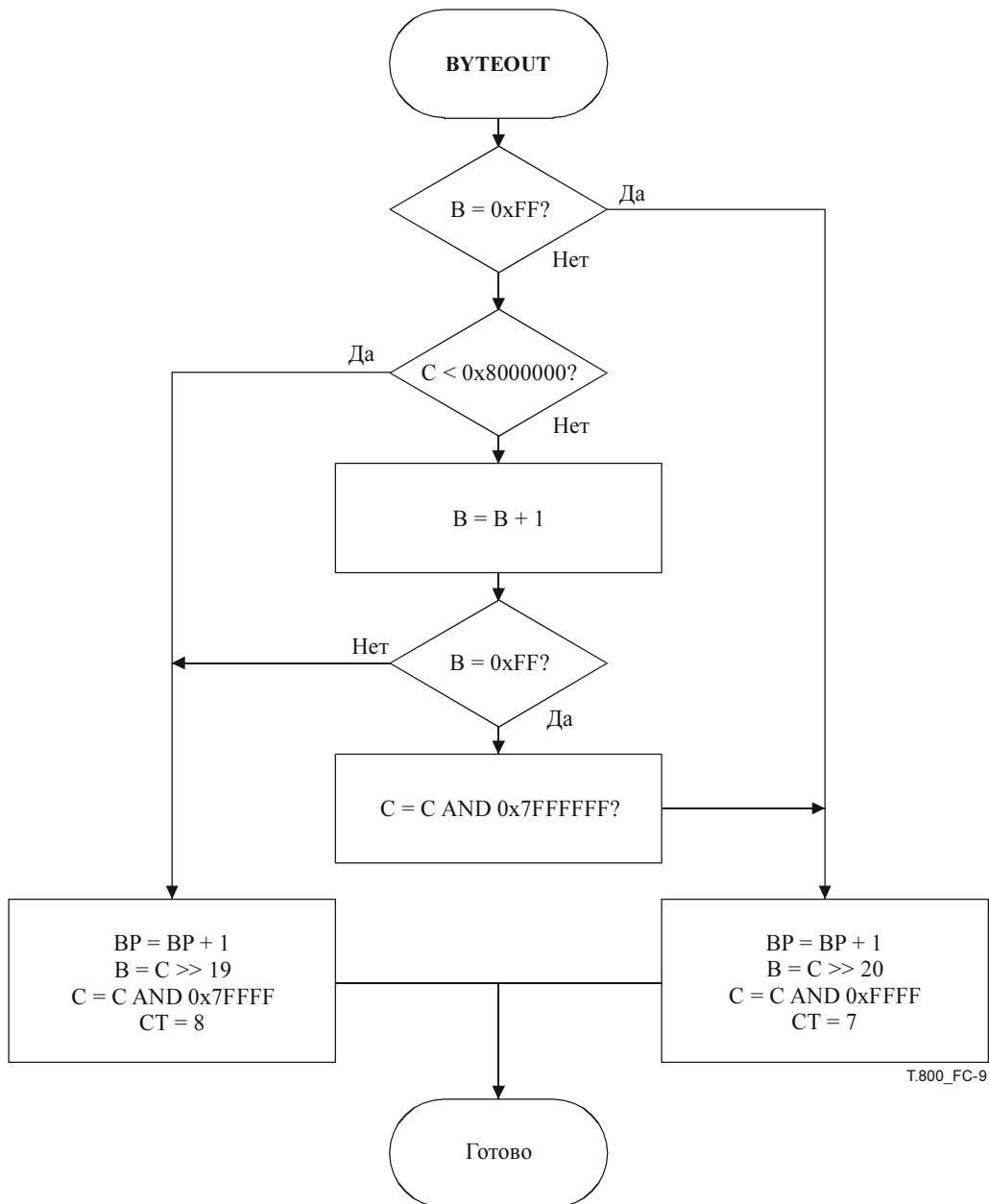


Рисунок С.9 – Процедура BYTEOUT для кодировщика

Процедура в блоке правого нижнего участка выполняет стаффинг бита после байта 0xFF; схожая процедура на левом участке выполняется в тех случаях, когда нет необходимости в стаффинге битов.

Байт В указывает на ВР с помощью буфера данных сжатого изображения. Если значение В не равно 0xFF, то проверяют бит переноса. Если бит переноса установлен, его добавляют к В. Значение В снова проверяют, чтобы выяснить, необходимо ли вставить бит в следующем байте. После определения необходимости стаффинга битов и выбора соответствующего пути ВР увеличивают и новое значение В удаляют из битов "b" кодового регистра.

С.2.8 Инициализация кодировщика (INITENC) (информативное)

Процедуру INITENC используют для запуска арифметического кодера. После инициализации MPS и I выполняют основные шаги, представленные на рисунке С.10.

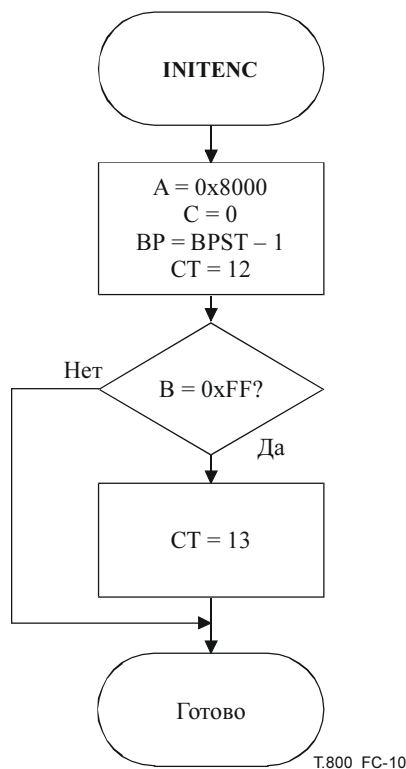


Рисунок С.10 – Инициализация кодировщика

Регистр интервала и кодовый регистр устанавливают в начальные значения. Счетчик битов также устанавливают в первоначальное состояние. Установка $CT = 12$ связана с существованием в регистре трех битов разделителя, которые следует заполнить, прежде чем достигнуто поле, из которого удаляют байты. Указатель BP всегда отмечает байт, предшествующий позиции $BPST$, в которой размещают первый байт. Поэтому, если предшествующий байт равен $0xFF$, произойдет стаффинг неинформативного бита, что однако может быть скомпенсировано увеличением CT . В таблице D.7 представлены исходные установочные параметры для MPS и I .

С.2.9 Завершение кодирования (FLUSH) (информативное)

Процедуру $FLUSH$ используют для того, чтобы завершить операции кодирования и сформировать требуемый маркер окончания (см. рисунок С.11). Эта процедура гарантирует, что префикс $0xFF$ к коду маркера перекроет конечные биты данных сжатого изображения. Это также гарантирует, что любой код маркера в конце данных сжатого изображения будет распознан и интерпретирован перед завершением процесса декодирования.

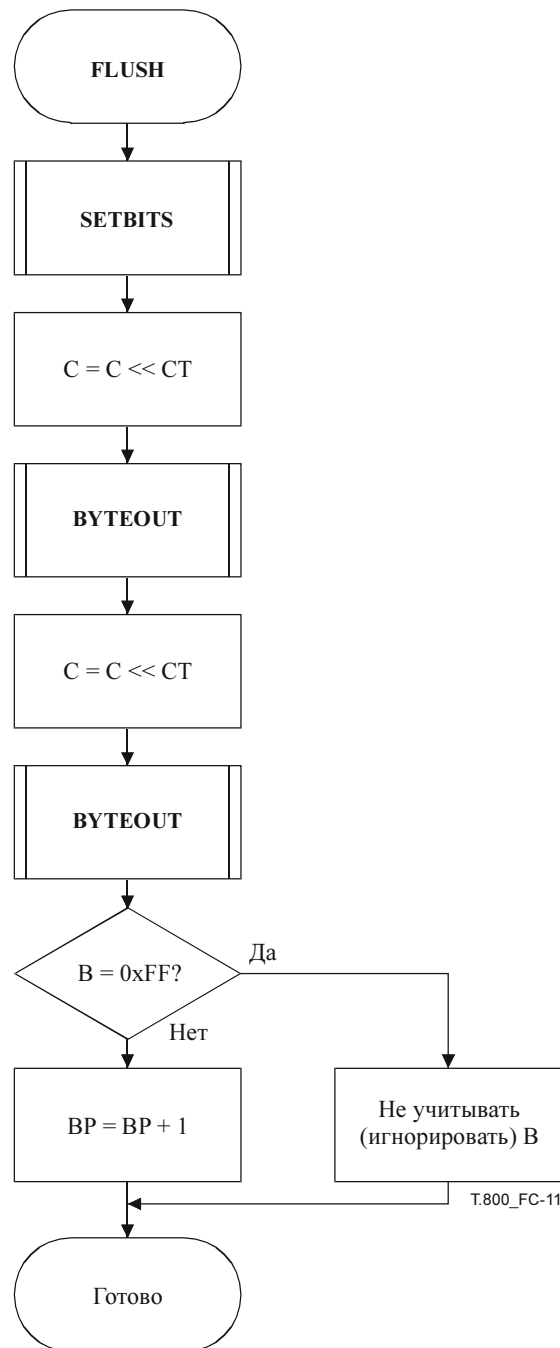


Рисунок С.11 – Процедура FLUSH

Первая часть процедуры FLUSH устанавливает в регистре С на значение 1 столько битов, сколько возможно (см. рисунок С.12). Ограничивающим верхним пределом для регистра С является сумма регистра С и регистра интервала. 16 битов младшего разряда С устанавливают на 1 и результат сравнивают с верхним пределом. Если значение С слишком велико, то первый единичный бит удаляют и приводят С к значению внутри интервала.

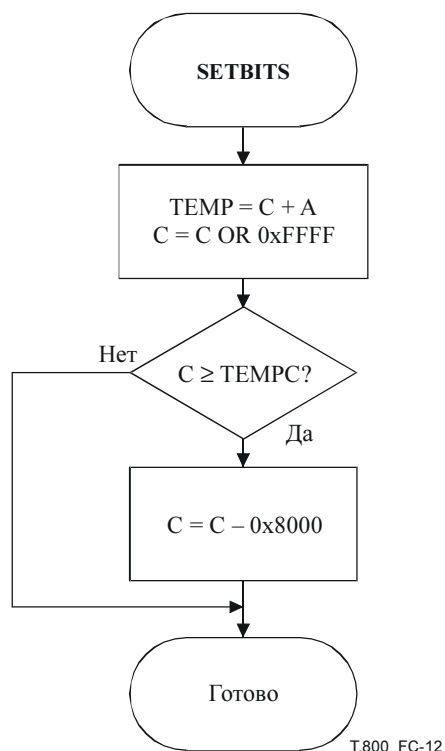


Рисунок С.12 – Установка конечных битов в регистре С

В регистре С байт заканчивают сдвигом С, а затем удаляют два байта. Если байт в буфере В равен 0xFF, то его не учитывают. В противном случае, буфер В является выходным для потока битов.

ПРИМЕЧАНИЕ. – Это единственное нормативное дополнение для завершения в Рекомендации МСЭ-Т Т.88 | ИСО/МЭК 14492. Однако дальнейшее сокращение потока битов допустимо по данной Рекомендации | Международному стандарту при условии, что будет обеспечено правильное декодирование (см. D.4.2).

С.3 Процедура арифметического декодирования

На рисунке С.13 показана элементарная блок-схема бинарного адаптивного арифметического декодера. На вход арифметического декодера подаются данные сжатого изображения CD и контекст CX из устройства моделирования декодера (не показано). Выходом декодера является решение D. Для устройств моделирования кодировщика и декодера необходимо подавать абсолютно одинаковые контексты CX для каждого заданного решения.



Рисунок С.13 – Входы и выходы арифметического декодера

Устройство DECODER инициализирует декодер процедурой INITDEC (рисунок С.14). Контексты CX и байты данных сжатого изображения (если это необходимо) считывают и передают в DECODE до тех пор, пока не будут считаны все контексты. Программа DECODE декодирует бинарное решение D и возвращает значение или 0, или 1. В устройство DECODE встроены процедуры вероятностной оценки, которые предусматривают адаптивные оценки вероятности для каждого контекста. После того как считаны все контексты, восстанавливают данные сжатого изображения.

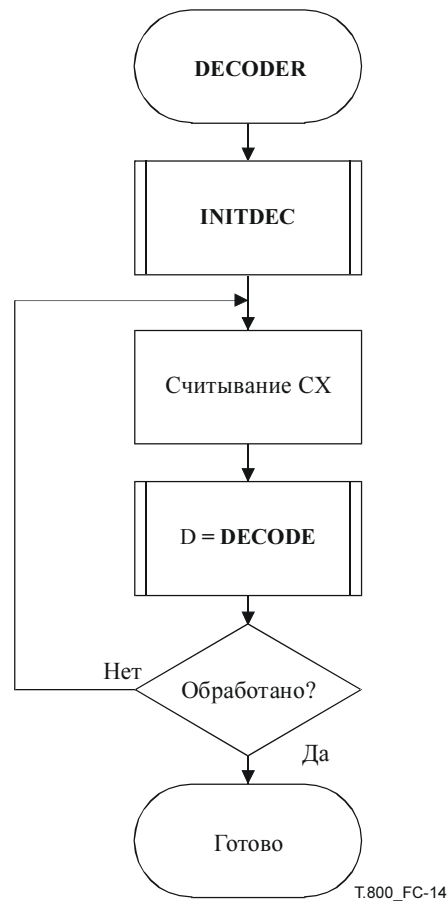


Рисунок С.14 – Декодер для кодера MQ

С.3.1 Принципы работы кодового регистра декодера

Приведенные в этом Приложении блок-схемы допускают следующие структуры регистров для декодера:

Таблица С.3 – Структуры регистра декодера

	MSB	LSB
регистр Chigh	xxxx xxxx	xxxx xxxx
регистр Clow	bbbb bbbb	0000 0000
регистр A	aaaa aaaa	aaaa aaaa

Регистры Chigh и Clow могут быть представлены как один 32-битовый регистр C, в котором перенормировка передвигает бит новых данных с позиции регистра MSB Clow на позицию регистра LSB Chigh. Однако только в регистре Chigh используют сравнения декодирования. Новые данные вставляют в биты "b" регистра Clow по одному байту каждый раз.

Более подробное описание обработки данных с битами вставки будет рассмотрено в этом Приложении далее.

Отметим, что сравнения, представленные в различных процедурах этого параграфа, предполагают точности большие, чем 16 битов. Логические сравнения могут использовать 16-битовую точность.

С.3.2 Декодирование решения (DECODE)

Декодер декодирует одно бинарное решение каждый раз. После декодирования решения декодер вычитает величину из данных сжатого изображения, которая была добавлена кодировщиком. Оставшаяся величина в данных сжатого изображения – это сдвиг от базы текущего интервала к интервалу, размещенному во всех еще не декодированных бинарных решениях. При первом тестировании по процедуре DECODE размер регистра Chigh сравнивают с интервалом LPS (см. рисунок С.15). До тех пор пока не потребуется условный обмен, данное тестирование определяет, какой из символов декодируют: MPS или LPS. Если значение Chigh логически больше или равно вероятностной оценке LPS Q_e для текущего индекса I, сохраненного в CX, то Chigh уменьшится на эту величину. Если A не меньше, чем 0x8000, то значение MPS, сохраненного в CX, используют для установки декодированного решения D.

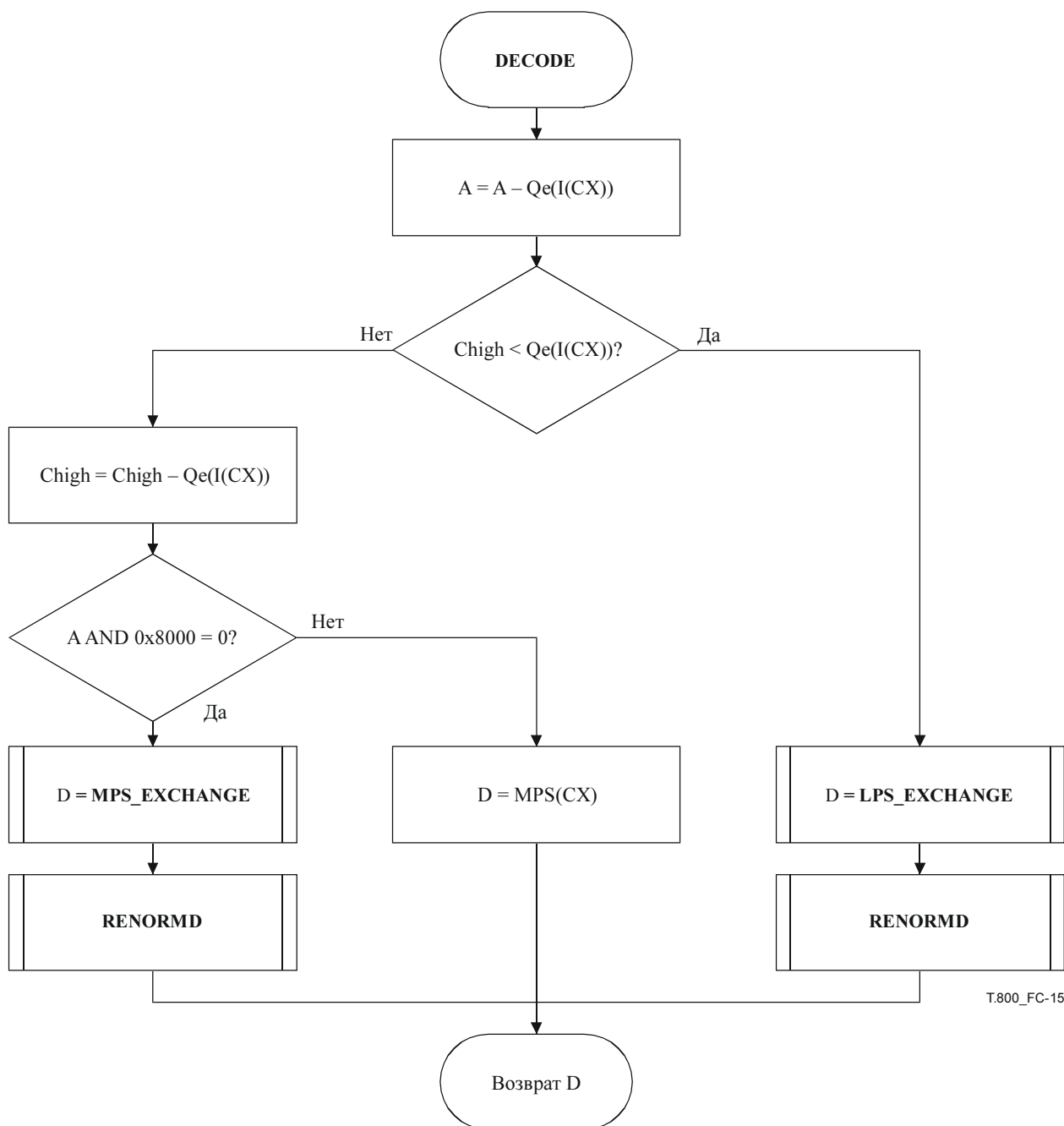


Рисунок С.15 – Декодирование MPS или LPS

Если существует необходимость перенормировки, то может происходить условный обмен MPS/LPS. На рисунке С.16 представлена процедура условного обмена для пути MPS. Пока величина A интервала MPS, рассчитываемая на первом этапе вычислений рисунка С.16, логически не меньше, чем вероятностная оценка $Qe(I(CX))$ для LPS, решение может быть установлено из $MPS(CX)$. В этом случае индекс $I(CX)$ обновляют из следующего индекса MPS (NMPS) в столбце таблицы С.2. Если, однако, интервал LPS больше, происходит условный обмен и появляется индекс LPS. Решение D устанавливают путем инвертирования $MPS(CX)$. Если SWITCH имеет значение "1", вероятностное обновление сдвигает значение MPS и обновляет индекс $I(CX)$ из следующего индекса LPS (NLPS) в столбце таблицы С.2. Необходимо, чтобы вероятностная оценка в декодере была идентична вероятностной оценке в кодировщике.

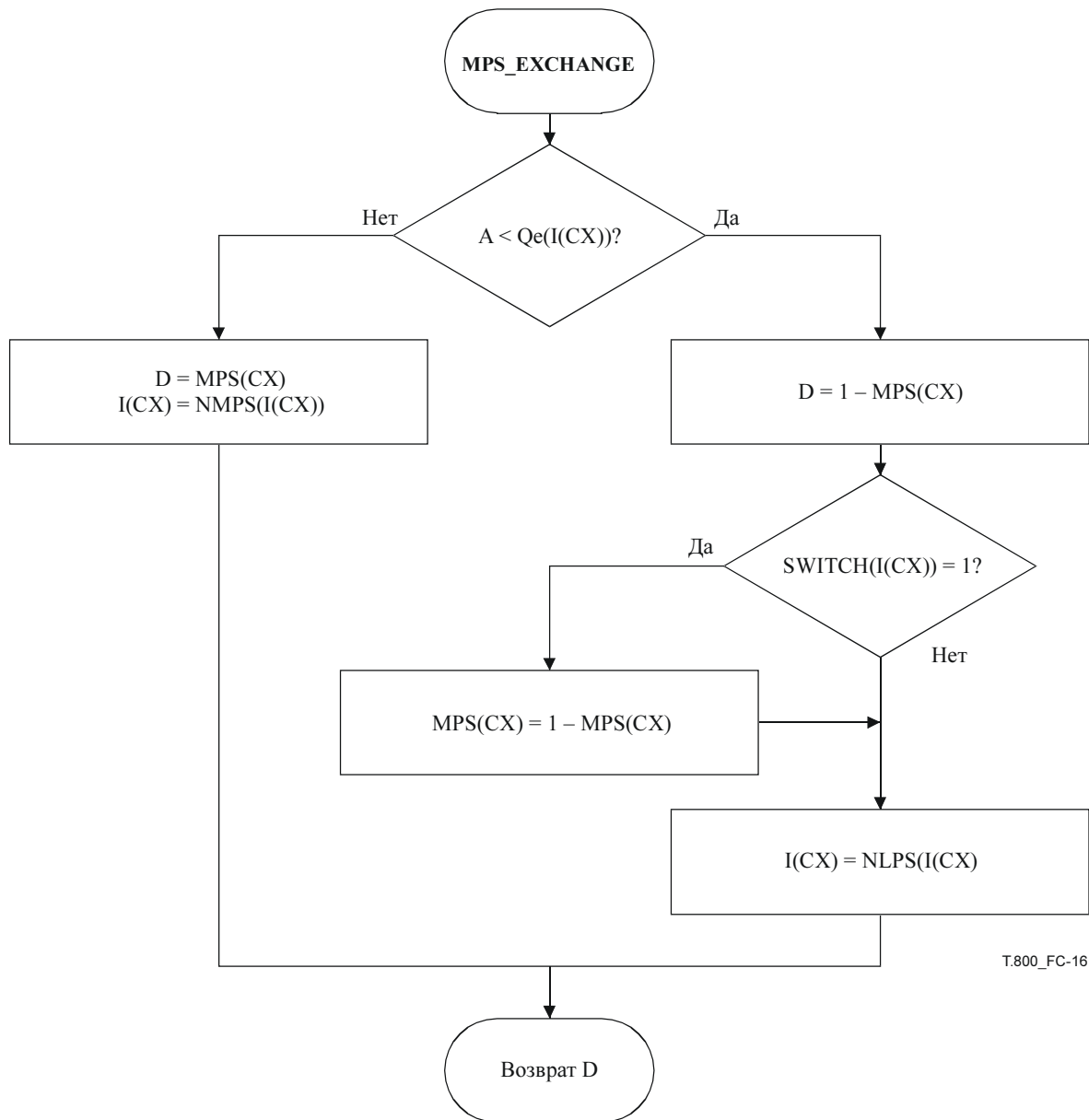
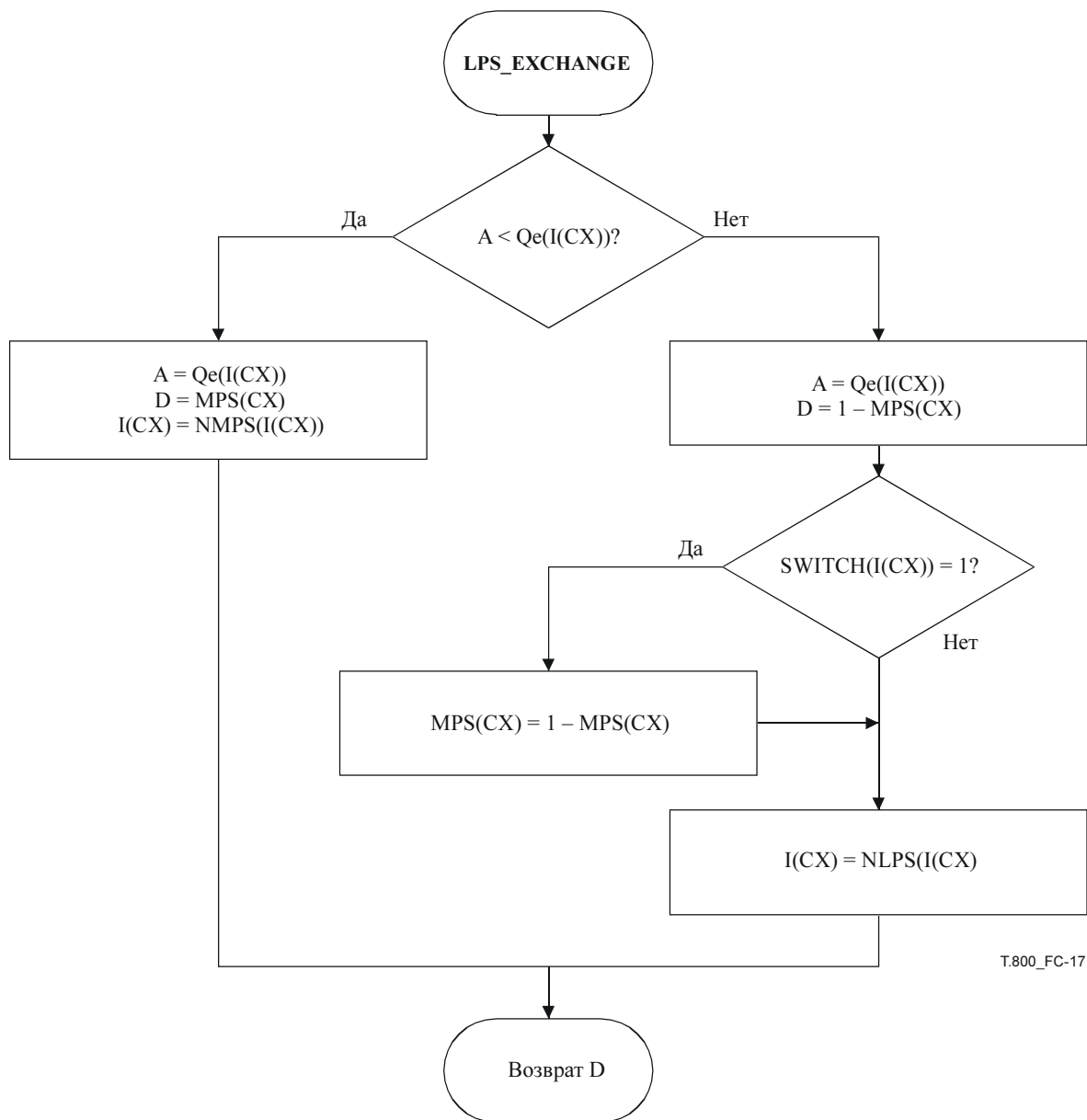


Рисунок С.16 – Процедура условного обмена декодера для пути MPS

На рисунке С.17 представлена процедура условного обмена декодера LPS_EXCHANGE для пути LPS. С помощью того же логического сравнения между величиной A интервала MPS и $Qe(I(CX))$ интервала LPS определяют, произошел ли условный обмен. На обоих путях алгоритма значение A нового интервала устанавливают на $Qe(I(CX))$. По левой ветви пути условный обмен происходит точно также как решение и обновление для случая MPS. По правой ветви пути решение LPS и обновление повторяют предыдущую схему.



T.800_FC-17

Рисунок С.17 – Процедура условного обмена декодера для ветви LPS

С.3.3 Перенормировка в декодере (RENORMD)

На рисунке С.18 представлена процедура RENORMD для перенормировки декодера. Счетчик А отслеживает число сжатых битов в части Slow регистра С. Если СТ равно нулю, в часть Slow вставляют новый байт по процедуре BYTEIN. Регистр С в этой процедуре объединяет регистры Chigh и Slow.

Регистр интервала А и кодовый регистр С сдвигаются на один бит каждый раз до тех пор, пока А не станет меньше, чем 0x8000.

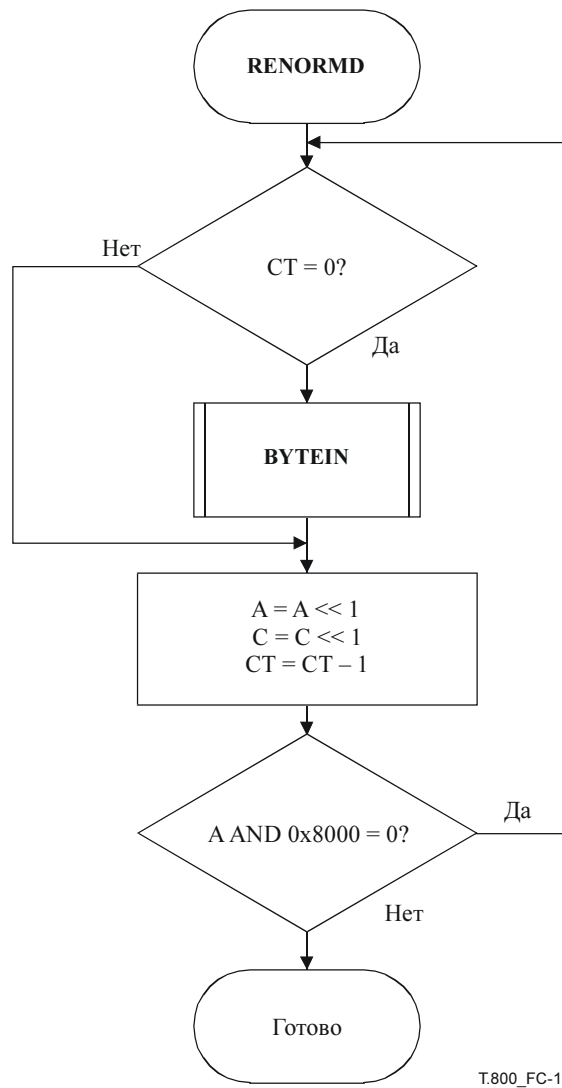


Рисунок С.18 – Процедура перенормировки декодера

С.3.4 Вход данных сжатого изображения (BYTEIN)

На рисунке С.19 представлена процедура BYTEIN, которую вызывают из процедуры RENORMD. Эта процедура считывает один байт данных, предусматривая в этом процессе возможную вставку битов, следующих за байтом 0xFF. Эта процедура также выявляет коды маркера, которые должны присутствовать в конце прохода кодирования. Регистр С в этой процедуре объединяет регистры Chigh и Clow.

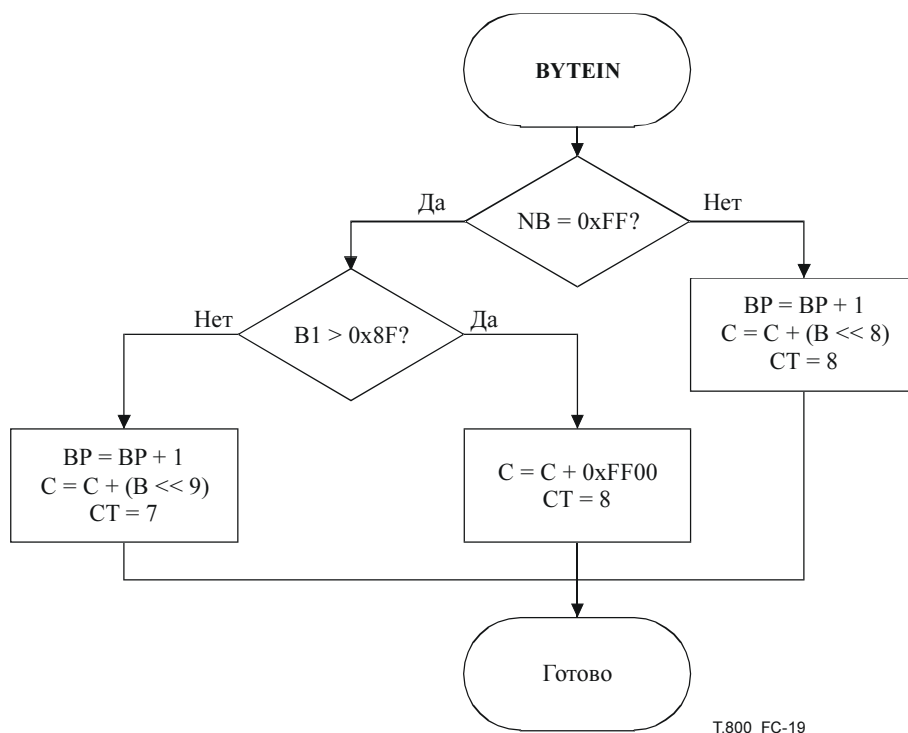


Рисунок С.19 – Процедура BYTEIN для декодера

С помощью буфера данных сжатого изображения байт В указывает на указатель ВР. Если байт В не равен 0xFF, то ВР увеличивается, и новое значение В вставляют в 8 битов высокого порядка регистра Slow.

В том случае, если байт В равен 0xFF, то проверяют В1 (байт, на который указывает ВР+1). Если В1 превышает 0x8F, то В1 должен быть одним из кодов маркера. Код маркера интерпретируют как обязательный, а указатель буфера по-прежнему указывает на префикс 0xFF кода маркера, который арифметически завершает данные сжатого изображения. Единичные биты подают в декодер до тех пор, пока не завершён процесс декодирования. Это показано на рисунке С.19 – добавлением к регистру С значения 0xFF00 и установкой счетчика битов СТ на 8.

Если В1 не является кодом маркера, то значение ВР возрастает, чтобы указать на следующий байт, который содержит бит стаффинга. К регистру С добавляют байт В с таким выравниванием, чтобы бит стаффинга (который содержит какой-нибудь перенос) добавлялся к биту наименьшего порядка регистра Chigh.

С.3.5 Инициализация декодера (INITDEC)

Для запуска арифметического декодера используют процедуру INITDEC. После инициализации MPS и I, выполняют основные шаги, представленные на рисунке С.20.

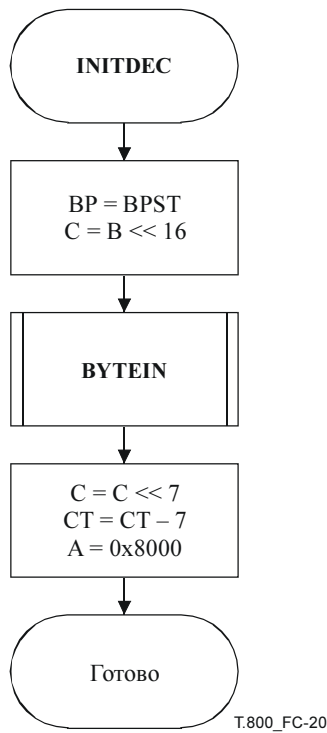


Рисунок С.20 – Инициализация декодера

Указатель данных сжатого изображения BP инициализируют в $BPST$ (указывая на первый сжатый байт). Первый байт данных сжатого изображения сдвигают на байт младшего разряда регистра $Chigh$, а затем считывают новый байт. Регистр C сдвигается на 7 битов, и CT уменьшается на 7, приводя регистр C к выравниванию с начальным значением A . Регистр интервала A устанавливают в положение, совпадающее с исходным значением в кодировщике. В таблице D.7 приведены первоначальные установки для MPS и I .

С.3.6 Переустановка статистических данных арифметического кодирования

В определенные моменты декодирования происходит переустановка всех или некоторых статистических данных арифметического кодирования. Данный процесс включает возврат $I(CX)$ и $MPS(CX)$ к первоначальным значениям для всех или некоторых значений CX (см. таблицу D.7).

С.3.7 Сохранение статистических данных арифметического кодирования

В некоторых случаях декодеру необходимо сохранить или восстановить некоторые значения $I(CX)$ и $MPS(CX)$.

Приложение D

Моделирование битовых коэффициентов

(Это Приложение составляет неотъемлемую часть данной Рекомендации | Международного стандарта)

В этом Приложении и всех его подразделах блок схемы и таблицы являются нормативными (обязательными) только в том смысле, что они определяют выходные данные, которые при альтернативных применениях должны повторяться.

В этом Приложении определено моделирование и сканирование коэффициентов, представленных битами.

Блоки кодирования (см. Приложение В) декодируют в битовые матрицы, начиная с наиболее значимой битовой матрицы с ненулевыми элементами и заканчивая наименее значимой битовой матрицей. Для каждой битовой матрицы блока кодирования используют специальный шаблон сканирования при каждом из трех кодирующих проходов. Каждый битовый коэффициент в битовой матрице появляется только в одном из трех кодирующих проходов, которые называются восстановление значимости, подстройкой величины и проходом очистки. Для каждого прохода создаются контексты, которые поступают на арифметический кодер СХ вместе с потоком битов СД (см. С.3).

D.1 Шаблон сканирования блока кодирования

Каждую битовую матрицу блока кодирования сканируют в особом порядке. Начиная с верхнего левого края, сканируют первые четыре коэффициента в первом столбце, затем первые четыре коэффициента второго столбца и т. д. до правого края блока кодирования. Затем сканирование возвращается к левому краю блока второй строки, и сканируется второй набор по четыре коэффициента из каждого столбца. Этот процесс продолжается до нижнего края блока. Если высота блока не кратна 4, в последнем наборе коэффициентов из каждого столбца будет меньше 4 коэффициентов. На рисунке D.1 показан пример сканирования по шаблону для блока кодирования.



Рисунок D.1 – Пример шаблона сканирования битовой матрицы блока кодирования

D.2 Битовые коэффициенты и их значения

D.2.1 Обозначения в общем случае

Определенные в этом Приложении процедуры декодирования представлены для каждого коэффициента преобразования (u, v) поддиапазона b , декодированные биты которого будут использованы для восстановления значения коэффициента преобразования $q_b(u, v)$. Представлены следующие биты: знаковый бит $s_b(u, v)$ и число $N_b(u, v)$ декодированных значений MSB в порядке от наиболее значимого к наименее значимому. $MSB_i(b, u, v)$ – это i -ый наиболее значимый бит (MSB) коэффициента преобразования (u, v) поддиапазона b ($i = 1, \dots, N_b(u, v)$). Как видно из равенства (D-1), знаковый бит $s_b(u, v)$ принимает значение единицы для отрицательных коэффициентов и нуля – для положительных. Число $N_b(u, v)$ декодированных битов MSB включает число всех нулей наиболее значимых битовых матриц, о которых указано в заголовке пакета (см. В.10.5).

D.2.2 Обозначения для случая с областью ROI

В случае присутствия сегмента маркера RGN (указывающего на присутствие ROI), для декодирования битов необходимы модификации, а также числа декодированных битов $N_b(u, v)$. Эти модификации описаны в Н.1. В отсутствие сегмента маркера RGN не требуется никакой модификации.

D.3 Декодирующие проходы по битовым матрицам

Каждый коэффициент блока кодирования имеет бинарную переменную состояния, которая называется состоянием значимости. Во время процесса декодирования блока кодирования состояния значимости инициализируют как 0 (коэффициент незначительный) или как 1 (коэффициент значительный). "Состояние значимости" на битовой матрице меняется от незначительного к значительному в тех местах, где обнаружен бит величины наибольшего значения, равный 1. Вектор контекста данного текущего коэффициента – это бинарный вектор, который состоит из состояний значимости с 8-ю прилегающими друг к другу коэффициентами, как показано на рисунке D.2. С целью создания вектора контекста для декодирования текущих коэффициентов, любой близлежащий коэффициент, который не является обрабатываемым коэффициентом блока, полагают незначительным (т. е. имеющим нулевую значимость).

D_0	V_0	D_1
H_0	X	H_1
D_2	V_1	D_3

T.800_FD-2

Рисунок D.2 – Прилегающие состояния, которые используют для формирования контекста

В общем, текущий коэффициент имеет 256 возможных векторов контекста. Они объединяются в небольшое число контекстов в соответствии с правилами формирования контекста, описанными ниже. Для каждого из четырех кодирующих прохода определены четыре правила формирования контекста: кодирование значимости, кодирование знака, кодирование подстройки величины коэффициента и кодирование очистки. Эти операции кодирования выполняются за три кодирующих прохода по каждой битовой матрице: значимость и знак в проходе воспроизведения значимости, кодирование подстройки величины в проходе подстройки величины, а также очистка и кодирование знака в проходе очистки. Для данной операции кодирования метка контекста (или контекст), поступающая в устройство арифметического кодирования, – это метка, присвоенная текущему контексту коэффициента.

ПРИМЕЧАНИЕ.. – Хотя ради определенности в таблицах ниже для обозначения контекстов использованы отдельные целочисленные значения, символы для меток контекста зависят от их применения, и значения символов не предписаны этой Рекомендацией | Международным стандартом.

Для первой битовой матрицы текущего блока с ненулевым элементом выполняется только проход очистки. Оставшиеся битовые матрицы декодируют в три прохода. Каждый битовый коэффициент декодируют полностью за один из трех кодирующих проходов. В каком из проходов будет декодирован коэффициент, зависит от условий для этого прохода. В общем, проход воспроизведения значимости включает коэффициенты, которые предсказаны или с "наибольшей вероятностью" станут значимыми с соответствующими битовыми знаками. Проход подстройки величины включает значимые коэффициенты. Проход очистки включает все оставшиеся коэффициенты.

D.3.1 Декодирующий проход воспроизведения значимости

Восемь коэффициентов, которые прилегают к текущему (обозначенному на рисунке D.2 как X), используют для создания вектора контекста, который помещают в один из 9 контекстов, показанных в таблице D.1. Если коэффициент значимый, то для создания контекста ему присваивают значение 1, в противном случае – 0. Размещение контекстов зависит также от поддиапазона.

Таблица D.1 – Контексты для кодирующих проходов восстановления значимости и очистки

Поддиапазоны LL и LH (вертикальный проход)			Диапазон HL (горизонтальный проход)			Поддиапазон HH (диагональный проход)		Метка контекста ^a
$\sum H_i$	$\sum V_i$	$\sum D_i$	$\sum H_i$	$\sum V_i$	$\sum D_i$	$\sum(H_i + V_i)$	$\sum D_i$	
2	x^b	x	x	2	x	x	≥ 3	8
1	≥ 1	x	≥ 1	1	x	≥ 1	2	7
1	0	≥ 1	0	1	≥ 1	0	2	6
1	0	0	0	1	0	≥ 2	1	5
0	2	x	2	0	x	1	1	4
0	1	x	1	0	x	0	1	3
0	0	≥ 2	0	0	≥ 2	≥ 2	0	2
0	0	1	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0

a) Заметим, что в этой Рекомендации | Международном стандарте метки контекста индексируют только для удобства идентификации. Действительные идентификаторы зависят от применения.

b) "x" обозначает состояние, о котором "не заботятся".

Проход восстановления значимости включает только биты коэффициентов, которые были значительными (состояние значимости уже было установлено) и имеют ненулевой контекст. Все прочие коэффициенты пропускают. Контекст доставляют в арифметический декодер (вместе с потоком битов), а декодированные битовые коэффициенты возвращают. Если значение бита равно 1, то состояние значимости устанавливают на 1, а непосредственно следующим битом, подлежащим декодированию, является знаковый бит этого коэффициента. В противном случае состояние значимости остается 0. Когда рассматривают контексты последовательных коэффициентов, используют наиболее значимые текущие состояния.

D.3.2 Декодирование знакового бита

Метку контекста для декодирования знакового бита определяют, используя другой вектор контекста из окружения. Вычисление метки контекста можно рассматривать как процесс из двух шагов. Первый шаг суммирует вклад от вертикального и горизонтального окружения. Второй шаг снижает число этих вкладов до одной из 5 меток контекстов.

При первом шаге рассматривают совместно два вертикальных окружения (см. рисунке D.2). Каждое окружение может иметь одно из трех состояний: значимое положительное, значимое отрицательное или незначительное. Если оба вертикальных окружения – значимые с одним и тем же знаком, или если только одно из них значимое, тогда значение вертикального вклада единица для положительного знака и минус 1, – если знак отрицательный. Если оба вертикальных окружения незначительные или оба значимые, но с разными знаками, тогда вертикальный вклад равен 0. Горизонтальный вклад создается аналогичным образом. Напомним еще раз, что все окружающие значения вне этого блока считаются незначительными. Эти вклады показаны в таблице D.2 .

Таблица D.2 – Вклады от вертикальных и горизонтальных окружений в знак контекста

V0 (или H0)	V1 (или H1)	Вклад V (или H)
значимый, положительный	значимый, положительный	1
значимый, отрицательный	значимый, положительный	0
незначительный	значимый, положительный	1
значимый, положительный	значимый, отрицательный	0
значимый, отрицательный	значимый, отрицательный	-1
незначительный	значимый, отрицательный	-1
значимый, положительный	незначительный	1
значимый, отрицательный	незначительный	-1
незначительный	незначительный	0

При втором шаге девять перемещений вертикального и горизонтального вкладов уменьшают до пяти. В таблице D.3 показаны эти метки контекстов. Контекст доставляют в арифметический декодер вместе с потоком битов. Возвращенный декодером бит D (см. Приложение C) – это логическое исключаящее ИЛИ с $XORbit$, которые, как показано в таблице D.3, формируют знаковый бит. Используют следующее равенство:

$$signbit = D \otimes XORbit, \tag{D-1}$$

где *signbit* – это знак текущего коэффициента (один бит означает отрицательный коэффициент, а ноль – положительный), *D* – это значение, возвращенное декодером от данной метки и потока битов, а значение *XORbit* для текущей метки контекста указано в таблице D.3.

Таблица D.3 – Контексты знаков от вертикальных горизонтальных вкладов

Горизонтальный вклад	Вертикальный вклад	Метка контекста	XORbit
1	1	13	0
1	0	12	0
1	-1	11	0
0	1	10	0
0	0	9	0
0	-1	10	1
-1	1	11	1
-1	0	12	1
-1	-1	13	1

D.3.3 Проход подстройки величины

Проход подстройки величины включает биты от коэффициентов, которые являются значимыми (за исключением тех, которые уже стали значимыми непосредственно перед проходом воспроизведения значимости).

Использованный контекст определяют суммированием состояния значимости от горизонтального, вертикального и диагонального вкладов. Эти состояния известны декодеру как текущие, а не как состояния, использованные до прохода декодирования. Кроме того, это зависит от того, присутствует или нет первый бит подстройки (бит, который следует сразу после битов значимости и знака). В таблице D.4 показаны три контекста этого прохода.

Таблица D.4 – Контексты для подстройки величин кодирующих проходов

$\sum H_i + \sum V_i + \sum D_i$	Первая подстройка этого коэффициента	Метка контекста
x^a	ложь	16
≥ 1	истина	15
0	истина	14
a) "x" обозначает состояние, о котором "не заботятся".		

Контекст вместе с потоком битов направляют в арифметический кодер. Возвращенный бит – это значение текущего коэффициента в текущей битовой матрице.

D.3.4 Проход очистки

Оставшиеся коэффициенты первоначально считались незначительными и не обрабатывались проходом воспроизведения значимости. Проход очистки не только использует прилегающий контекст, такой как от прохода воспроизведения значимости по таблице D.1, но также длину контекста.

Во время прохода прилегающие контексты для коэффициентов в этом проходе восстанавливаются с помощью таблицы D.1. Теперь метка контекста должна иметь какую-нибудь величину, поскольку коэффициенты, которые во время прохода воспроизведения значимости были приняты как значимые, остаются значимыми и при проходе очистки. Длины декодируют с помощью особого единственного контекста. Если отсканированные в столбце четыре смежных коэффициента все декодированы при проходе очистки, и метка контекста для всех равна 0 (включая коэффициенты контекста от предыдущих проходов значимости, подстройки величин и очистки), то особый единственный контекст подают в арифметический декодер вместе с потоком битов. Если возвращенный символ 0, то все четыре смежных коэффициента в столбце считают незначительными и обращают в ноль.

В другом случае, если возвращенный символ 1, то, по крайней мере, один из четырех смежных коэффициентов в столбце – значимый. Следующие два бита, которые возвращаются с помощью контекста UNIFORM (индекс 46 в таблице C.2), указывают, какой из коэффициентов столбца сверху вниз является первым значащим. Два бита, которые декодируют с помощью контекста UNIFORM, – это биты наиболее значимый MSB, а затем наименее значимый LSB. Знаковый бит коэффициента определяют, как описано в D.3.2. Декодирование любого оставшегося коэффициента продолжают по правилу, описанному в D.3.1.

Если не все четыре смежных коэффициента в столбце декодированы во время прохода очистки или если карман контекста для любого из них не нулевой, то биты коэффициентов декодируют с помощью контекста таблицы D.1 точно так же, как во время прохода воспроизведения значимости. Здесь используют такой же контекст, как и в проходе воспроизведения значимости (т. е. используют состояние, а также модель). В таблице D.5 показана логика прохода очистки.

Таблица D.5 – Обрабатываемая декодером длина при проходах очистки

Четыре смежных коэффициента столбца, которые остались для декодирования, и каждый текущий имеет контекст 0	Символы с длиной контекста	Четыре смежных бита, предназначенные для декодирования, являются нулями	Символы, декодированные с помощью контекста UNIFORM ^{a)}	Число коэффициентов, предназначенных для декодирования
истина	0	истина	нет	нет
истина	1	ложь пропуск к первому знаку коэффициента пропуск ко второму знаку коэффициента пропуск к третьему знаку коэффициента пропуск к четвертому знаку коэффициента	MSB LSB 00 01 10 11	3 2 1 0
ложь	нет	x	нет	остаток столбца

^{a)} См. Приложение С.

Если в блоке кодирования остается меньше четырех рядов, то не используют никакую длину кодирования. И еще, состояние значимости любого коэффициента меняется сразу после декодирования первой величины бита 1.

D.3.5 Пример кодирующих проходов и восстановления значимости (информативное)

В таблице D.6 показан пример порядка декодирования квантованных коэффициентов для одного из четырех сканированных столбца. В этом примере полагаем, что все прилегающие структуры, не включенные в таблицу, равны нулю. В примере показано, в каком кодирующем проходе декодируют каждый бит. Знак бита декодируют после собственного единичного бита, и этот знак показан в таблице знаком + или -. Самый первый проход в новом блоке всегда является проходом очистки, поскольку в нем не может быть битов с предсказанной значимостью или с подстройкой. После первого прохода декодированный единичный бит первого коэффициента вызывает декодирование второго коэффициента в проходе значимости следующей битовой матрицы. Декодированный единичный бит последнего коэффициента при втором проходе очистки вызывает декодирование третьего коэффициента при следующем проходе значимости.

Таблица D.6 – Пример порядка кодирования части битовой матрицы и восстановления значимости

Кодирующие проходы	10	1	3	-7	Значение коэффициента
	+	+	+	-	Знак коэффициента
	1	0	0	0	Величина коэффициента (от MSB до LSB)
	0	0	0	1	
	1	0	1	1	
	0	1	1	1	
Очистка	1+	0	0	0	
Значимость		0			
Подстройка	0				
Очистка			0	1-	
Значимость		0	1+		
Подстройка	1			1	
Очистка					
Значимость		1+			
Подстройка	0		1	1	
Очистка					

D.4 Инициализация и завершение

Когда контексты инициализированы или повторно инициализированы, их значения устанавливаются, как показано в таблице D.7.

Таблица D.7 – Исходные состояния для всех контекстов

Контекст	Исходный индекс из таблицы C.3	MPS
UNIFORM	46	0
Длина	3	0
Нули для всех прилегающих (в таблице D.1 метка контекста 0)	4	0
Все прочие контексты	0	0

При нормальном режиме (нет избирательного обхода арифметического кодирования) арифметический кодер должен завершать работу либо в конце каждого кодирующего прохода, либо только в конце каждого блока кодирования (см. D.4.1). В таблице D.8 показаны два примера завершения кодирующих проходов в блоке кодирования. Для завершения работы использованы сигналы маркеров COD или СОС (см. A.6.1 и A.6.2).

Таблица D.8 – Шаблоны завершения работы арифметического кодера

#	Проход	Операция завершения кодирования только при последнем проходе	Операция завершения кодирования при каждом проходе
1	очистка	Арифметический кодер (АС)	АС, завершение
2	восстановление значимости	АС	АС, завершение
2	подстройка величины	АС	АС, завершение
2	очистка	АС	АС, завершение
...
конец	восстановление значимости	АС	АС, завершение
конец	подстройка величины	АС	АС, завершение
конец	очистка	АС, завершение	АС, завершение

Если работа арифметического кодера завершается многократно, то о длине каждого сегмента завершения сообщается в заголовке пакета, как описано в В.10.7.

ПРИМЕЧАНИЕ. – При завершении никогда не создается байт, расположенный в диапазоне от 0xFF90 и включая 0xFFFF. Эти значения допустимы только как значения маркеров в потоке битов.

D.4.1 Ожидаемое завершение кодированного потока

Декодер предвидит, что данное число байтов кодированного потока будет декодировано данным числом кодирующих проходов до того, как арифметический кодер завершит работу. В процессе декодирования из кодированного потока байты поступают последовательно до тех пор, пока не будут выбраны все байты, предназначенные для этих кодирующих проходов. Число байтов, соответствующее кодирующим проходам, определено в заголовке пакета. Часто в этом месте имеются дополнительные символы, подлежащие декодированию. Поэтому декодер должен быть рассчитан на передачу более широкого потока в арифметический кодер (с помощью байтов 0xFF), чем необходимо, чтобы были декодированы все символы.

Достаточно присоединить не более двух байтов 0xFF. Это вызовет появление на входе арифметического кодера, по крайней мере, пары последовательных байтов 0xFF, которые будут интерпретированы как маркер окончания потока (см. C.3.4). Настоящий поток битов не имеет маркера завершения. Однако длина байта будет явным сигналом, который позволит маркеру завершить синхронизацию с арифметическим декодером.

ПРИМЕЧАНИЕ. – Присоединенные таким образом два байта представляют простейший способ. Однако существуют и другие эквивалентные способы. Это может играть роль, поскольку при некоторых реализациях арифметический кодер может придавать особое значение маркеру завершения.

D.4.2 Завершение работы арифметического кодера

Эту задачу выполняет процедура FLUSH (см. С.2.9). Однако, поскольку процедура FLUSH увеличивает длину кодированного потока, а желательным может оказаться быстрое завершение, возможно использование и других методов. При любом методе требуется размещение в потоке всех необходимых байтов таким образом, чтобы от декодера не требовалось сигнала возврата для поиска позиции, в которой должен начинаться следующий сегмент кодированного потока.

Если установлен флаг предсказуемого завершения (см. COD и СОС в А.6.1 и А.6.2), должны быть использованы указанные ниже процедуры завершения. В следующих шагах использованы обозначения из С.2:

- 1) Идентифицировать число битов в кодовом регистре С, которое должно пройти через буфер байтов. Это число задано как $k = (11 - СТ) + 1$.
- 2) Для $k > 0$:
 - сдвинуть С влево на СТ и установить $СТ = 0$.
 - выполнить процедуру BYTEOUT. При этом СТ станет равным числу битов, сброшенных из регистра С.
 - вычесть СТ из k.
- 3) Выполнить процедуру BYTEOUT, чтобы переместить в кодированный поток содержимое буфера байтов регистра. Если в буфере байтов имеется значение байта 0xFF, этот шаг должен быть пропущен.

Соответствующая этому случаю сокращенная длина – это просто общее число байтов, перемещенных в кодированный поток.

Если флаг предсказуемого завершения не установлен, последний байт на выходе в описанной выше процедуре должен быть модифицирован в определенных границах, не затрагивая символы, которые предназначаются для декодирования. В некоторых случаях возможно увеличение значения последнего байта до значения 0xFF, которое не должно отправляться. Можно заметить, что это случается примерно в 1/8 от общего времени.

D.4.3 Вычисление длины (информативное)

Чтобы заключить данные сжатого изображения в пакеты, необходимо определить число байтов, которые требуются включить. Если закончен кодирующий проход данных сжатого изображения, можно использовать алгоритм, описанный в предыдущем разделе. В противном случае, кодировщик должен подсчитать, достаточна ли длина соответствующих байтов, чтобы декодер смог восстановить кодирующие проходы.

D.5 Символ сегментации для защиты от ошибок

Символ сегментации – это специальный символ, который используют для сигнализации в сегментах маркеров COD или СОС (см. А.6.1 и А.6.2). В конце каждой битовой матрицы этот символ кодируют с помощью контекста UNIFORM арифметического кодера. Правильность декодирования этого символа подтверждает правильность декодирования всей битовой матрицы, что позволяет обнаруживать ошибки. Символ сегментации в декодере, 1010 или 0xA, должен декодироваться в конце каждой битовой матрицы (в конце прохода очистки). Если символ сегментации правильно не декодирован, в этой битовой матрице присутствуют ошибки.

ПРИМЕЧАНИЕ. – Эту операцию можно использовать как при предсказуемом завершении, так и без него.

D.6 Избирательный обход арифметического кодирования

Этот тип кодирования допускает обход арифметического кодера для проходов восстановления значимости и подстройки величины, которые начинаются с пятой значимой битовой матрицы блока кодирования. Независимо от типа кодирования используют сигналы маркеров COD или СОС (см. А.6.1 и А.6.2).

Первый проход очистки (который начинается с первой битовой матрицы блока кодирования с ненулевым элементом) и три следующие серии кодирующих проходов – воспроизведения значимости, подстройки величины и очистки – декодируют арифметическим кодером. Четвертый проход – очистка – должен включать завершение работы арифметического кодера (см. таблицу D.9).

Таблица D.9 – Избирательный обход арифметического кодирования

Номер битовой матрицы	Тип прохода	Операции кодирования
1	очистка	арифметическое кодирование (АС)
2	воспроизведение значимости	АС
2	подстройка величины	АС
2	очистка	АС
3	воспроизведение значимости	АС
3	подстройка величины	АС
3	очистка	АС
4	воспроизведение значимости	АС
4	подстройка величины	АС
4	очистка	АС, завершение
5	воспроизведение значимости	незаконченный
5	подстройка величины	незаконченный, завершение
5	очистка	АС, завершение
...
последний	воспроизведение значимости	незаконченный
последний	подстройка величины	незаконченный, завершение
последний	очистка	АС, завершение

Начиная с четвертого прохода восстановления значимости и подстройки величины, биты, которые должны были возвращаться от арифметического кодера, вместо этого поступают непосредственно из потока битов. (Процедура, которая отменяет эффект вставки битов, предшествует возврату битов. Особенность процедуры – удаление первого бита после байта со значением 0xFF.) После прохода подстройки величины поток битов "завершен" заполнением границы байтов.

Если установлен флаг предсказуемого завершения (см. COD и СОС в А.6.1 и А.6.2) и собраны все биты от прохода настройки величины, все оставшиеся биты в последнем байте заполняют альтернативной последовательностью нулей и единиц. Эта последовательность должна начинаться с 0, независимо от числа битов, которые предстоит заполнить.

Если установлен флаг каждого прохода кодирования (см. COD и СОС в А.6.1 и А.6.2), проходы восстановления значимости завершаются таким же образом, что и проходы подстройки величины.

Проходы очистки продолжают получать данные сжатого изображения непосредственно от арифметического кодера и всегда завершаются.

Знаковый бит вычисляют из равенства (D-2):

$$\text{signbit} = \text{raw_value}, \quad (\text{D-2})$$

где $\text{raw_value} = 1$ – отрицательный знаковый бит, а 1 – положительный. В таблице D.9 показана последовательность кодирования.

О сумме длин каждого завершенного сегмента и оставшихся незавершенных проходов в заголовок пакета подается сигнал, как описано в В.10.7. Если выбором определено завершение каждого прохода кодирования (см. А.6.1 и А.6.2), то каждый проход (включая незаконченные) будет завершен.

ПРИМЕЧАНИЕ 1. – Если изображение кодируют с помощью ROI, использование режима избирательного обхода может значительно снизить эффективность сжатия.

Если в потоке битов встречается величина 0xFF, то первый бит следующего байта отбрасывается. Последовательность битов, использованных при избирательном обходе арифметического кодирования, может быть вставлена в байты процедурой вставки битов (стаффинга).

В кодировщике биты упаковывают в байты от наиболее значимого к наименее значимому биту. Как только байт собран, его направляют в поток битов. Если значение байта – 0xFF, в наиболее значимый бит следующего байта вставляют единственный нулевой бит. Когда собраны все биты прохода кодирования, последний байт пакуют на границе байтов и отправляют. Последний байт никогда не должен принимать значение 0xFF.

ПРИМЕЧАНИЕ 2. – Поскольку декодер присоединяет по правилам значения 0xFF к потоку битов, представляющему проход кодирования (см. D.4.1), возможно усечение потока битов. Если установлен флаг предсказуемого завершения (см. COD и СОС в А.6.1 и А.6.2), то такое усечение запрещено. Последний байт не может иметь значение 0xFF, поскольку процедура стаффинга битов присоединяет новый байт, следующий за FF, который имеет нулевое значение наиболее значимого бита и неиспользуемые биты, заполненные альтернативной последовательностью значений битов 0 и 1.

D.7 Формирование вертикального причинного контекста

Этот тип кодирования ограничивает формирование контекста текущим и прошедшим сканированиями блока кодирования (четыре ряда сканированных коэффициентов по вертикали). То есть любой коэффициент от сканирования следующего блока кодирования считают незначительным. При этом независимо от типа кодирования используют сигналы маркеров COD или СОС (см. А.6.1 и А.6.2).

Для иллюстрации на рисунке D.1 бит, обозначенный как 14, декодируют обычным образом, используя смежные состояния, как описано на рисунке D.2, независимо от того, является ли контекст вертикально причинным. Однако, если используют формирование вертикального причинного контекста, бит, обозначенный как 15, декодируют, полагая $D_2 = V_1 = D_3 = 0$ на рисунке D.2.

D.8 График последовательности кодирования блока

На рисунке D.3 графически показаны шаги моделирования каждой битовой матрицы каждого блока кодирования. Принятые решения показаны в таблице D.10, а биты и контекст, отправленные в кодер, приведены в таблице D.11. На рисунке и в таблицах контекст показан без избирательного обхода арифметического кодирования или вертикальной причинной модели.

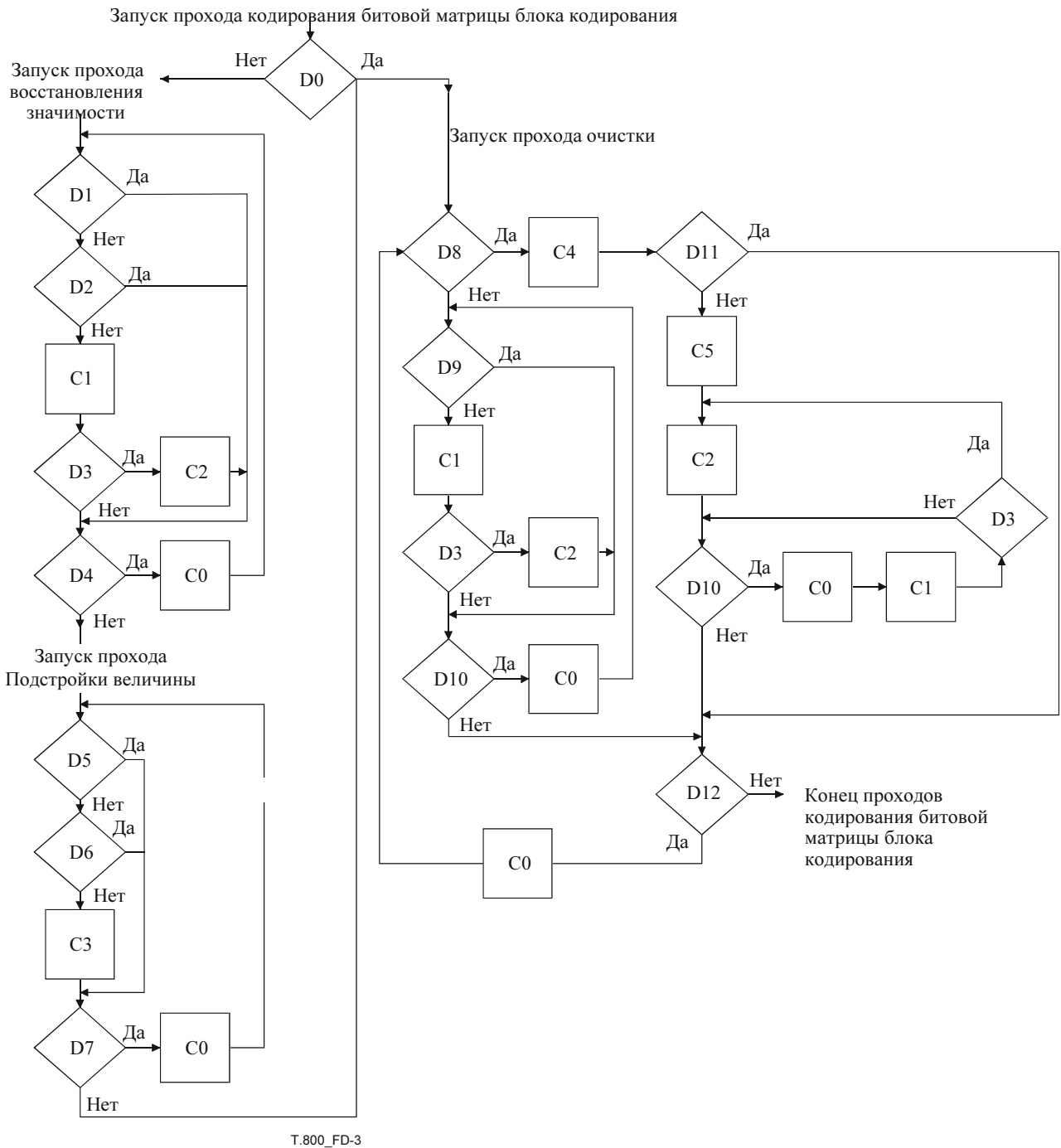


Рисунок D.3 – Последовательность операций для всех проходов кодирования битовой матрицы блока кодирования

Таблица D.10 – Решения в последовательности операций модели контекста

Решение	Вопрос	Решение
D0	Это первая значимая битовая матрица блока кодирования?	см. D.3
D1	Этот текущий коэффициент значимый?	см. D.3.1
D2	Это нулевой карман контекста? (см. табл. D.1)	см. D.3.1
D3	Действительно ли этот текущий коэффициент стал значимым?	см. D.3.1
D4	Существуют ли еще коэффициенты воспроизведения значимости?	
D5	Этот коэффициент незначительный?	см. D.3.3
D6	Кодировался ли этот коэффициент при последнем проходе воспроизведения значимости?	см. D.3.3
D7	Есть ли еще коэффициенты в этом проходе подстройки величины?	
D8	Является ли каждый из этих четырех смежных незакодированных коэффициентов столбца с нулевым контекстом?	см. D.3.4
D9	Этот коэффициент значимый или был ли этот бит закодирован во время прохода значимости воспроизведения?	см. D.3.4
D10	Остаются ли еще коэффициенты из этих четырех столбцов коэффициентов?	
D11	Все четыре смежных бита являются нулями?	см. D.3.4
D12	Остались еще коэффициенты при проходе очистки?	

Таблица D.11 – Последовательности операций декодирования модели контекста

Код	Декодированный символ	Контекст	Краткое пояснение	Описание
C0	–	–	Переходить к следующему коэффициенту или столбцу	
C1	Новый значимый?	Таблица D.1, 9 меток контекста	Декодировать значимый бит текущего коэффициента (восстановление значимости или очистка)	см. D.3.1
C2	Знаковый бит	Таблица D.3, 5 меток контекста	Декодировать знаковый бит текущего коэффициента	см. D.3.2
C3	Бит текущей величины	Таблица D.4, 3 метки контекста	Декодировать бит текущего коэффициента при проходе подстройки величины	см. D.3.3
C4	0 1	Метка длины контекста	Декодировать длину из четырех нулей. Декодировать длину не из четырех нулей	см. D.3.4
C5	00 01 10 11	UNIFORM	Первый коэффициент – первый с ненулевым битом Второй коэффициент – первый с ненулевым битом Третий коэффициент – первый с ненулевым битом Четвертый коэффициент – первый с ненулевым битом	см. D.3.4 и табл. C.2

Приложение Е

Квантование

(Данное Приложение составляет неотъемлемую часть данной Рекомендации | Международного стандарта)

В этом Приложении и во всех его подразделах блок-схемы и таблицы являются нормативными (обязательными) только в том смысле, что они определяют выходные данные, которые при альтернативных применениях должны повторяться.

Данное Приложение определяет формы инверсного квантования, которые используют для восстановления коэффициентов преобразования компонента тайла. В этом Приложении предусмотрена информация о квантовании коэффициентов преобразования для кодирования. С помощью процесса квантования коэффициенты преобразования понижаются в точности.

Е.1 Процедура инверсного квантования

Для каждого коэффициента преобразования (u, v) данного поддиапазона b значение коэффициента преобразования $q_b(u, v)$ задают следующим равенством:

$$\overline{q}_b(u, v) = (1 - 2s_b(u, v)) \cdot \left(\sum_{i=1}^{N_b(u, v)} MSB_i(b, u, v) \cdot 2^{M_b - i} \right), \quad (E-1)$$

где $s_b(u, v)$, $N_b(u, v)$ и $MSB_i(b, u, v)$ описаны в D.2, а значение M_b рассчитывают из равенства (E-2), где число защитных битов G и показатель степени ε_b определены в сегментах маркера QCD или QCC (см. A.6.4 и A.6.5).

$$M_b = G + \varepsilon_b - 1. \quad (E-2)$$

Для формирования восстановленного коэффициента преобразования $Rq_b(u, v)$ используют каждый декодированный коэффициент преобразования $q_b(u, v)$ поддиапазона b . Это описано в E.1.1.

ПРИМЕЧАНИЕ. – Декодирование только битовых матриц $N_b(u, v)$ (см. D.2.1) эквивалентно декодированию данных, которые кодированы при использовании скалярного квантователя с длиной шага $2^{M_b - N_b(u, v)} \cdot \Delta_b$ для всех коэффициентов этого блока кодирования. Из-за характера трех кодирующих проходов (см. D.3) значение $N_b(u, v)$ может быть разным для различных коэффициентов внутри одного и того же блока кодирования.

Е.1.1 Необратимое преобразование

Е.1.1.1 Определение длины шага квантования

Длину шага квантования Δ_b данного поддиапазона b для необратимого преобразования рассчитывают по динамическому диапазону R_b поддиапазона b , показателю степени ε_b и мантиссе μ_b :

$$\Delta_b = 2^{R_b - \varepsilon_b} \left(1 + \frac{\mu_b}{2^{11}} \right). \quad (E-3)$$

ПРИМЕЧАНИЕ. – В Равенстве (E-3) знаменатель 2^{11} обусловлен распределением 11 битов в кодовом потоке для мантиссы μ_b , см. таблицу A.30.

В равенстве (E-3) показатель степени ε_b и мантиссу μ_b определяют в сегментах маркера QCD или QCC (см. A.6.4 и A.6.5). Номинальный динамический диапазон R_b (см. равенство (E-4)) представляет собой сумму R_l (число битов, используемых для представления примеров оригинального компонента тайла, которые могут быть извлечены из маркера SIZ – см. таблицу A.11 в A.5.1) и основания показателя степени 2 коэффициента передачи поддиапазона ($gain_b$) текущего поддиапазона b . Коэффициент передачи поддиапазона ($gain_b$) изменяется пропорционально виду поддиапазона b ($levLL$, $levLH$ или $levHL$, $levHH$, см. F.3.1), см. таблицу E.1.

Таблица Е.1 – Коэффициент передачи поддиапазона

поддиапазон b	$gain_b$	$\log_2(gain_b)$
$levLL$	1	0
$levLH$	2	1
$levHL$	2	1
$levHH$	4	2

$$R_b = R_L + \log_2(gain_b) \quad (E-4)$$

О паре показателей степени/мантисса (ϵ_b, μ_b) сообщают либо в кодированный поток для каждого поддиапазона (расширенное квантование), либо сообщают только для поддиапазона $N_{L}LL$ и отводят для всех остальных поддиапазонов (ответвленное квантование) (см. таблицу А.30). В случае ответвленного квантования все пары показателя степени/мантисса (ϵ_b, μ_b) получают из единичной пары показателя степени/мантисса (ϵ_o, μ_o), которая соответствует поддиапазону $N_{L}LL$:

$$(\epsilon_b, \mu_b) = (\epsilon_o - N_L + n_b, \mu_o), \quad (E-5)$$

где n_b обозначает число уровней декомпозиции из исходного компонента тайла в поддиапазон b .

ПРИМЕЧАНИЕ. – Квантованный коэффициент преобразования для данного поддиапазона b может превышать динамический диапазон R_b .

Е.1.1.2 Восстановление коэффициента преобразования

Равенство (Е-6) представляет восстановленный коэффициент преобразования для необратимого преобразования:

$$Rq_b(u, v) = \begin{cases} (\overline{q}_b(u, v) + r2^{M_b - N_b(u, v)}) \cdot \Delta_b & \text{для } \overline{q}_b(u, v) > 0 \\ (\overline{q}_b(u, v) - r2^{M_b - N_b(u, v)}) \cdot \Delta_b & \text{для } \overline{q}_b(u, v) < 0, \\ 0 & \text{для } \overline{q}_b(u, v) = 0 \end{cases} \quad (E-6)$$

где r – восстановленный параметр, который декодер может выбирать произвольно.

ПРИМЕЧАНИЕ. – Восстановленный параметр r может быть выбран, например, для того, чтобы (при восстановлении) добиться наглядности или объективного качества. Как правило, значения параметра r понижают в диапазоне $0 \leq r < 1$, а общее значение $r = 1/2$. (Это примечание относится и к Е.1.2.)

Е.1.2 Обратимое преобразование

Е.1.2.1 Определение длины шага квантования

Для обратимого преобразования длина шага квантования Δ_b равна единице (квантование не выполняют).

Е.1.2.2 Восстановление коэффициента преобразования

При обратимом преобразовании восстановление реконструированного коэффициента преобразования $Rq_b(u, v)$ происходит в зависимости от того, декодированы ли все биты коэффициента, то есть, какое из двух условий выполняется: $N_b(u, v) = M_b$ или $N_b(u, v) < M_b$.

Равенство (Е-7) представляет восстановленный коэффициент преобразования $Rq_b(u, v)$ для случая $N_b(u, v) = M_b$:

$$Rq_b(u, v) = \overline{q}_b(u, v). \quad (E-7)$$

Равенство (Е-8) представляет восстановленный коэффициент преобразования $Rq_b(u, v)$ для случая $N_b(u, v) < M_b$:

$$Rq_b(u, v) = \begin{cases} \left\lceil \left(\overline{q}_b(u, v) + r2^{M_b - N_b(u, v)} \right) \cdot \Delta_b \right\rceil & \text{для } \overline{q}_b(u, v) > 0 \\ \left\lfloor \left(\overline{q}_b(u, v) - r2^{M_b - N_b(u, v)} \right) \cdot \Delta_b \right\rfloor & \text{для } \overline{q}_b(u, v) < 0, \\ 0 & \text{для } \overline{q}_b(u, v) = 0 \end{cases} \quad (\text{E-8})$$

Е.2 Квантование скалярного коэффициента (информативное)

Для необратимого сжатия после необратимого прямого дискретного частотного преобразования (см. Приложение F), каждый из коэффициентов преобразования $a_b(u, v)$ поддиапазона квантуют до значения $q_b(u, v)$. Равенство (E-9) представляет значение:

$$q_b(u, v) = \text{sign}(a_b(u, v)) \cdot \left\lfloor \frac{|a_b(u, v)|}{\Delta_b} \right\rfloor, \quad (\text{E-9})$$

где Δ_b – длина шага квантования. Показатель степени ϵ_b и мантисса, соответствующие Δ_b , могут быть получены из равенства (E-5). Эти значения должны быть записаны в маркерах кодированного потока QCD или QCC (см. A.6.4 и A.6.5).

При обратимом сжатии длина шага квантования должна быть равна 1. В этом случае параметр ϵ_b следует записать в маркерах кодированного потока QCD или QCC (см. A.6.4 и A.6.5). Параметр ϵ_b рассчитывают следующим образом:

$$\epsilon_b = R_I + \log_2(\text{gain}_b) + \zeta_c, \quad (\text{E-10})$$

где значения R_I и gain_b описаны в E.1.1. Если преобразование RCT не используют, значение ζ_c равно нулю, в противном случае значение ζ_c равно числу дополнительных битов, которые добавляет преобразование RCT (см. G.2.1).

При обратимом и необратимом сжатии возможно переполнение или отклонение от номинального диапазона целочисленного представления $|q_b(u, v)|$ (например, во время подсчета плавающей запятой). Чтобы предотвратить это явление, количество битов M_b для целочисленного представления $q_b(u, v)$, которые используют со стороны кодировщика, определяют равенством (E-2). В маркерах QCD или QCC следует определять количество защитных битов G (см. A.6.4 и A.6.5). Типичные значения количества защитных битов составляют $G = 1$ или $G = 2$.

Приложение F

Дискретное частотное (волновое) преобразование компонентов тайла

(Это Приложение составляет неотъемлемую часть данной Рекомендации | Международного стандарта)

В этом Приложении блок-схемы и таблицы являются нормативными (обязательными) только в том смысле, что они определяют выходные данные, которые при альтернативных применениях должны повторяться.

В этом Приложении описано прямое дискретное частотное преобразование, используемое для одного компонента тайла, а также описано инверсное дискретное частотное преобразование, используемое для восстановления компонента тайла.

F.1 Параметры компонента тайла

Рассмотрим компонент тайла, определенный координатами tcx_0 , tcx_1 , tcy_0 и tcy_1 , заданными равенством (B-12) в Приложении B.3. Координаты (x, y) компонента тайла со значениями образца $I(x, y)$ расположены в диапазоне:

$$tcx_0 \leq x < tcx_1 \text{ и } tcy_0 \leq y < tcy_1. \quad (F-1)$$

F.2 Дискретные частотные преобразования

F.2.1 Низкочастотная и высокочастотная фильтрация (информативное)

Для прямого дискретного частотного преобразования (FDWT) в данной Рекомендации | Международном стандарте используют преобразование одномерной декомпозиции поддиапазона одномерного массива образцов в низкочастотные и высокочастотные коэффициенты. Низкочастотные коэффициенты представляют вариант выборки из массива оригинала образцов с низким разрешением. Высокочастотные коэффициенты представляют вариант выборки оставшихся образцов, необходимых для более тщательного восстановления оригинала из низкочастотного массива.

Для инверсного дискретного частотного преобразования (IDWT) в данной Рекомендации | Международном стандарте используют восстановление одномерного поддиапазона одномерного массива образцов из низкочастотных и высокочастотных коэффициентов.

F.2.2 Уровни декомпозиции

Каждый компонент тайла преобразуют в сигналы двухразмерных поддиапазонов (называемых поддиапазонами), каждый из которых представляет действие этого сигнала в разных частотных диапазонах и с различными пространственными разрешениями. N_L обозначает число уровней декомпозиции.

F.2.3 Дискретные частотные фильтры (информативное)

Эта Рекомендация | Международный стандарт определяет одно обратимое и одно необратимое преобразование. При условии, что образцы компонента тайла являются целочисленными, обратимое преобразование требует детального описания процедуры округления для промежуточного преобразования нецелочисленных коэффициентов.

F.3 Необратимое дискретное частотное преобразование

F.3.1 Процедура IDWT

Инверсное дискретное частотное преобразование (IDWT) преобразует набор поддиапазонов $a_b(u_b, v_b)$ в сдвиги по уровню постоянного тока (DC) компонента тайла $I(x, y)$ (процедура IDWT). Кроме того, процедура IDWT принимает на входе параметр N_L , который представляет число уровней декомпозиции (см. рисунок F.1). О числе уровней декомпозиции N_L сообщается в маркерах COD или COC (см. A.6.1 и A.6.2).

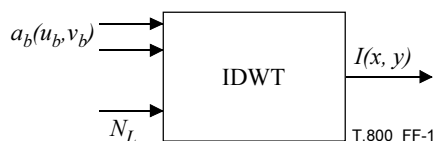


Рисунок F.1 – Входы и выходы процедуры IDWT

Поддиапазоны обозначают следующим образом: за индексом lev , соответствующим уровню декомпозиции, следуют две буквы, LL, HL, LH или HH.

Поддиапазон $b = levLL$ соответствует низкочастотному варианту поддиапазона $(lev - 1)LL$, который отфильтрован низкочастотным фильтром по вертикали и низкочастотным – по горизонтали. Поддиапазон $b = 0LL$ соответствует оригиналу компонента тайла. Поддиапазон $b = levHL$ соответствует варианту поддиапазона $(lev - 1)LL$, который отфильтрован низкочастотным фильтром по вертикали и высокочастотным – по горизонтали. Поддиапазон $b = levLH$ соответствует варианту поддиапазона $(lev - 1)LL$, который отфильтрован высокочастотным фильтром по вертикали и низкочастотным – по горизонтали. Поддиапазон $b = levHH$ соответствует варианту поддиапазона $(lev - 1)LL$, который отфильтрован высокочастотным фильтром по вертикали и высокочастотным – по горизонтали.

Для заданного значения N_L в кодированном потоке представлены только следующие поддиапазоны в следующем порядке (эти поддиапазоны достаточны для полного восстановления оригинала компонента тайла):

$$N_L LL, N_L HL, N_L LH, N_L HH, (N_L - 1)HL, (N_L - 1)LH, (N_L - 1)HH, \dots, 1HL, 1LH, 1HH.$$

Для данного поддиапазона b число n_b представляет уровень декомпозиции, на котором он производится во время кодирования. Этот уровень представлен в таблице F.1:

Таблица F.1 – Уровень декомпозиции n_b для поддиапазона b

b	$N_L LL$	$N_L HL$	$N_L LH$	$N_L HH$	$(N_L - 1)HL$	$(N_L - 1)LH$	$(N_L - 1)HH$...	1HL	1LH	1HH
n_b	N_L	N_L	N_L	N_L	$N_L - 1$	$N_L - 1$	$N_L - 1$...	1	1	1

Для случая, когда $N_L = 2$, поддиапазоны показаны на рисунке F.2.

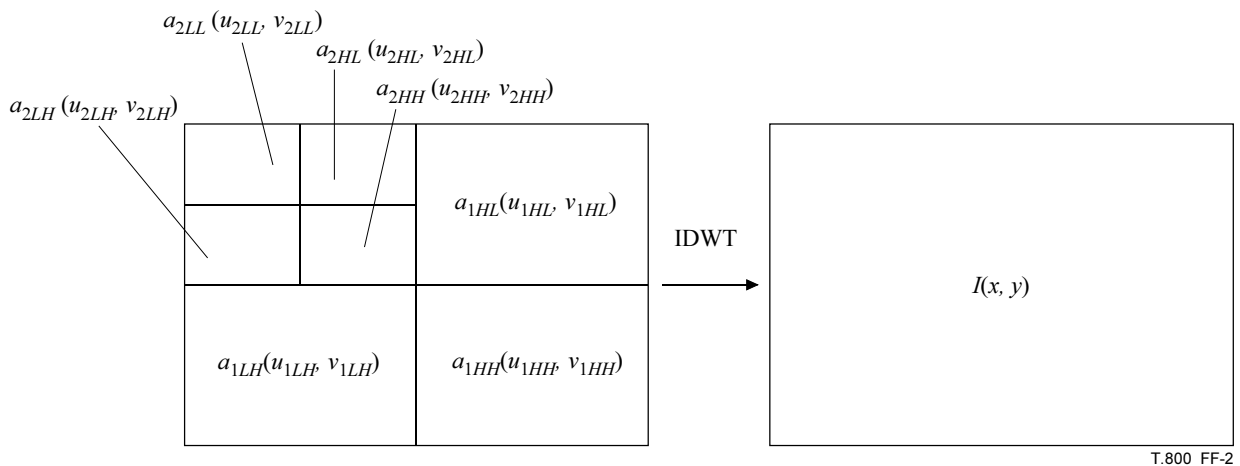


Рисунок F.2 – Процедура IDWT для случая ($N_L = 2$)

Процедура IDWT запускается инициализацией переменной lev от текущего уровня декомпозиции до N_L . Процедура (см. F.3.2) выполняется на каждом уровне lev , где этот уровень понижается с каждой итерацией до тех пор, пока не будет выполнено N_L итераций. Процедуру 2D_SR повторяют над поддиапазоном $levLL$, образованным каждой итерацией. Наконец, поддиапазон $a_{0LL}(u_{0LL}, v_{0LL})$ становится выходным массивом $I(x, y)$.

Как определено равенством (B-15), индексы (u_b, v_b) коэффициентов поддиапазонов $a_b(u_b, v_b)$ для данного поддиапазона b расположены в диапазонах:

$$tbx_0 \leq u_b < tbx_1 \text{ и } tby_0 \leq v_b < tby_1. \tag{F-2}$$

Процедура IDWT описана на рисунке F.3.

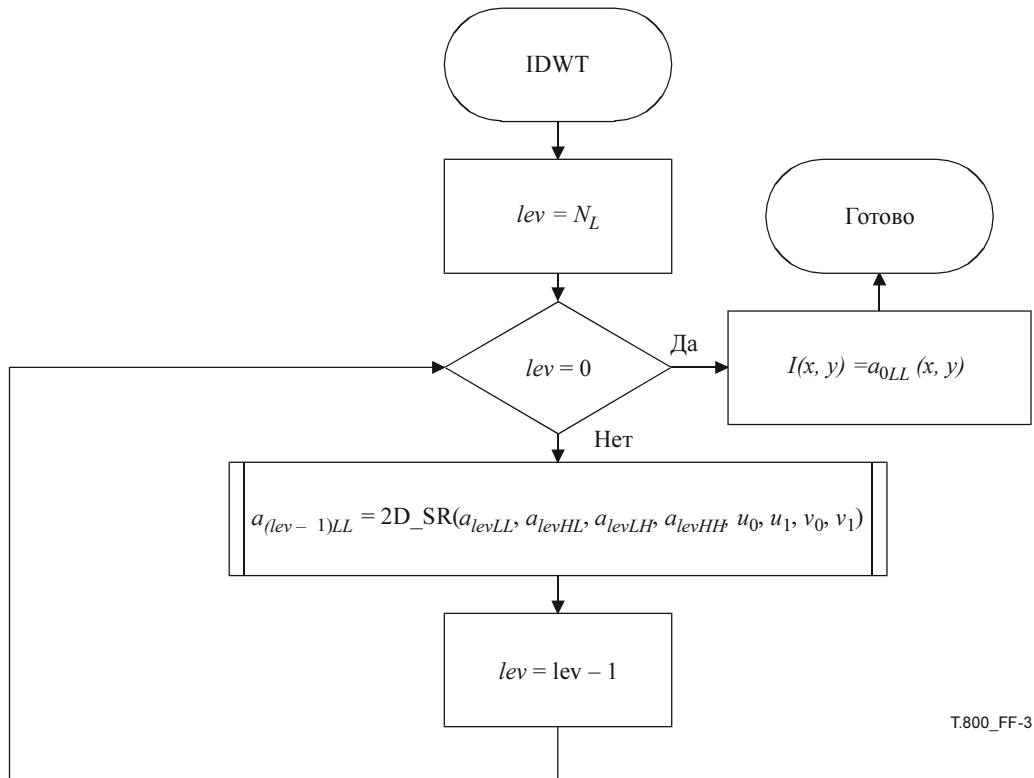


Рисунок F.3 – Процедура IDWT

F.3.2 Процедура 2D_SR

Процедура 2D_SR производит восстановление поддиапазона $a_{(lev-1)LL}(u, v)$ из четырех поддиапазонов $a_{levLL}(u, v)$, $a_{levHL}(u, v)$, $a_{levLH}(u, v)$ и $a_{levHH}(u, v)$ (см. рисунок F.4). Общее число коэффициентов восстановленного поддиапазона $levLL$ равно сумме общего числа коэффициентов четырех поддиапазонов на входе процедуры 2D_SR (см. рисунок F.5).

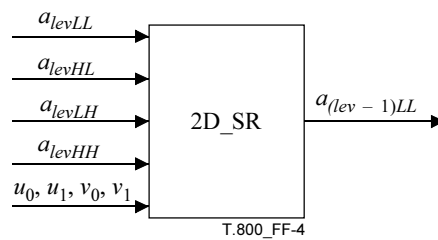


Рисунок F.4 – Входы и выходы процедуры 2D_SR

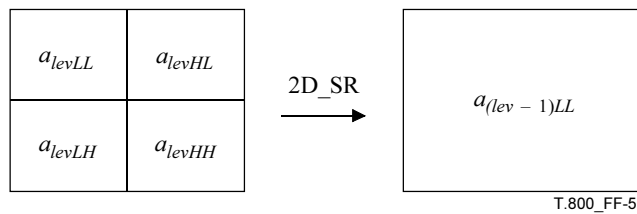


Рисунок F.5 – Один уровень восстановления в один поддиапазон из четырех (процедура 2D_SR)

Сначала четыре поддиапазона $a_{levLL}(u, v)$, $a_{levHL}(u, v)$, $a_{levLH}(u, v)$ и $a_{levHH}(u, v)$ чередуют для формирования массива $a(u, v)$, используя процедуру 2D_INTERLEAVE. Затем процедура 2D_SR применяет ко всем рядам $a(u, v)$ процедуру HOR_SR и наконец применяет процедуру VER_SR ко всем столбцам $a(u, v)$, чтобы восстановить поддиапазон $a_{(lev-1)LL}(u, v)$. Процедура 2D_SR показана на рисунке F.6.

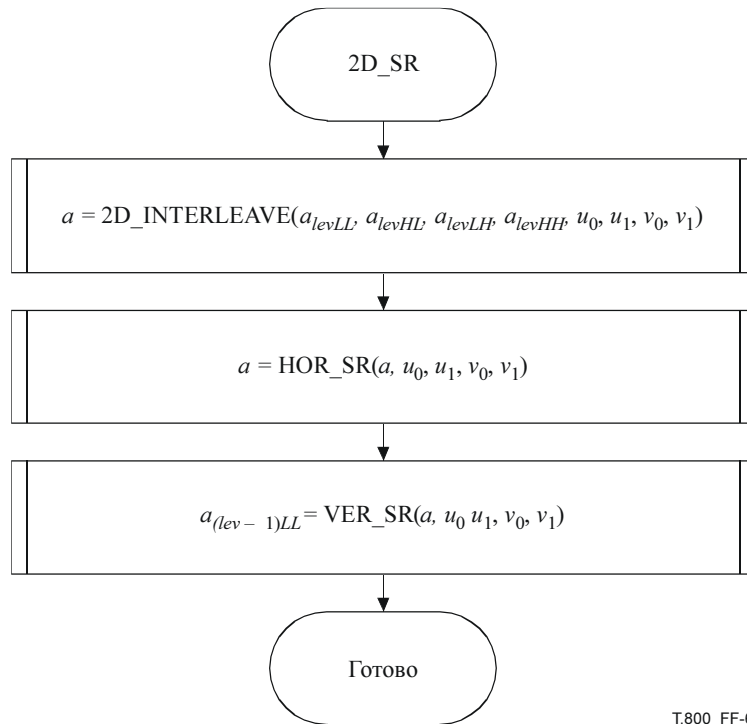


Рисунок F.6 – Процедура 2D_SR

F.3.3 Процедура чередования 2D_INTERLEAVE

Как показано на рисунке F.7, процедура 2D_INTERLEAVE чередует коэффициенты от четырех поддиапазонов a_{levLL} , a_{levHL} , a_{levLH} , a_{levHH} для формирования $a(u, v)$. Значения u_0, u_1, v_0, v_1 , которые использует процедура 2D_INTERLEAVE, это значения $tbx_0, tbx_1, tby_0, tby_1$ соответствующего поддиапазона $b = (lev - 1)LL$ (см. определение в равенстве (B-15)).

Способ, которым чередуют эти поддиапазоны для формирования выходного значения $a(u, v)$, описан процедурой 2D_INTERLEAVE, показанной на рисунке F.8.

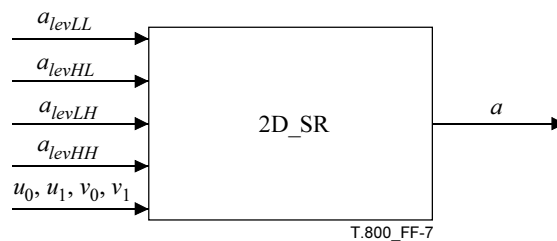
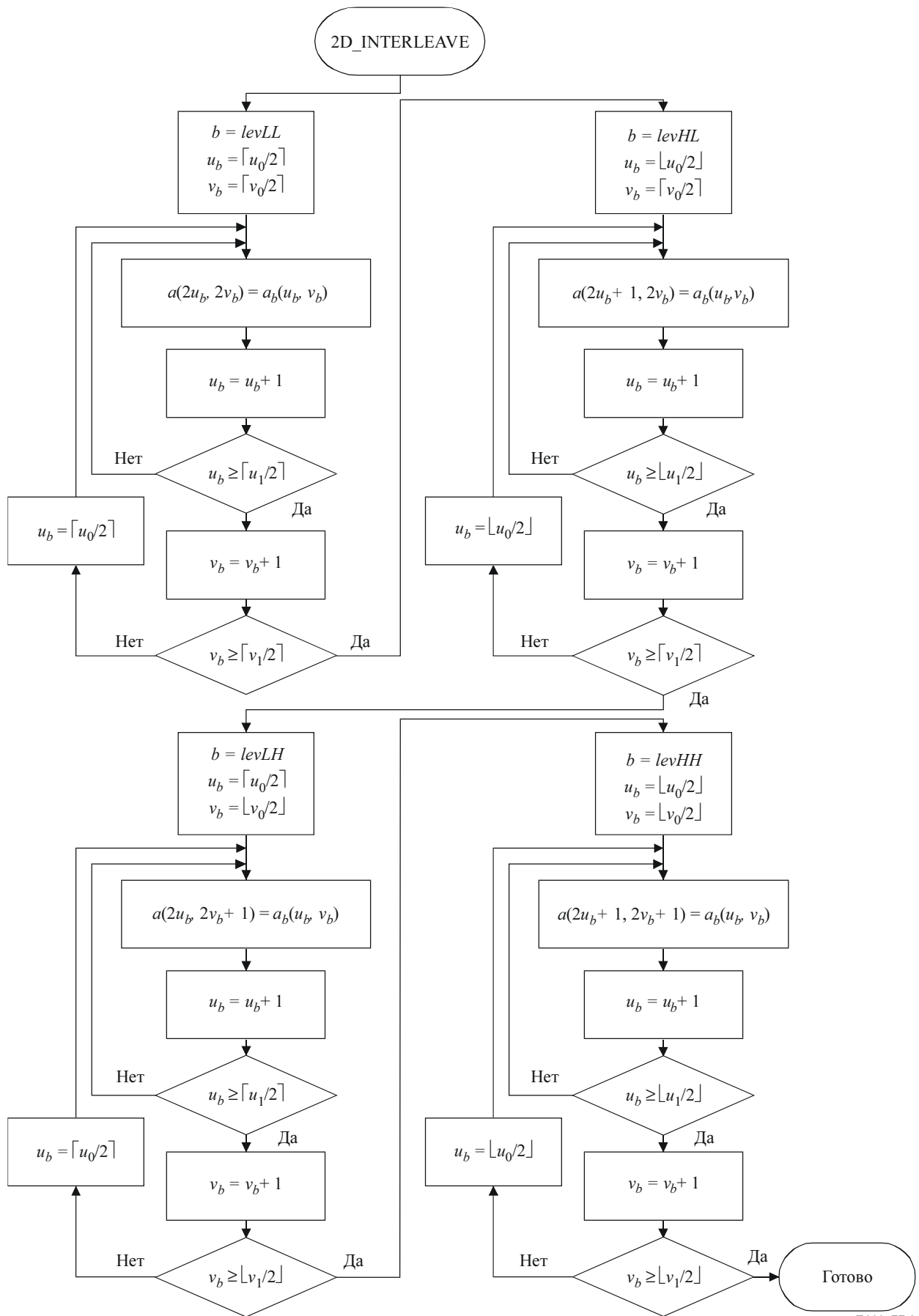


Рисунок F.7 – Параметры процедуры 2D_INTERLEAVE



T.800_FF-8

Рисунок F.8 – Процедура 2D_INTERLEAVE

F.3.4 Процедура HOR_SR

Процедура HOR_SR выполняет горизонтальное восстановление поддиапазона из двухмерного массива коэффициентов. Это требует на входе двухмерного массива $a(u, v)$ с горизонтальной и вертикальной протяженностью его аргументов $u_0 \leq u < u_1$ и $v_0 \leq v < v_1$ (см. рисунок F.9). На выходе создают версию с горизонтальной фильтрацией входного массива ряд за рядом.

Как показано на рисунке F.10, процедура HOR_SR применяет восстановление одномерного поддиапазона (процедурой 1D_SR) к каждому ряду v входного массива $a(u, v)$ и возвращает в каждый ряд сохраненный результат.

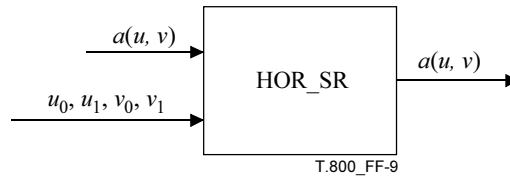


Рисунок F.9 – Входы и выходы процедуры HOR_SR

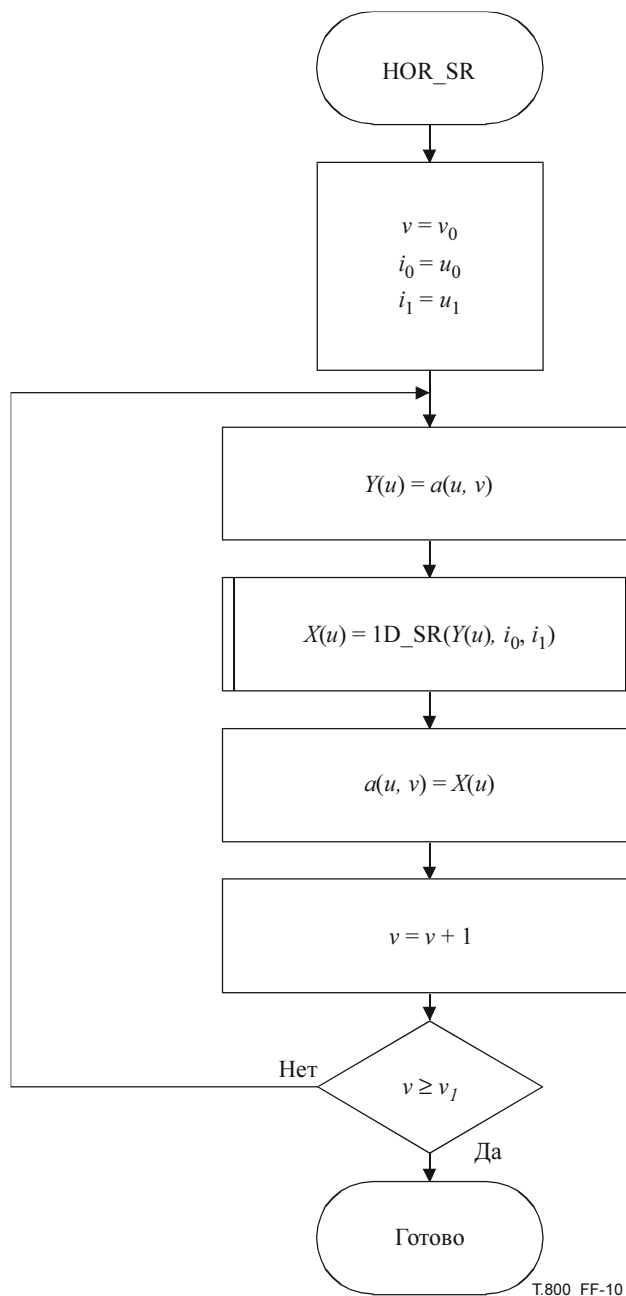


Рисунок F.10 – Процедура HOR_SR

F.3.5 Процедура VER_SR

Процедура VER_SR выполняет вертикальное восстановление поддиапазона из двухразмерного массива коэффициентов. Это требует на входе двухразмерного массива $a(u, v)$ с горизонтальной и вертикальной протяженностью его аргументов $u_0 \leq u < u_1$ и $v_0 \leq v < v_1$ (см. рисунок F.11). На выходе создают версию с вертикальной фильтрацией входного массива, столбец за столбцом.

Как показано на рисунке F.12, процедура VER_SR применяет восстановление одномерного поддиапазона (процедурой 1D_SR) к каждому столбцу u входного массива $a(u, v)$ и возвращает в каждый столбец сохраненный результат.

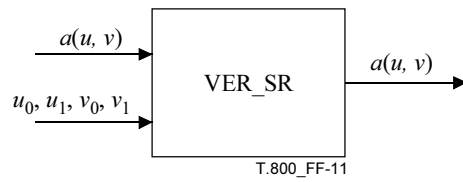


Рисунок F.11 – Входы и выходы процедуры VER_SR

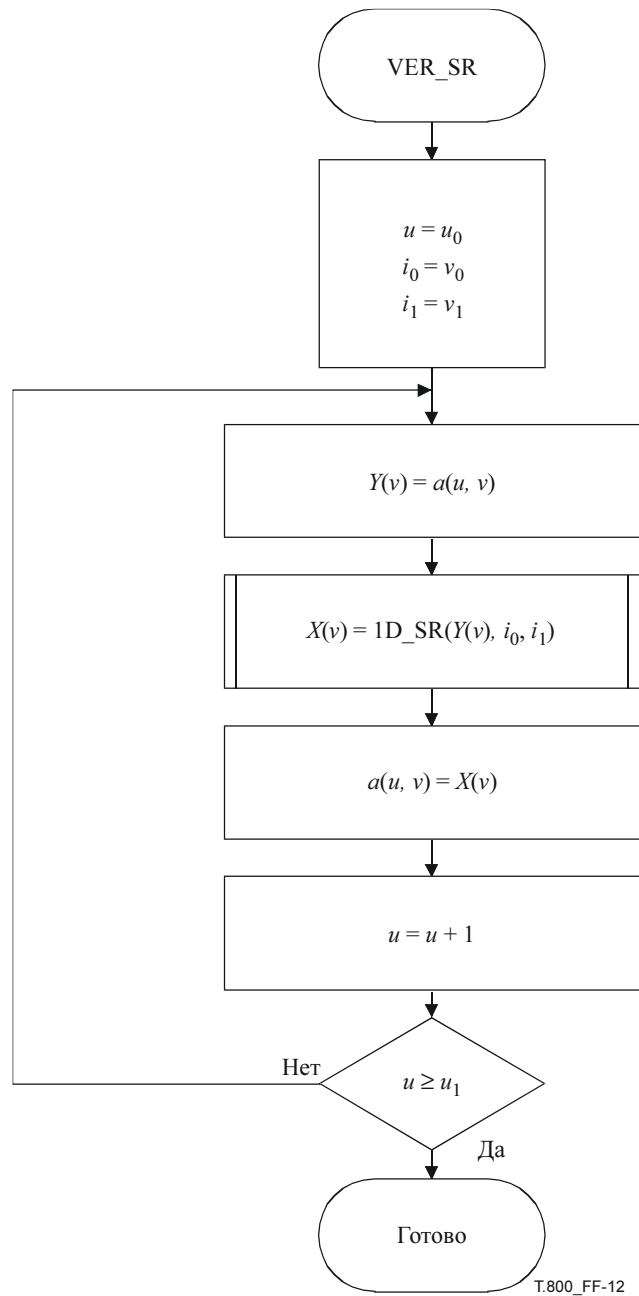


Рисунок F.12 – Процедура VER_SR

F.3.6 Процедура 1D_SR

Как показано на рисунке F.13, процедура 1D_SR требует на входе одномерного массива $Y(i)$ с протяженностью коэффициентов $i_0 \leq i < i_1$. На выходе создается массив с теми же самыми индексами (i_0, i_1) .

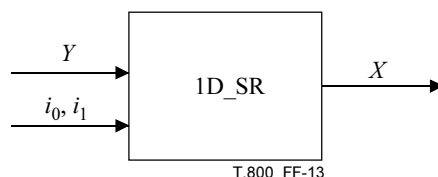


Рисунок F.13 – Параметры процедуры 1D_SR

Для сигналов единичной длины (то есть $i_0 = i_1 - 1$), процедура 1D_SR устанавливает значение от $X(i_0)$ до $X(i_1)$, если i_0 – это четное целое, и до $X(i_0) = Y(i_0)/2$, если i_0 – это нечетное целое.

Как показано на рисунке F.14, для сигналов с длиной большей или равной двум (т. е. $i_0 < i_1 - 1$) процедура 1D_SR сначала использует процедуру 1D_EXTR для расширения сигнала Y за левую и правую границы, т. е. до расширенного сигнала Y_{ext} , а затем процедуру 1D_FILTR, чтобы инверсно отфильтровать расширенный сигнал Y_{ext} и получить требуемый отфильтрованный сигнал X . Процедуры 1D_EXTR и 1D_FILTR зависят от выбора либо необратимого частотного преобразования (необратимого преобразования) 9-7, либо обратимого частотного преобразования (обратимого преобразования) 5-3. Об этом сигнализируют маркеры COD или COC (см. A.6.1 и A.6.2).

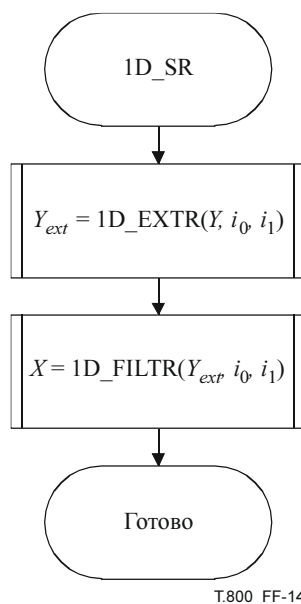


Рисунок F.14 – Процедура 1D_SR

F.3.7 Процедура 1D_SR

Как показано на рисунке F.15, процедура 1D_SR расширяет сигнал Y с помощью коэффициентов i_{left} влево и коэффициентов i_{right} вправо. Расширение сигнала необходимо для фильтрации сигнала на обеих границах.

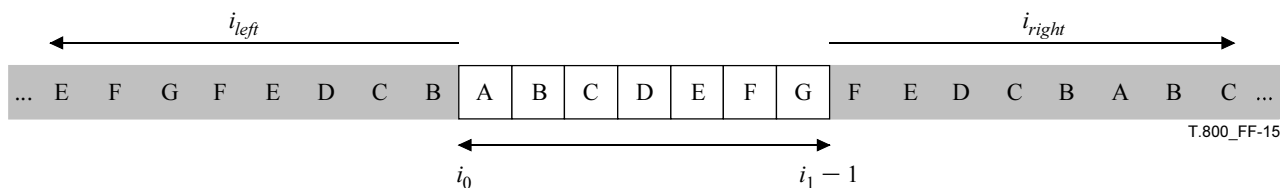


Рисунок F.15 – Периодическое симметричное расширение сигнала

Первым коэффициентом Y является коэффициент i_0 , а последним – коэффициент $i_1 - 1$. Эта процедура расширения известна как "Периодическое симметричное расширение". Симметричное расширение заключается в расширении сигнала с коэффициентами, которые получаются отражением сигнала сцентрированного на первом коэффициенте (коэффициенте i_0) для расширения сигнала влево и аналогичного расширения вправо от сигнала с последним коэффициентом (коэффициентом $i_1 - 1$). Периодическое симметричное расширение – это обобщение симметричного расширения на более общий случай, в котором число коэффициентов, с помощью которых расширяют сигнал в любую сторону, может превышать длину сигнала $i_1 - i_0$. Именно этот случай имеет место при высоких уровнях декомпозиции.

С помощью процедуры 1D_EXTR вычисляют значения $Y_{ext}(i)$ при аргументах i за границей диапазона $i_0 \leq i < i_1$, как это показано равенством (F-3):

$$Y_{ext}(i) = Y(PSE_O(i, i_0, i_1)), \tag{F-3}$$

где $PSE_O(i, i_0, i_1)$ задано равенством (F-4):

$$PSE_O(i, i_0, i_1) = i_0 + \min(\text{mod}(i - i_0, 2(i_1 - i_0 - 1)), 2(i_1 - i_0 - 1) - \text{mod}(i - i_0, 2(i_1 - i_0 - 1))). \tag{F-4}$$

Две процедуры расширения определяют в зависимости от вида преобразования: частотного преобразования 5-3 (процедура 1D_EXTR₅₋₃) или частотного преобразования 9-7 (процедура 1D_EXTR₉₋₇). Эти процедуры отличаются только минимальными значениями параметров расширения ($i_{left5-3}$ и $i_{right5-3}$ для преобразования 5-3 и $i_{left9-7}$ и $i_{right9-7}$ – для преобразования 9-7), которые приведены в таблицах F.2 и F.3 в зависимости от четности индексов i_0 и i_1 . Значения, большие или равные тем, которые приведены в таблицах F.2 и F.3, создают идентичный массив X на выходе процедуры 1D_IFILTR на рисунке F.14.

Таблица F.2 – Расширение влево

i_0	$i_{left5-3}$	$i_{left9-7}$
четное	1	3
нечетное	2	4

Таблица F.3 – Расширение вправо

i_1	$i_{right5-3}$	$i_{right9-7}$
нечетное	1	3
четное	2	4

F.3.8 Процедура 1D_FILTR

Обратимая процедура фильтрации 1D_FILTR_{5-3R} и необратимая процедура фильтрации 1D_FILTR_{9-7I} установлены в зависимости от того, используется ли обратимое преобразование 5-3 или необратимое 9-7.

Как показано на рисунке F.16, обе процедуры требуют на входе расширенного 1D сигнала Y_{ext} с индексом первого коэффициента i_0 и индексом коэффициента i_1 , следующего непосредственно за последним коэффициентом ($i_1 - 1$). Обе процедуры создают на выходе сигнал X .



Рисунок F.16 – Параметры процедуры 1D_FILTR

ИСО/МЭК 15444-1:2002 (R)

Обе процедуры используют фильтрацию с подъемом основания, которое заключается в обработке сигнала последовательностью очень простых операций фильтрации, названных ступенями подъема. Эти ступени попеременно модифицируют коэффициенты с нечетными индексами взвешенной суммой коэффициентов с четными индексами, а коэффициенты с четными индексами – взвешенной суммой коэффициентов с нечетными индексами.

Ф.3.8.1 Процедура 1D_FILTR_{5-3R}

Процедура 1D_FILTR_{5-3R} использует фильтрацию с подъемом основания совместно с операциями округления. Равенство (F-5) первым преобразовывает все обозначенные значения n , а следующее равенство (F-6) использует значения, вычисленные из равенства (F-5):

$$X(2n) = Y_{ext}(2n) - \left\lfloor \frac{Y_{ext}(2n-1) + Y_{ext}(2n+1) + 2}{4} \right\rfloor \quad \text{для} \quad \left\lfloor \frac{i_0}{2} \right\rfloor \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 1, \quad (F-5)$$

$$X(2n+1) = Y_{ext}(2n+1) + \left\lfloor \frac{X(2n) + X(2n+2)}{2} \right\rfloor \quad \text{для} \quad \left\lfloor \frac{i_0}{2} \right\rfloor \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor. \quad (F-6)$$

Значения $X(k)$ таковы, что $i_0 \leq k < i_1$ на выходе процедуры 1D_FILTR_{5-3R}.

Ф.3.8.2 Процедура 1D_FILTR_{9-7I}

Процедура 1D-FILTR_{9-7I} использует фильтрацию с подъемом основания (без операции округления). Параметры подъема (α , β , γ , δ) и параметр масштабирования K для всех ступеней фильтрации определены в Ф.3.8.2.1.

Равенство (F-7) описывает две шага масштабирования (1 и 2) и четыре шага (с 3 по 6) одномерной (1D) фильтрации, которую выполняют над расширенным сигналом $Y_{ext}(n)$, чтобы получить коэффициенты $i_l - i_0$ сигнала X . Эти шаги выполняют в следующем порядке.

Первый, шаг 1, выполняют над всеми величинами n такими, что $\left\lfloor \frac{i_0}{2} \right\rfloor - 1 \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 2$, а шаг 2 – над всеми

величинами n такими, что $\left\lfloor \frac{i_0}{2} \right\rfloor - 2 \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 2$.

Далее шаг 3 выполняют над всеми величинами n такими, что $\left\lfloor \frac{i_0}{2} \right\rfloor - 1 \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 2$, и используют величины, полученные при вычислении шагов 1 и 2.

Затем шаг 4 выполняют над всеми величинами n такими, что $\left\lfloor \frac{i_0}{2} \right\rfloor - 1 \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 1$, и используют величины, полученные при вычислении шагов 2 и 3.

Далее шаг 5 выполняют над всеми величинами n такими, что $\left\lfloor \frac{i_0}{2} \right\rfloor \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor + 1$, и используют величины, полученные при вычислении шагов 3 и 4.

Наконец шаг 6 выполняют над всеми величинами n такими, что $\left\lfloor \frac{i_0}{2} \right\rfloor \leq n < \left\lfloor \frac{i_1}{2} \right\rfloor$, и используют величины, полученные при вычислении шагов 4 и 5.

$$\left\{ \begin{array}{ll} X(2n) = KY_{ext}(2n) & [\text{ШАГ1}] \\ X(2n+1) = (1/K)Y_{ext}(2n+1) & [\text{ШАГ2}] \\ X(2n) = X(2n) - \delta(X(2n-1) + X(2n+1)) & [\text{ШАГ3}] \\ X(2n+1) = X(2n+1) - \gamma(X(2n) + X(2n+2)) & [\text{ШАГ4}] \\ X(2n) = X(2n) - \beta(X(2n-1) + X(2n+1)) & [\text{ШАГ5}] \\ X(2n+1) = X(2n+1) - \alpha(X(2n) + X(2n+2)) & [\text{ШАГ6}] \end{array} \right. \quad (F-7)$$

где значения параметров фильтрации (α , β , γ , δ) и K приведены в таблице F.4.

Таблица F.4 – Определение параметров фильтрации для необратимого фильтра 9-7

Параметр	Точное выражение	Аппроксимация
α	$-g_4 g_3$	-1,586 134 342 059 924
β	$g_3 r_1$	-0,052 980 118 572 961
γ	$r_1 s_0$	0,882 911 075 530 934
δ	$s_0 t_0$	0,443 506 852 043 971
K	$I t_0$	1,230 174 104 914 001

Значения $X(k)$ в диапазоне $i_0 \leq k < i_1$ формируют выход процедуры 1D_FILTR₁.

F.3.8.2.1 Параметры фильтрации для процедуры 1D_FILTR_{9,71}

Параметры фильтрации (α , β , γ , δ , K) определены в таблице F.4 в выражениях параметров g_n из таблицы F.5 и параметров (r_0 , r_1 , s_0 , t_0) из таблицы F.6. Параметры g_n определены в выражениях параметров x_1 , $\Re x_2$ и $|x_2|^2$, данных в таблице F.7. Во всех таблицах выражения для всех параметров даны в замкнутой форме, включая аппроксимации до 15 десятичных знаков.

Таблица F.5 – Определение коэффициентов g_n

n	Коэффициенты g_n	Аппроксимация g_n
0	$5x_1(48 x_2 ^2 - 16\Re x_2 + 3) / 32$	-0,602 949 018 236 360
1	$-5x_1(8 x_2 ^2 - \Re x_2) / 8$	0,266 864 118 442 875
2	$5x_1(4 x_2 ^2 + 4\Re x_2 - 1) / 16$	0,078 223 266 528 990
3	$-5x_1(\Re x_2) / 8$	-0,016 864 118 442 875
4	$5x_1 / 64$	-0,026 748 757 410 810

Таблица F.6 – Промежуточные выражения (r_0 , r_1 , s_0 , t_0)

Параметр	Точное выражение	Аппроксимация
r_0	$-g_0 + 2g_1g_4 g_3$	1,449 513 704 087 943
r_1	$-g_2 + g_4 + g_1g_4 g_3$	0,318 310 318 985 991
s_0	$g_1 - g_3 - g_3r_0 r_1$	0,360 523 644 801 462
t_0	$r_0 - 2r_1$	0,812 893 066 115 961

Таблица F.7 – Промежуточные выражения

Параметр	Точное выражение	Аппроксимация
A	$\sqrt[3]{\frac{63 - 14\sqrt{15}}{1080\sqrt{15}}}$	0,128 030 244 703 494
B	$-\sqrt[3]{\frac{63 + 14\sqrt{15}}{1080\sqrt{15}}}$	-0,303 747 672 895 197
x_1	$A + B - 1 \mid 6$	-0,342 384 094 858 369
$\Re x_2$	$-\frac{(A + B)}{2} - \frac{1}{6}$	-0,078 807 952 570 815
$ x_2 ^2$	$\left[\frac{(A + B)}{2} + \frac{1}{6}\right]^2 + \frac{3(A - B)^2}{4}$	0,146 034 820 982 800

F.4 Прямое преобразование (информативное)

F.4.1 Процедура FDWT (информативное)

Прямое дискретное частотное преобразование (FDWT) преобразует сдвиг уровня постоянного тока (DC) образцов компонентов тайла $I(x, y)$ в набор поддиапазонов с коэффициентами $a_b(u_b, v_b)$ (процедура FDWT). Процедура FDWT (см. рисунок F.17) также принимает на входе число уровней декомпозиции N_L , о которых сообщается в маркерах COD или COC (см. A.6.1 и A.6.2).

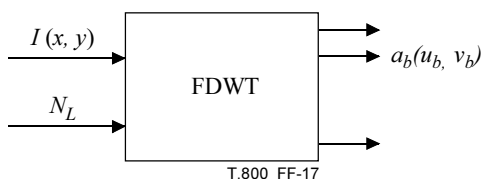


Рисунок F.17 – Входы и выходы процедуры FDWT

Как показано на рисунке F.18, в этом случае все поддиапазоны с $N_L=2$ могут быть представлены следующим образом:

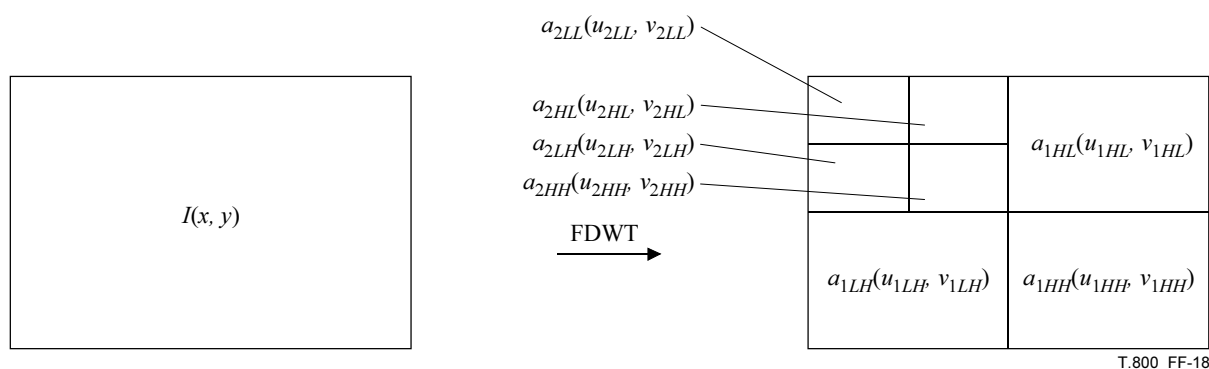


Рисунок F.18 – FDWT ($N_L = 2$)

Процедуру FDWT запускают инициализацией переменной lev (текущий уровень декомпозиции) на нуль и установкой поддиапазона $a_{0LL}(u_{0LL}, v_{0LL})$ на входной массив $I(u, v)$. Процедуру 2D_SD выполняют для каждого уровня lev , причем уровень lev возрастает на единицу с каждой итерацией до тех пор, пока не будет выполнено N_L итераций. Процедуру 2D_SD повторяют над поддиапазоном $levLL$, который создается при каждой итерации.

Как определено в Приложении В (см. равенство (B-15)), координаты поддиапазона $a_{levLL}(u, v)$ расположены на отрезке, определенном как:

$$tbx_0 \leq u < tbx_1 \quad u \quad tby_0 \leq v < tby_1 \quad (F-8)$$

Процедура FDWT показана на рисунке F.19.

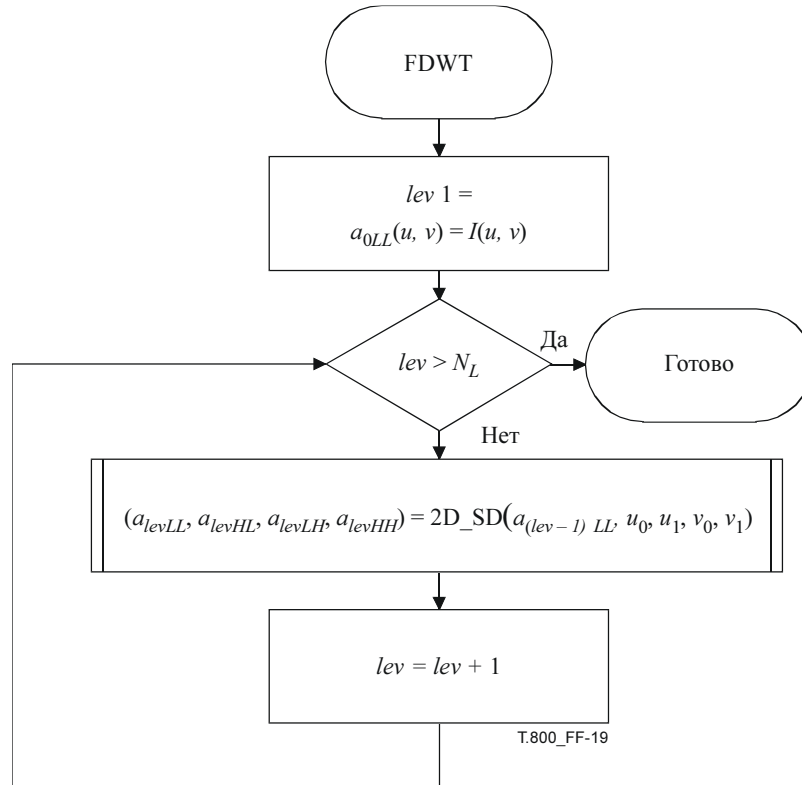


Рисунок F.19 – Процедура FDWT

F.4.2 Процедура 2D_SD (информативное)

Процедура 2D_SD выполняет декомпозицию двухразмерного массива коэффициентов или образцов $a_{(lev-1)LL}(u, v)$ в четыре группы коэффициентов поддиапазонов $a_{levLL}(u, v)$, $a_{levHL}(u, v)$, $a_{levLH}(u, v)$ и $a_{levHH}(u, v)$.

Общее число коэффициентов поддиапазона lev_{LL} равно сумме общего числа коэффициентов четырех поддиапазонов в результате действия процедуры 2D_SD.

На рисунке F.20 описаны входные и выходные параметры процедуры 2D_SD.

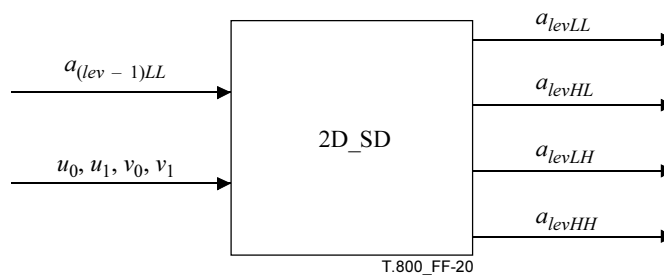


Рисунок F.20 – Входы и выходы процедуры 2D_SD

Рисунок F.21 иллюстрирует декомпозицию поддиапазона, которую выполняет процедура 2D_SD.

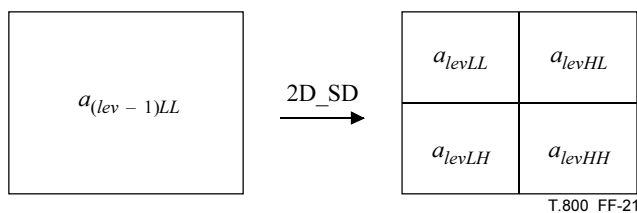


Рисунок F.21 – Один уровень декомпозиции в четырех поддиапазонах (процедура 2D_SD)

Процедура 2D_SD использует сначала процедуру VER_SD для всех столбцов $a(u, v)$, а затем использует процедуру HOR_SD для всех рядов $a(u, v)$. Полученные таким образом из $a(u, v)$ коэффициенты разбивают на четыре поддиапазона, используя процедуру 2D_DEINTERLEAVE.

На рисунке F.22 показана процедура 2D_SD.

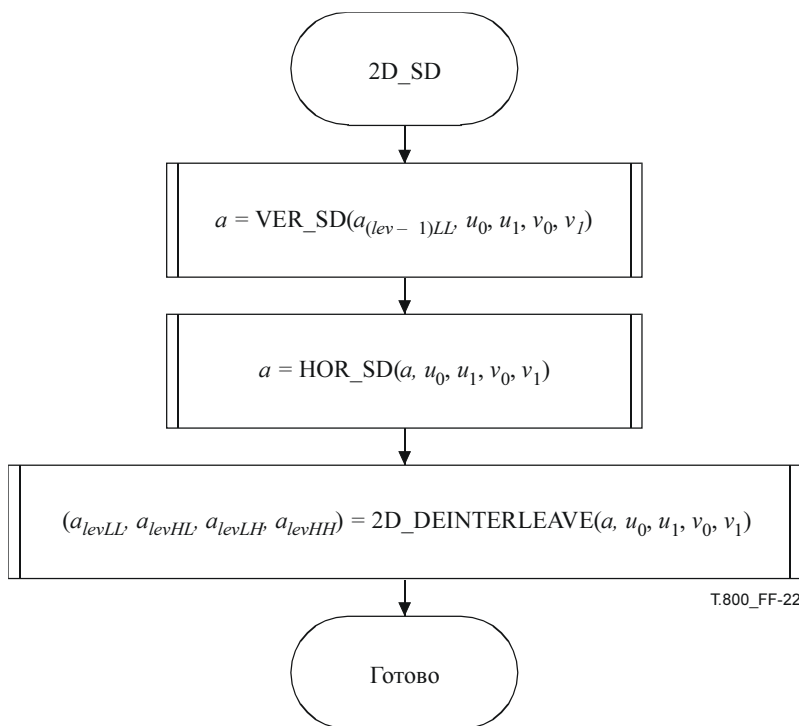


Рисунок F.22 – Процедура 2D_SD

F.4.3 Процедура VER_SD (информативное)

Процедура VER_SD выполняет вертикальную декомпозицию поддиапазона с двухразмерным массивом коэффициентов. Это требует на входе двухразмерного массива $a_{(lev-1)LL}(u, v)$, горизонтального и вертикального расширения коэффициентов, определенного как $u_0 \leq u < u_1$ и $v_0 \leq v < v_1$ (см. рисунок F.23), и обеспечивает на выходе отфильтрованный по вертикали столбец за столбцом вариант входного массива $a(u, v)$. Используемые процедурой VER_SD значения u_0, u_1, v_0, v_1 являются теми значениями $tbx_0, tbx_1, tby_0, tby_1$, которые соответствуют поддиапазону $b = (lev - 1)LL$ (см. определение в равенстве (B-15)).

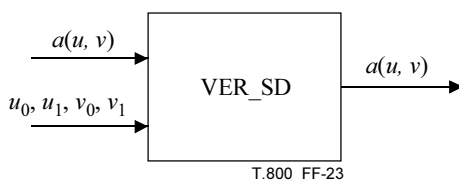


Рисунок F.23 – Входы и выходы процедуры VER_SD

Как показано на рисунке F.24, процедура VER_SD использует одномерную декомпозицию поддиапазона (процедура1D_SD) для каждого столбца входного массива $a(u, v)$ и сохраняет результат в том же столбце.

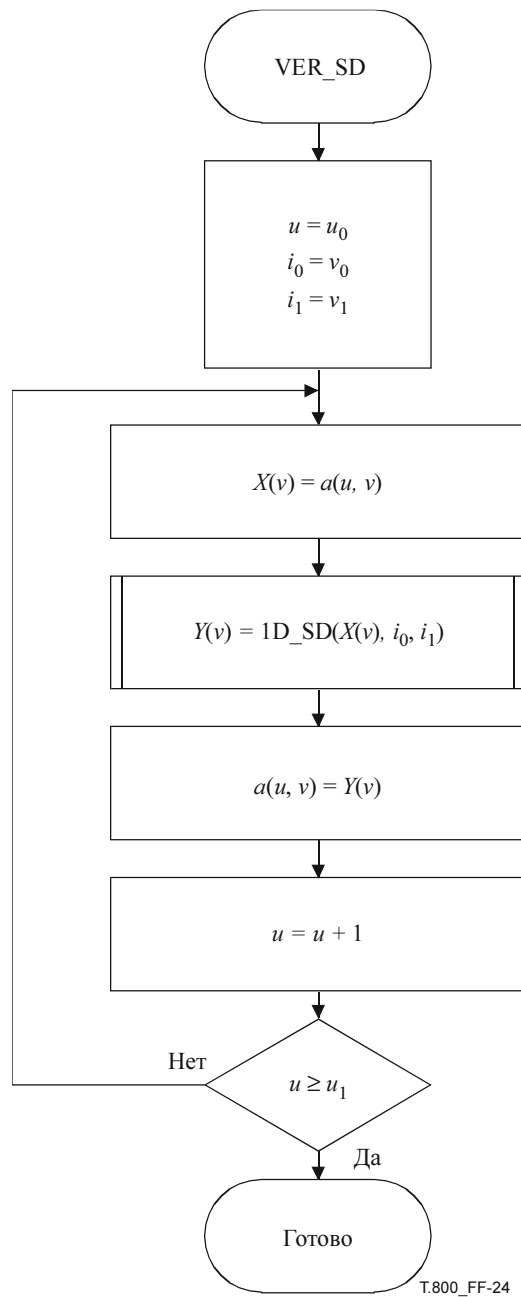


Рисунок F.24 – Процедура VER_SD

F.4.4 Процедура HOR_SD (информативное)

Процедура HOR_SD выполняет горизонтальную декомпозицию поддиапазона с двухразмерным массивом коэффициентов. Это требует на входе двухразмерного массива $a(u, v)$ с горизонтальным и вертикальным расширением коэффициентов, определенным как $u_0 \leq u < u_1$ и $v_0 \leq v < v_1$ (см. рисунок F.25), и обеспечивает на выходе отфильтрованный по горизонтали ряд за рядом вариант входного массива.

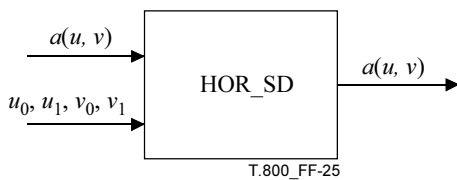


Рисунок F.25 – Входы и выходы процедуры HOR_SD

Как показано на рисунке F.26, процедура HOR_SD использует одномерную декомпозицию поддиапазона (процедура 1D_SD) для каждого ряда входного массива $a(u, v)$ и сохраняет результат в том же ряду.

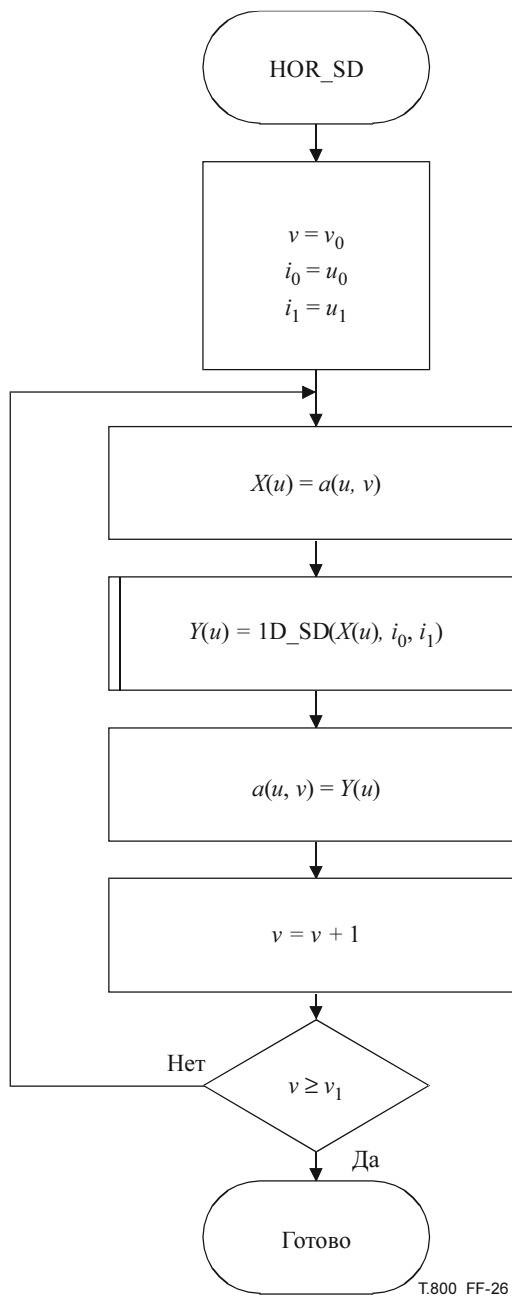


Рисунок F.26 – Процедура HOR_SD

F.4.5 Процедура 2D_DEINTERLEAVE (информативное)

Как показано на рисунке F.27, процедура 2D_DEINTERLEAVE разбивает коэффициенты $a(u, v)$ на четыре поддиапазона. Распределение зависит от координат (u_0, v_0) первого коэффициента (u, v) .

Способ, с помощью которого эти поддиапазоны формируют из выходных данных $a(u, v)$ процедуры HOR_SD, описан процедурой 2D_DEINTERLEAVE, как показано на рисунке F.28.

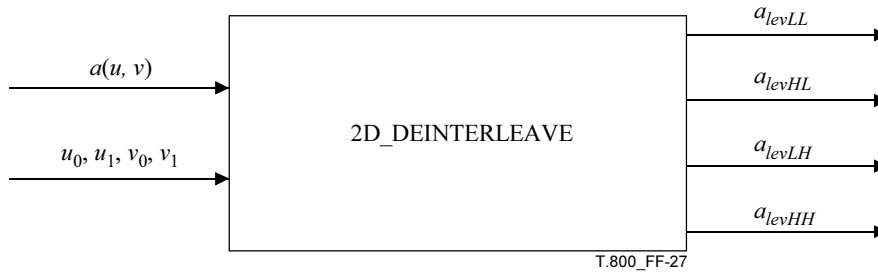
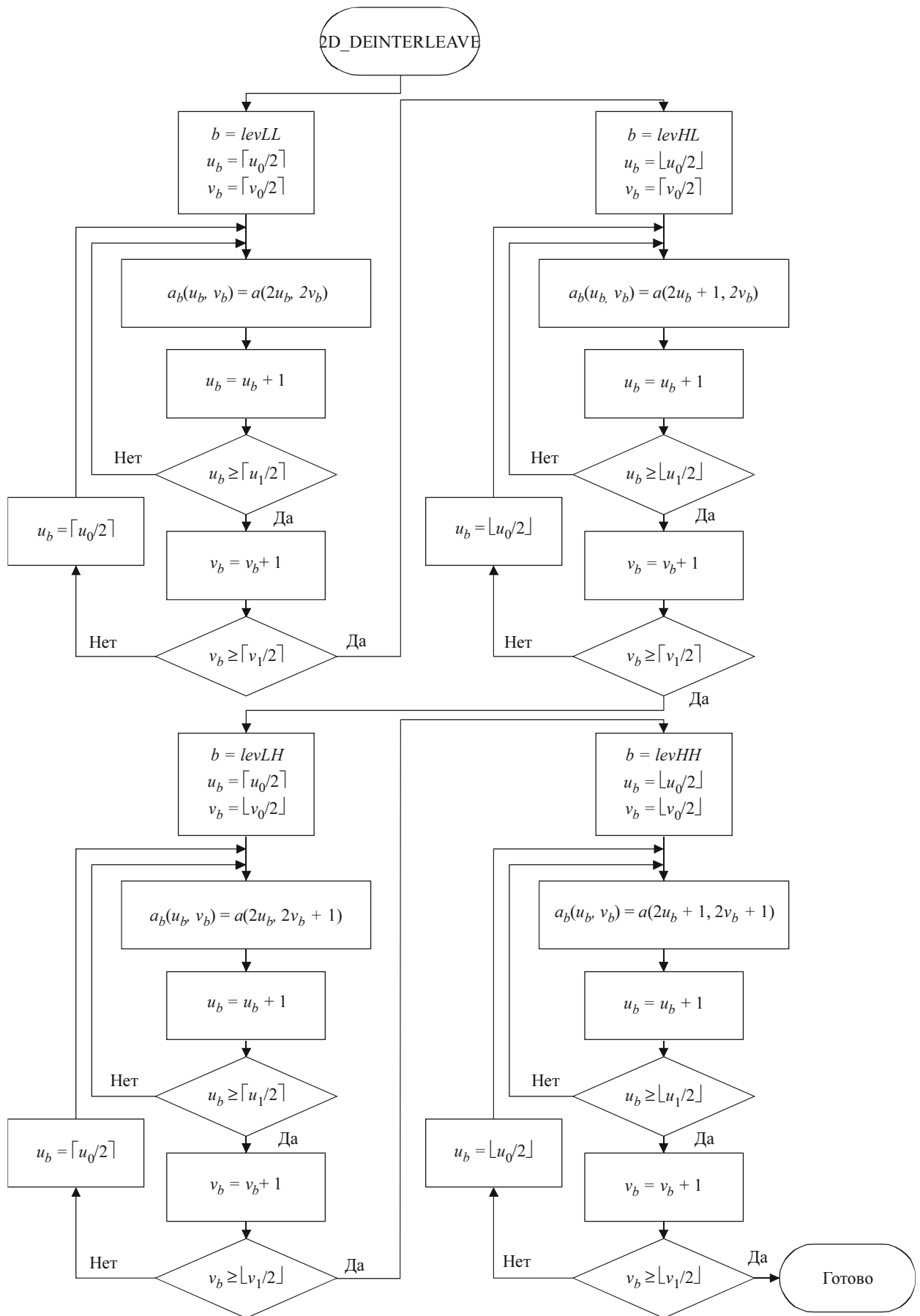


Рисунок F.27 – Параметры процедуры 2D_DEINTERLEAVE



T.800_FF-28

Рисунок F.28 – Процедура 2D_DEINTERLEAVE

F.4.6 Процедура 1D_SD (информативное)

Как показано на рисунке F.29, процедура 1D_SD требует на входе одномерного массива $X(i)$ с расширением коэффициентов, определенным как $i_0 \leq i < i_1$, и создает на выходе массив $Y(i)$ с теми же индексами (i_0, i_1).

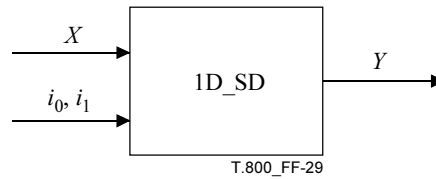


Рисунок F.29 – Параметры процедуры 1D_SD

Для сигналов единичной длины (то есть $i_0 = i_1 - 1$) процедура 1D_SD создает набор значений от $Y(i_0)$ до $Y(i_0) = X(i_0)$, если i_0 – четное целое, и до $Y(i_0) = 2X(i_0)$, если i_0 – нечетное целое.

Как показано на рисунке F.30, для сигналов с длиной большей или равной двум (т. е. $i_0 < i_1 - 1$) процедура 1D_SD сначала использует процедуру 1D_EXTD для расширения сигнала X за левую и правую границы, что приводит к расширению сигнала X_{ext} , а затем процедуру 1D_FILTD для фильтрации расширенного сигнала X_{ext} и создания требуемого отфильтрованного сигнала Y .

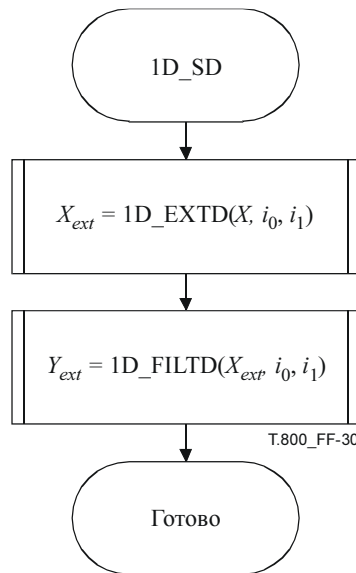


Рисунок F.30 – Процедура 1D_SD

F.4.7 Процедура 1D_EXTD (информативное)

Процедура 1D_EXTD идентична процедуре 1D_EXTR, за исключением значений параметров $i_{left_{9,7}}$, $i_{right_{9,7}}$, $i_{left_{5,3}}$ и $i_{right_{5,3}}$, которые показаны в таблицах F.8 и F.9.

Таблица F.8 – Расширение влево

i_0	$i_{left_{5,3}}$	$i_{left_{9,7}}$
четное	2	4
нечетное	1	3

Таблица F.9 – Расширение вправо

i_l	$i_{right_{5-3}}$	$i_{right_{9-7}}$
нечетное	2	4
четное	1	3

F.4.8 Процедура 1D_FILTD (информативное)

Данная Рекомендация | Международный стандарт определяет одну необратимую процедуру фильтрации (1D_FILTD₉₋₇) и одну обратимую (1D_FILTD_{5-3R}) в зависимости от того, выбрано ли необратимое преобразование 9-7 или обратимое 5-3.

Как показано на рисунке F.31, обе процедуры используют в качестве входного расширенный одномерный (1D) сигнал X_{ext} с индексом первого коэффициента i_0 и индексом коэффициента i_l , который непосредственно следует за последним коэффициентом ($i_l - 1$). Обе процедуры создают выходной сигнал Y . Коэффициенты сигнала Y с четными индексами являются низкочастотным вариантом расширенного сигнала X_{ext} , а с нечетными индексами – высокочастотным вариантом расширенного сигнала X_{ext} .

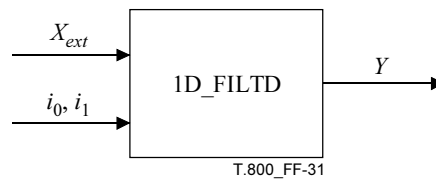


Рисунок F.31 – Параметры процедуры 1D_FILTD

F.4.8.1 Процедура 1D_FILTD_{5-3R} (информативное)

Описанное в этом разделе обратимое преобразование является применением фильтрации с подъемом основания для обратимого частотного фильтра 5-3. Обратимое преобразование определено с использованием фильтрации с подъемом основания. Коэффициенты выходного сигнала Y с нечетными индексами вычисляются сначала для всех величин n таких, что $\left\lceil \frac{i_0}{2} \right\rceil - 1 \leq n < \left\lceil \frac{i_l}{2} \right\rceil$, как дано в равенстве (F-9):

$$Y(2n + 1) = X_{ext}(2n + 1) - \left[\frac{X_{ext}(2n) + X_{ext}(2n + 2)}{4} \right]. \tag{F-9}$$

Затем вычисляются коэффициенты выходного сигнала Y с четными индексами из значений расширенного сигнала X_{ext} с четными индексами и коэффициентов сигнала Y с нечетными индексами для всех величин n таких, что $\left\lceil \frac{i_0}{2} \right\rceil \leq n < \left\lceil \frac{i_l}{2} \right\rceil$ как это показано в равенстве (F-10):

$$Y(2n) = X_{ext}(2n) + \left[\frac{Y(2n - 1) + Y(2n + 1) + 2}{4} \right]. \tag{F-10}$$

Значения $Y(k)$ с $i_0 \leq k < i_l$ формируют выход процедуры 1D_FILTD_R.

F.4.8.2 Процедура 1D_FILTDI (информативное)

Описанное в этом разделе необратимое преобразование является применением фильтрации DWT с подъемом основания для необратимого частотного фильтра 9-7.

Равенство (F-11) описывает четыре шага подъема (с 1 по 4) и два шага масштабирования (5 и 6) одномерной (1D) фильтрации, выполняемой над расширенным сигналом $X_{ext}(n)$ для вычисления коэффициентов $i_1 - i_0$ сигнала Y . Эти шаги выполняют в следующем порядке.

Сначала выполняют шаг 1 для всех значений n таких, что $\left\lceil \frac{i_0}{2} \right\rceil - 2 \leq n < \left\lceil \frac{i_1}{2} \right\rceil + 1$.

Затем выполняют шаг 2 для всех значений n таких, что $\left\lceil \frac{i_0}{2} \right\rceil - 1 \leq n < \left\lceil \frac{i_1}{2} \right\rceil + 1$, и используют величины, полученные при вычислении шага 1.

Затем выполняют шаг 3 для всех значений n таких, что $\left\lceil \frac{i_0}{2} \right\rceil - 1 \leq n < \left\lceil \frac{i_1}{2} \right\rceil$, и используют величины, полученные при вычислении шагов 1 и 2.

Затем выполняют шаг 4 для всех значений n таких, что $\left\lceil \frac{i_0}{2} \right\rceil \leq n < \left\lceil \frac{i_1}{2} \right\rceil$, и используют величины, полученные при вычислении шагов 2 и 3.

Наконец, выполняют шаг 5 для всех значений n таких, что $\left\lceil \frac{i_0}{2} \right\rceil \leq n < \left\lceil \frac{i_1}{2} \right\rceil$, и используют величины, полученные при вычислении шага 3, а шаг выполняют 6 для всех значений n таких, что $\left\lceil \frac{i_0}{2} \right\rceil \leq n < \left\lceil \frac{i_1}{2} \right\rceil$, и используют величины, полученные при вычислении шага 4.

$$\left\{ \begin{array}{l} Y(2n+1) = X_{ext}(2n+1) + \alpha(X_{ext}(2n) + X_{ext}(2n+2)) \quad [\text{ШАГ1}] \\ Y(2n) = X_{ext}(2n) + \beta(Y(2n-1) + Y(2n+1)) \quad [\text{ШАГ2}] \\ Y(2n+1) = Y(2n+1) + \gamma(Y(2n) + Y(2n+2)) \quad [\text{ШАГ3}] \\ Y(2n) = Y(2n) + \delta(Y(2n-1) + Y(2n+1)) \quad [\text{ШАГ4}] \\ Y(2n+1) = KY(2n+1) \quad [\text{ШАГ5}] \\ Y(2n) = (1/K)Y(2n) \quad [\text{ШАГ6}] \end{array} \right. \quad (\text{F-11})$$

где параметры подъема α , β , γ , δ , и K показаны в таблице F.4.

Для значений $i_0 \leq k < i_1$ формируют выходные данные процедуры 1D_FILTD_r.

Приложение G

Сдвиг уровня DC и многократные преобразования компонента

(Это Приложение составляет неотъемлемую часть данной Рекомендации | Международного стандарта)

В этом Приложении и во всех его подразделах блок-схемы и таблицы являются нормативными (обязательными) только в том смысле, что они определяют выходные данные, которые при альтернативных применениях должны повторяться.

Это Приложение определяет сдвиг уровня DC, который преобразует величины со знаком (положительным или отрицательным), полученные в результате процесса декодирования, в соответствующие восстановленные образцы.

В Приложении описаны два различных метода сложного преобразования компонента. Эти многократные преобразования компонента используют для улучшения эффективности сжатия. Методы не относятся к тем сложным преобразованиям компонента, которые используют, чтобы отобразить цветные величины для вывода изображения на экран. Одно сложное преобразование компонента является обратимым и может быть использовано для кодирования с потерями или без потерь. Другое преобразование – необратимое – может быть использовано только для кодирования с потерями.

G.1 Сдвиг уровня DC компонентов тайла

На рисунке G.1 показан процесс сдвига уровня DC в системе со сложным преобразованием компонента.

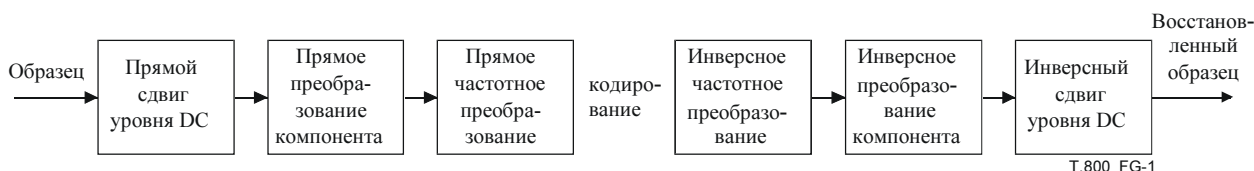


Рисунок G.1 – Размещение сдвига уровня DC с преобразованием компонента

На рисунке G.2 показан процесс сдвига уровня DC в системе без сложного преобразования компонента.

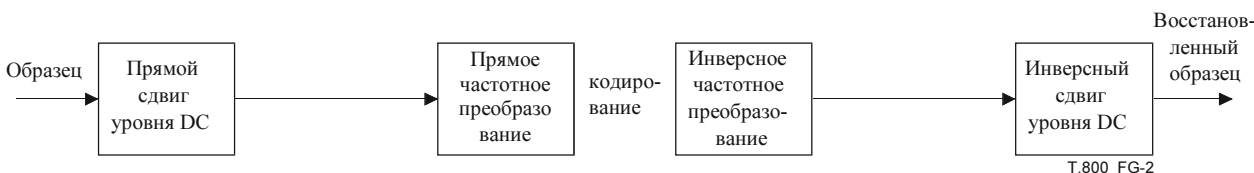


Рисунок G.2 – Размещение сдвига уровня DC без преобразования компонента

G.1.1 Сдвиг уровня DC компонентов тайла (информативное)

Сдвиг уровня DC выполняют только на образцах компонентов без знака. Сдвиг предшествует вычислению прямого сложного преобразования компонента (RCT или ICT), если такое преобразование используют. В противном случае сдвиг выполняют перед частотным преобразованием, которое описано в Приложении F. Если значением MSB является $Ssiz^i$ (из сегмента маркера SIZ, см. A.5.1), равное нулю, то происходит сдвиг уровня во все образцах $I(x, y)$ i -го компонента вычитанием одной и той же величины из каждого образца:

$$I(x, y) \leftarrow I(x, y) - 2^{Ssiz^i}. \quad (G-1)$$

G.1.2 Инверсный сдвиг уровня DC компонентов тайла

Инверсный сдвиг уровня DC выполняют только на восстановленных образцах компонентов без знака. Сдвиг совершают после вычисления сложного инверсного преобразования компонента (RCT или ICT), если такое преобразование используют. В противном случае сдвиг выполняют после инверсного частотного преобразования, которое описано в Приложении F. Если значением MSB является $Ssiz^i$ (из сегмента маркера SIZ, см. A.6.1), равное нулю, то происходит сдвиг уровня во все образцах $I(x, y)$ i -го компонента добавлением одной и той же величины к каждому образцу:

$$I(x, y) \leftarrow I(x, y) + 2^{Ssiz^i}. \quad (G-2)$$

ПРИМЕЧАНИЕ. – В результате действий квантования восстановленные образцы $I(x, y)$ могут превысить динамический диапазон первоначальных образцов. Не существует нормативной процедуры для состояний переполнения или недостаточного заполнения. Несмотря на это, характерным решением является ограничение значения до ближайшего внутриверноначального динамического диапазона.

G.2 Обратимое сложное преобразование компонента (RCT)

Об использовании обратимого сложного преобразования компонента сообщают в сегменте маркера COD (см. А.6.1). Следует использовать RCT только с обратимым фильтром 5-3. Преобразование RCT является декоррелирующим преобразованием, которое применяют к первым трем компонентам изображения (пронумерованным как 0, 1 и 2). Эти три компонента входят в RCT и должны иметь одинаковое разделение на опорной сетке и одинаковую битовую глубину.

ПРИМЕЧАНИЕ. – Несмотря на то, что обратимое преобразование RCT предусматривает сжатие без потерь, оно может быть использовано в усеченных кодированных потоках для сжатия с потерями.

G.2.1 Прямое преобразование RCT (информативное)

Перед выполнением прямого преобразования RCT образцы компонентов изображения без знака сдвигают по уровню DC.

Прямое преобразование RCT применяют к компонентам $I_0(x, y)$, $I_1(x, y)$, $I_2(x, y)$:

$$Y_0(x, y) = \left\lfloor \frac{I_0(x, y) + 2I_1(x, y) + I_2(x, y)}{4} \right\rfloor, \quad (G-3)$$

$$Y_1(x, y) = I_2(x, y) - I_1(x, y), \quad (G-4)$$

$$Y_2(x, y) = I_0(x, y) - I_1(x, y). \quad (G-5)$$

Если компоненты I_0 , I_1 , и I_2 нормированы с одинаковой точностью, то равенства (G-4) и (G-5) приводят к числовой точности Y_1 и Y_2 , то есть на один бит больше, чем точность первоначальных компонентов. Такое повышение в точности необходимо для того, чтобы обеспечить обратимость.

G.2.2 Инверсное преобразование RCT

После выполнения описанного в Приложении F инверсного частотного преобразования применяют следующее инверсное преобразование RCT:

$$I_1(x, y) = Y_0(x, y) - \left\lfloor \frac{Y_2(x, y) + Y_1(x, y)}{4} \right\rfloor, \quad (G-6)$$

$$I_0(x, y) = Y_2(x, y) + I_1(x, y), \quad (G-7)$$

$$I_2(x, y) = Y_1(x, y) + I_1(x, y). \quad (G-8)$$

После того как выполнено инверсное преобразование RCT, над компонентами изображения без знака производят инверсный сдвиг по уровню DC.

G.3 Обратимое сложное преобразование компонента (ICT)

Этот раздел определяет обратимое сложное преобразование компонента. Об использовании обратимого сложного преобразования компонента сообщают в сегменте маркера COD (см. А.6.1). Следует использовать ICT только с необратимым фильтром 9-7. Преобразование ICT является декоррелирующим преобразованием, которое применяют к первым трем компонентам изображения (пронумерованным как 0, 1 и 2). Эти три компонента входят в ICT и должны иметь одинаковое разделение на опорной сетке и одинаковую битовую глубину.

G.3.1 Прямое преобразование ICT (информативное)

Прямое преобразование ICT применяют к компонентам $I_0(x, y)$, $I_1(x, y)$, $I_2(x, y)$:

$$Y_0(x, y) = -0,299 I_0(x, y) - 0,587 I_1(x, y) + 0,114 I_2(x, y), \quad (G-9)$$

$$Y_1(x, y) = -0,16875 I_0(x, y) - 0,331260 I_1(x, y) + 0,5 I_2(x, y), \quad (\text{G-10})$$

$$Y_2(x, y) = 0,5 I_0(x, y) - 0,41869 I_1(x, y) - 0,08131 I_2(x, y). \quad (\text{G-11})$$

ПРИМЕЧАНИЕ. – Если первые три компонента являются компонентами красного, зеленого и синего, то прямое преобразование ICT является преобразованием YCbCr.

G.3.2 Инверсное преобразование ICT

После выполнения, описанного в Приложении F, инверсного частотного преобразования, применяют следующее инверсное преобразование ICT:

$$I_0(x, y) = Y_0(x, y) + 1,402 Y_2(x, y), \quad (\text{G-12})$$

$$I_1(x, y) = Y_0(x, y) - 0,34413 Y_1(x, y) - 0,71414 Y_2(x, y), \quad (\text{G-13})$$

$$I_2(x, y) = Y_0(x, y) + 1,772 Y_1(x, y). \quad (\text{G-14})$$

Равенства (G-12), (G-13) и (G-14) не предполагают требуемой точности для коэффициентов. После того как выполнено инверсное преобразование ICT, над компонентами изображения без знака производят инверсный сдвиг по уровню DC.

G.4 Выборка компонента цветности и опорная сетка

Взаимосвязь между компонентами и опорной сеткой отмечена в маркере SIZ (см. A.5.1) и описана в разделе B.2.

Приложение Н

Кодирование изображений с областями интересов

(Это Приложение составляет неотъемлемую часть данной Рекомендации | Международного стандарта)

В этом Приложении и во всех его подразделах блок-схемы и таблицы являются нормативными (обязательными) только в том смысле, что они определяют выходные данные, которые при альтернативных применениях должны повторяться.

Данное Приложение описывает технологию области интересов (ROI). Область интересов ROI является частью изображения, которую кодируют в кодированном потоке раньше, чем остальную часть изображения (фон). Кодирование осуществляют таким образом, что информация, соответствующая ROI, предшествует той информации, которая соответствует фону. Используемый метод (описываемый в этом Приложении) называется методом Maxshift.

Н.1 Декодирование ROI

Определяемую в этом разделе процедуру применяют только в случае наличия сегмента маркера RGN (указывающего на доступность области ROI).

Эта процедура перестраивает значимые биты коэффициентов ROI и коэффициентов фона. Процедуру определяют, используя следующие шаги:

- 1) Вычисляют масштабируемую величину s из параметра SPrgn сегмента маркера RGN в кодированном потоке (см. А.6.3). Следующие шаги (2, 3 и 4) применяют к каждому из коэффициентов (u, v) поддиапазона b .
- 2) Если $N_b(u, v) < M_b$ (см. описание $N_b(u, v)$ в D.2.1 и M_b в равенстве (E-2)), то не требуется никаких модификаций.
- 3) Если $N_b(u, v) \geq M_b$ и, по крайней мере, одно из начальных значений M_b (см. описание в E.1) в MSB – наиболее значимый бит ($i = 1, \dots, M_b$) – отлично от нуля, то значение $N_b(u, v)$ обновляют как $N_b(u, v) = M_b$.
- 4) Если $N_b(u, v) \geq M_b$ и все начальные значения M_b в MSB равны нулю, то производят следующие изменения:
 - а) не учитывают начальные положения s в MSB и смещают остальные положения s в MSB в соответствии с равенством (H-1), для $i = 1, \dots, M_b$:

$$MSB_i(b, u, v) = \begin{cases} MSB_{i+s}(b, u, v) & \text{если } i + s \leq N_b(u, v); \\ 0 & \text{если } i + s > N_b(u, v); \end{cases} \quad (\text{H-1})$$

- б) обновляют значение $N_b(u, v)$ в соответствии с равенством (H-2):

$$N_b(u, v) = \max(0, N_b(u, v) - s). \quad (\text{H-2})$$

Н.2 Описание метода Maxshift

Н.2.1 Кодирование ROI (информативное)

Кодирование квантованного коэффициента преобразования выполняют подобно кодированию без областей интересов ROI. Со стороны кодировщика создают маску ROI, которая описывает, какой квантованный коэффициент преобразования должен быть закодирован с наилучшим качеством (вплоть до кодирования без потерь) для того, чтобы закодировать ROI с наивысшим качеством (до кодирования без потерь). Маска ROI является битовым отображением, описывающим эти коэффициенты. Подробное формирование маски описано в разделе Н.3.

Квантованные коэффициенты преобразования вне маски ROI, называемые коэффициентами фона, уменьшают в масштабе с тем, чтобы соответствующие ROI биты размещать в битовых матрицах более высокого порядка, чем фон. Это означает, что если статистический кодер кодирует квантованные коэффициенты преобразования, то битовые матрицы, соответствующие ROI, кодируют раньше информации, которая соответствует фону.

Данный метод может быть описан, используя следующие шаги:

- 1) Сформировать маску ROI, $M(x, y)$ (см. Н.3).
- 2) Определить масштабируемую величину s (см. Н.2.2).
- 3) Добавить значение s в LSB (наименьший значимый бит) к каждому коэффициенту $|q_b(u, v)|$. Число M_b значимости битовых матриц составляет:

$$M'_b = M_b + s, \quad (\text{H-3})$$

где M_b вычисляются из равенства (E-2). Новое значение каждого из коэффициентов:

$$|q_b(u, v)| = |q_b(u, v)| \cdot 2^s. \quad (\text{H-4})$$

- 4) Уменьшить в масштабе все фоновые коэффициенты, имеющие вид $M(x, y)$, с использованием масштабируемой величины s (см. H.3). Таким образом, если фоновый коэффициент $|q_b(u, v)|$ имеет вид $M(x, y)$, то:

$$|q_b(u, v)| = \frac{|q_b(u, v)|}{2^s}. \quad (\text{H-5})$$

- 5) Записать масштабируемую величину s в кодированный поток, используя параметр SPrgn сегмента маркера RGN.

После выполнения вышеперечисленных шагов квантованные коэффициенты преобразования кодируют статистически обычным образом.

H.2.2 Выбор масштабируемой величины s со стороны кодировщика (информативное)

Масштабируемая величина s может быть выбрана в соответствии с равенством (H-6), где $\max(M_b)$ – максимальное число значимости битовых матриц (см. равенство (E-1)) для любых фоновых коэффициентов $q_{BG}(x, y)$ в любом блоке кодирования текущего компонента.

$$s \geq \max(M_b). \quad (\text{H-6})$$

Такой выбор гарантирует, что используемая масштабируемая величина будет достаточно большой, чтобы все значимые биты, соответствующие ROI, были бы в битовых матрицах более высокого порядка, чем все значимые биты, соответствующие фону.

H.3 Комментарии по кодированию области интересов (информативное)

Функциональное назначение ROI, которое описано в разделе H.2, зависит только от масштабируемой величины, выбираемой на стороне кодировщика и, аналогично, от амплитуды коэффициентов на стороне декодера. Со стороны кодировщика создают маску, соответствующую тем коэффициентам, которые необходимо кодировать с наилучшим качеством для выработки ROI с более высоким качеством, чем фон. Раздел H.3.1 описывает метод создания маски ROI для определенной области в изображении. Раздел H.3.2 представляет формирование маски в случае изображений со многими составляющими, а раздел H.3.3 – для несмежных областей. Раздел H.3.4 описывает возможный способ разрешения ситуации с возрастанием коэффициента битовой глубины. Раздел H.3.5 описывает, каким образом расширить маску ROI с тем, чтобы не следовать полностью в области изображения и применить метод Maxshift для кодирования с различным качеством ROI и фона.

H.3.1 Формирование маски области интересов (информативное)

Для достижения ROI с более высоким качеством, чем остальная часть изображения, и в тоже время поддержания достаточной степени сжатия, биты необходимо экономить, посылая меньшее количество информации для фона. Это выполняют, вычисляя маску ROI. Эта маска является битовым набором квантованных коэффициентов преобразования, кодирование которых достаточно для того, чтобы приемник восстановил требуемую область с наилучшим качеством, чем фон (вплоть до степени без потерь).

Для представления формирования маски ROI ограничимся единичным ROI и единичным компонентом изображения и определим образцы, которые принадлежат ROI в области изображения с помощью бинарной маски $M(x, y)$, где:

$$M(x, y) = \begin{cases} 1 & \text{необходимый волновой коэффициент } (x, y) \\ 0 & \text{потеря в точности } (x, y) \text{ без влияния ROI} \end{cases}. \quad (\text{H-7})$$

Эта маска является картой ROI в частотной области, так что она имеет значение отличное от нуля внутри области ROI и нулевое внутри. На каждом этапе маску поддиапазона LL обновляют ряд за рядом, а затем столбец за столбцом. После обновления маска показывает, какой из коэффициентов требуется на этом этапе для воспроизведения инверсным частотным преобразованием коэффициентов предыдущей маски.

Например, заключительным этапом инверсного частотного преобразования является соединение двух поддиапазонов в один. Чтобы отследить этот этап в обратном направлении, находят необходимые коэффициенты обоих поддиапазонов. Предварительный этап состоит в соединении четырех поддиапазонов в два. Для отслеживания этого этапа в обратном направлении определяют коэффициенты в четырех поддиапазонах, которые необходимы, чтобы получить идеальное восстановление коэффициентов, включенных в маску для двух поддиапазонов.

Для составления маски все этапы отслеживают в обратном направлении. Если переданы или получены коэффициенты, соответствующие маске, а инверсное частотное преобразование зависит от них, требуемая область ROI будет восстановлена с более высоким качеством, чем остальная часть изображения (вплоть до степени "без потерь", если без потерь были кодированы коэффициенты ROI).

Описание способа расширения маски, получаемой из различных фильтров, приведено ниже. Похожие методы могут быть использованы для других фильтров.

Н.3.1.1 Формирование маски области интересов, используя обратимый фильтр 5-3 (информативное)

Для получения оптимального набора квантованных коэффициентов, которые необходимо масштабировать, следует использовать равенства, описываемые в этом разделе.

Чтобы понять, какие коэффициенты следует использовать в маске, исследуют инверсное частотное преобразование. Равенства (F-5) и (F-6) представляют коэффициенты, необходимые для восстановления $X(2n)$ и $X(2n + 1)$ без потерь. Из рисунка Н.1 непосредственно видно, что этими коэффициентами являются $L(n)$, $L(n + 1)$, $H(n - 1)$, $H(n)$, $H(n + 1)$. Следовательно, если $X(2n)$ и $X(2n + 1)$ находятся в области ROI, то перечисленные коэффициенты верхнего и нижнего поддиапазонов находятся в маске. Отметим, что $X(2n)$ и $X(2n + 1)$ являются индексированными четными и нечетными точками соответственно, относительно начала опорной сетки.

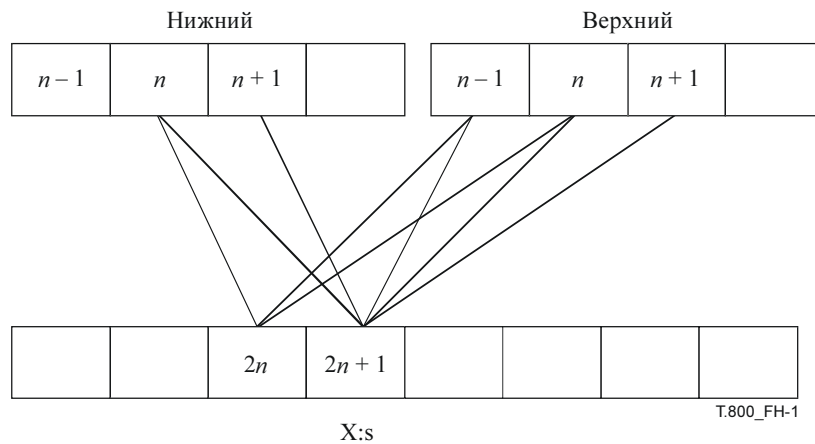


Рисунок Н.1 – Инверсное частотное преобразование с обратимым фильтром 5-3

Н.3.1.2 Формирование маски области интересов, при использовании обратимого фильтра 9-7 (информативное)

Процесс успешного декодирования не зависит от выбора образцов для масштабирования. Для получения оптимального набора квантованных коэффициентов, которые необходимо масштабировать, следует использовать равенства, описываемые в этом разделе.

Чтобы понять, какие коэффициенты следует использовать в маске, исследуют инверсное частотное преобразование аналогично предыдущему разделу Н.3.1.1. $X(2n)$ и $X(2n + 1)$ являются индексированными четными и нечетными точками соответственно, относительно начала опорной сетки (см. рисунок Н.2).

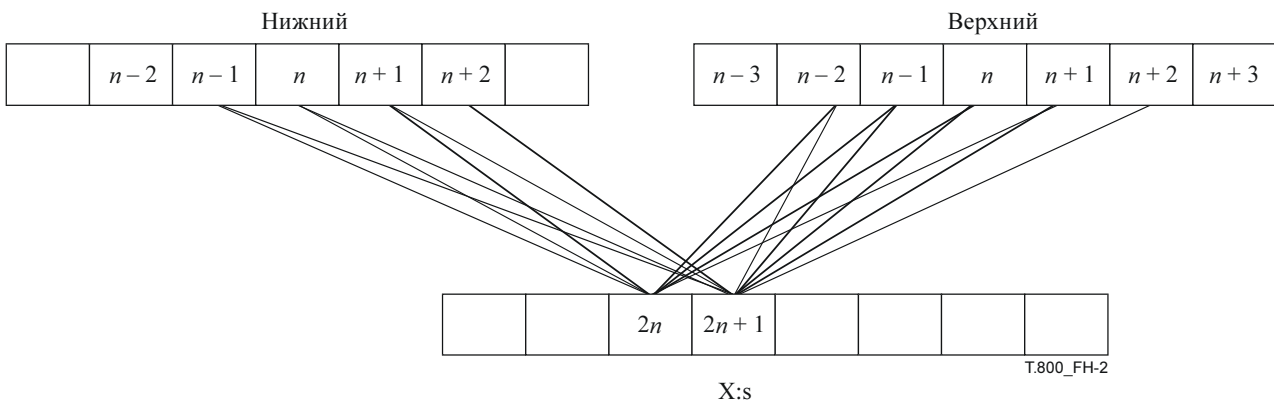


Рисунок Н.2 – Инверсное частотное преобразование с обратимым фильтром 9-7

Необходимые для восстановления $X(2n)$ и $X(2n + 1)$ без потерь коэффициенты могут быть найдены сразу: от $L(n - 1)$ до $L(n + 2)$ и от $H(n - 2)$ до $H(n + 2)$. Следовательно, если $X(2n)$ и $X(2n + 1)$ находятся в области ROI, то перечисленные коэффициенты верхнего и нижнего поддиапазонов находятся в маске.

Н.3.2 Изображение со многими составляющими, комментарий (информативное)

В случае цветowych изображений этот метод применяют отдельно для каждого цветового компонента. Если некоторые из цветовых компонентов являются предварительными образцами, то маску для компонентов предварительных образцов создают таким же образом, как и маску для компонентов без предварительных образцов.

Н.3.3 Несмежные области, комментарий (информативное)

Если область ROI состоит из несмежных частей, то все части имеют одинаковую масштабируемую величину s .

Н.3.4 Точность выполнения, комментарий (информативное)

Метод кодирования ROI в некоторых случаях допускает возникновение ситуаций, в которых превышен динамический диапазон. Однако это можно без труда разрешить, просто не учитывая наименее значимые битовые матрицы, которые превышают предел из-за операции масштабирования в сторону уменьшения. В результате, область ROI будет иметь лучшее качество, чем фон, даже если кодируют весь поток битов. Тем не менее могут возникнуть проблемы, если изображение кодируют с областями ROI в режиме без потерь. Исключение наименее значимых битовых матриц для фона, возможно, приведет к не закодированному фону без потерь, а при неблагоприятном варианте вообще не восстанавливаемому. Это зависит от доступного динамического диапазона.

Н.3.5 Пример использования метода Maxshift (информативное)

Как было описано выше, метод Maxshift позволяет пользователю/приложению определять сложные области произвольной формы, которым будет присвоен высший приоритет по сравнению с остальной частью изображения. Этот метод не требует кодирования или декодирования формы ROI.

Метод Maxshift дает возможность средствам реализации кодировщика использовать то число функциональных возможностей, которые поддерживает соответствующий декодер. Например, возможно использование метода Maxshift для кодирования изображения с разным качеством для ROI и фона. Изображение квантуют так, что ROI получает желаемое качество (с потерями или без потерь), а затем применяют метод Maxshift. Если изображение кодируют постепенно по уровню, то нет необходимости кодировать все уровни частотных коэффициентов, относящихся к фону. Это соответствует использованию различных шагов квантования для ROI и фона.

Если область ROI закодирована без потерь, то наиболее оптимальный набор частотных коэффициентов, дающих для ROI результат без потерь, описывают создаваемой маской с использованием алгоритмов, которые представлены в Н.3.1. Однако метод Maxshift поддерживает использование любой маски, поскольку декодеру нет необходимости формировать маску. Таким образом, кодировщик может включать весь поддиапазон в маску ROI, например, самый низкий поддиапазон, и поэтому передавать на начальных ступенях последовательной передачи вариант фона с низким разрешением. Это выполняют масштабированием всех квантованных коэффициентов преобразования целого поддиапазона. Другими словами, пользователь может решить, в каком поддиапазоне он начнет кодирование ROI, и поэтому нет необходимости ждать целой области ROI до получения какой-либо информации для фона.

Приложение I

Синтаксис файлового формата JP2

(Это Приложение составляет неотъемлемую часть данной Рекомендации | Международного стандарта)

В этом Приложении блок-схемы и таблицы являются нормативными (обязательными) только в том смысле, что они определяют выходные данные, которые при альтернативных применениях должны повторяться.

I.1 Область применения файлового формата

В этом Приложении рассмотрен оптимальный формат файла, который можно выбрать для приложений, чтобы использовать этот формат для переноса данных сжатого изображения JPEG 2000. Хотя этот формат используют не во всех приложениях, для многих приложений он может оказаться полезным. В тех приложениях, где используют этот формат, его могут применять таким образом, как это описано в данном Приложении.

В этом Приложении:

- определен бинарный контейнер для изображения и метаданных;
- определен механизм для выявления свойств изображения, таких как градация или цветовое пространство изображения;
- определен механизм, с помощью которого считыватель может распознать в файле информацию о правах на интеллектуальную собственность;
- определен механизм, с помощью которого метаданные (включая специальную информацию продавца) могут быть включены в файлы, определенные данной Рекомендацией | Международным стандартом.

I.2 Введение в файловый формат JP2

Файловый формат JPEG 2000 (файловый формат JP2) создает основу для хранения специальных данных (метаданных) совместно с кодированным потоком JPEG 2000, например, информацию отображения изображения. Поскольку для многих приложений требуется, чтобы аналогичные наборы информации были объединены с данными сжатого изображения, полезно определить формат этого набора данных совместно с определением технологии сжатия и синтаксисом кодированного потока.

Концептуально файловый формат JP2 включает кодированный поток JPEG 2000 совместно с другими основными информационными данными о потоке. Строительную единицу файлового формата JP2 называют блоком. Вся информация, которую содержит файл JP2, заключена в блоках. Данная Рекомендация | Международный стандарт определяет несколько типов блоков. Определение каждого отдельного типа блока определяет и род информации, которую можно найти в блоках этого типа. Некоторые блоки будут определены как содержащие другие блоки.

I.2.1 Идентификация файла

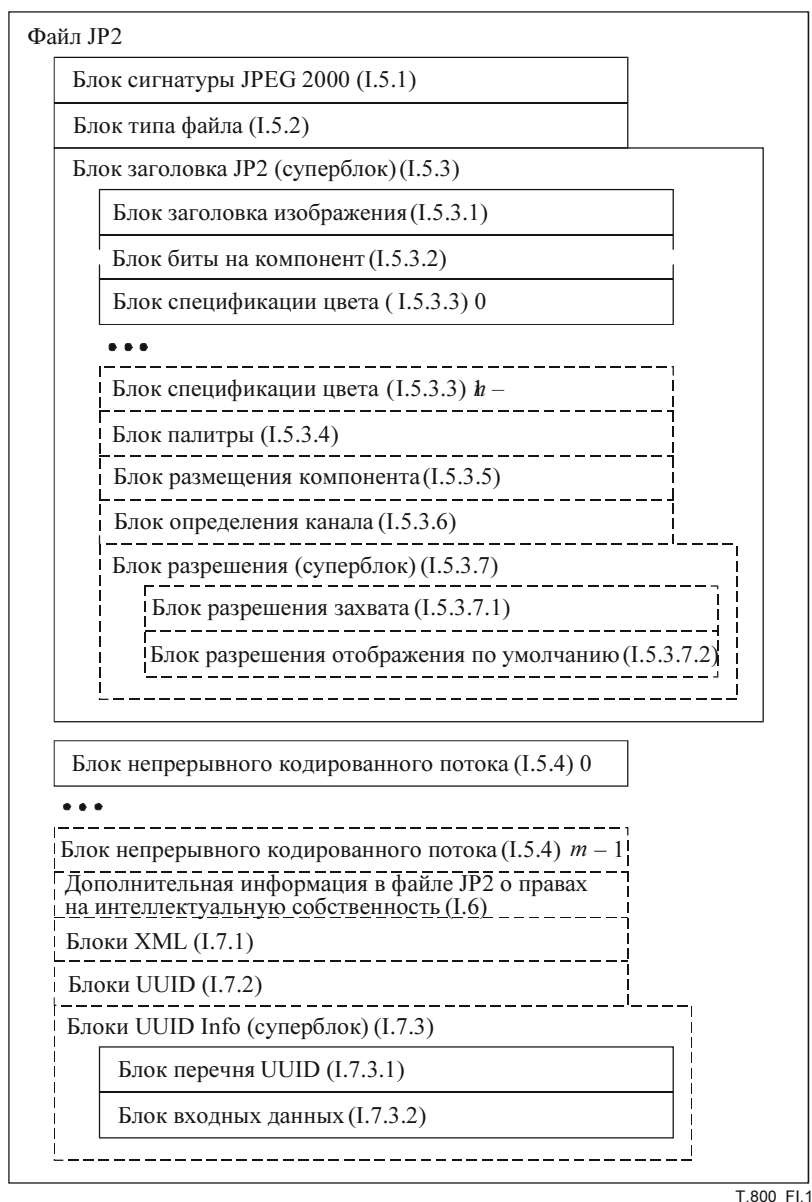
Файл JP2 можно идентифицировать несколькими механизмами. При хранении в традиционной компьютерной файловой системе файлу JP2 присваивают расширение ".jp2" (считыватели могли бы допустить смешанный случай для символов алфавита). В файловых системах Макинтош файлам JP2 может быть присвоен тип кода 'jp2\040'.

I.2.2 Файловая организация

Файл JP2 представляет набор блоков. Некоторые из этих блоков – независимые, а некоторые содержат в себе другие блоки. Бинарная структура файла является последовательностью смежных блоков. Запуском первого блока должен быть первый байт файла, а последний байт последнего блока должен быть последним байтом файла.

Бинарная структура блока определена в I.4.

Логическая структура файла JP2 показана на рисунке I.1. Блоки, очерченные пунктирными линиями, являются дополнительными. Однако блоки внутри дополнительных могут быть определены как обязательные. В этом случае, если существует дополнительный блок, обязательные блоки внутри дополнительного также существуют.



T.800_FI.1

Рисунок I.1 – Концептуальная структура файла JP2

На рисунке I.1 показаны только ограниченные взаимоотношения между блоками в файле. Конкретный порядок этих блоков в файле обычно не устанавливают. Однако блок сигнатуры JPEG 2000 должен быть первым в файле JP2, блок типа файла должен непосредственно следовать за блоком сигнатуры JPEG 2000, а блок заголовка JP2 должен находиться перед блоком непрерывного кодированного потока.

Показанный на рисунке I.1 файл – это строгая последовательность блоков. Прочие блоки могут располагаться между блоками, определенными данной Рекомендацией | Международным стандартом. Однако вся информация в файле JP2 должна иметь формат блока. Поток байтов не укладываются в формат блока и не должны находиться в этом файле.

Как показано на рисунке I.1, файл JP2 содержит блоки сигнатуры JPEG 2000, заголовка JP2 и один или более блоков непрерывного кодированного потока. Файл JP2 может также содержать другие блоки, определенные составителем файла. Например, файл JP2 может содержать несколько блоков XML (с метаданными) между блоком заголовка JP2 и первым блоком непрерывного кодированного потока.

1.2.3 Спецификация: оттенки серого, цвет, цветовая палитра, многокомпонентная форма

Формат файла JP2 предоставляет два метода для определения цветового пространства изображения. Метод нумерации определяет цветовое пространство изображения спецификацией численной величины, которая описывает цветовое пространство. В данной Рекомендации | Международном стандарте изображения в цветовых пространствах RGB и оттенков серого можно определить методами нумерации.

Файловый формат JP2 предоставляет также спецификацию цветового пространства изображения введением в этот файл ограниченной формы характеристик ICC. Эти характеристики должны быть либо монохромными, либо принадлежать к классу характеристик, основанному на трехкомпонентной матрице, как это определено в спецификации ICC: Profile Format Specification, ICC.1:1998-09. Это обеспечивает спецификации широкий диапазон цветовых пространств оттенков серого и класса RGB, а также нескольких других пространств, которые могут быть представлены этими двумя классами характеристик. В J.9 представлено более подробное описание разрешенных преобразований цветовых пространств, описано, как хранить эти преобразования в файле и как обрабатывать изображение с помощью такого преобразования без использования машины цвета ICC. Несмотря на ограничения, такие характеристики ICC полностью совместимы с общими характеристиками ICC, а изображение можно обработать, таким образом, с помощью любой совместимой с ICC машиной, которая поддерживает характеристики, определенные в ICC.1:1998-09.

В дополнение, для спецификации цветового пространства изображения эта Рекомендация | Международный стандарт дает средства, с помощью которых простой компонент цветного изображения может быть декодирован и обратно преобразован в многокомпонентную форму перемещением из пространства индексов в многокомпонентное пространство. Любое такое разложение палитры может быть использовано перед интерпретацией цветового пространства. В случае многоцветных изображений спецификация цветового пространства изображения применима к многокомпонентным значениям, хранимым в цветовой палитре.

1.2.4 Включение каналов непрозрачности

Формат JP2 обеспечивает средство индикации присутствия дополнительных каналов (таких, как канал непрозрачности), определение типа этих каналов, а также порядка и источника этих каналов (извлекаются ли они непосредственно из кодированного потока или создаются применением палитры к компоненту кодированного потока). Когда считывающее устройство (считыватель) открывает файл JP2, он определяет порядок и тип каждого компонента. В Приложении должно быть соответствие между определением и порядком компонента в файле JP2 и порядком, определенным спецификацией цветового пространства. Как только компоненты файла размещены в каналах цвета, декомпрессированное изображение может быть обработано с помощью необходимых преобразований пространства цвета.

Для многих приложений требуются иные компоненты, чем каналы цвета. Например, многие изображения, использованные на страницах Интернет, содержат информацию непрозрачности. Браузеры используют эту информацию для подмешивания изображения к заднему плану. Поэтому желательно включить в единый поток как цвет, так и дополнительные каналы.

Действия приложений с непрозрачностью и другими дополнительными каналами выходят за рамки задач данной Рекомендации | Международного стандарта.

1.2.5 Метаданные

Одним из важнейших аспектов файлового формата JP2 является способность добавлять к файлу JP2 метаданные. Поскольку вся информация инкапсулирована в блоки, а все блоки имеют типы, формат представляет для считывателя простой механизм извлечения соответствующей информации, в то же время игнорируя любой блок, в котором содержится информация, непонятная данному конкретному считывателю. Таким способом могут быть созданы новые блоки, как с помощью данной или другой Рекомендации | Международного стандарта, так и частным образом. Кроме того, любой новый блок, добавленный к файлу JP2, не должен изменять внешний вид изображения.

1.2.6 Согласование с файловым форматом

Все согласованные файлы должны содержать все блоки, которые требует эта Рекомендация | Международный стандарт, и эти блоки должны быть определены в этой Рекомендации | Международном стандарте. Кроме того, соответствующие считыватели должны корректно интерпретировать все требуемые этой Рекомендацией | Международным стандартом блоки и, следовательно, корректно интерпретировать все согласованные файлы.

1.3 Архитектура спецификации Оттенки серого/ Цвет/ Цветовая палитра/ Многокомпонентность

Одним из наиболее важных аспектов файлового формата является его свойство описать данные пространства цвета изображения. Для правильного отображения или интерпретации данных изображения важно, чтобы пространство цвета правильно характеризовало изображение. Файловый формат JP2 обеспечивает многоуровневый механизм, характеризующий пространство цвета изображения.

I.3.1 Метод нумерации

Простейший метод, характеризующий пространство цвета изображения – определение целочисленного кода, представляющего пространство цвета, в котором закодировано изображение. Этот метод обрабатывает спецификацию RGB, оттенков серого и изображения YCC. Расширения этого метода могут быть использованы для описания других пространств цвета, включая определение многокомпонентных изображений.

Например, файловое изображение может указывать, что конкретное изображение закодировано в пространстве цвета RGB. Чтобы правильно интерпретировать и отобразить изображение, Приложение должно с самого начала понимать определение пространства цвета RGB. Поскольку Приложение должно с самого начала понимать каждое специфическое пространство цвета, сложность метода зависит от точности спецификации этого пространства. Кроме того, сложность этого механизма пропорциональна числу пространств цвета, которые описаны и требуют согласования. Несмотря на то что этот метод обеспечивает высокий уровень взаимодействия для изображений, закодированных с помощью пространств цвета, для правильной интерпретации последних требуется согласование, поэтому этот метод не является гибким. В данной Рекомендации | Международном стандарте определен особый набор пространств цвета, которые требуют интерпретации для согласования.

I.3.2 Метод ограниченных характеристик ICC

В Приложении может быть также специфицировано пространство цвета изображения с использованием двух типов ограниченных характеристик ICC. Этот метод обрабатывает спецификацию наиболее часто используемых пространств цвета RGB и оттенков серого с помощью менее сложного метода.

Характеристики ICC – это стандартное представление преобразования, которое требуется, чтобы превратить одно пространство цвета в другое. С учетом формата файла JP2, характеристики ICC определяют, каким образом декомпрессированные образцы из кодированного потока превращают в стандартное пространство цвета (пространственные характеристики соединения – PCS). В зависимости от первоначального пространства цвета образцов преобразование может быть либо очень простым, либо очень сложным.

Спецификация формата характеристик ICC определяет два особых класса характеристик ICC, которые просты при внедрении. Это относится к спецификациям характеристик типа "монохромного входа" и "Основанных на трехкомпонентной матрице". Эти характеристики ограничивают преобразование от источника пространства цвета к PCS_{XYZ} применением нелинейной кривой и матрицы 3 × 3. На практике полагают, что все применения, включая простые устройства, в состоянии обрабатывать изображение с помощью такого преобразования. Таким образом, для правильной интерпретации пространства цвета любого изображения требуются все согласованные приложения, которые определяют пространство цвета, используя набор характеристик типа ICC.

Для файлового формата JP2 характеристики должны соответствовать характеристикам ICC, как это определено в спецификации формата ICC (Profile Format Specification, ICC.1:1998-09), а также вышеуказанным ограничениям. В J.9 содержится более подробное описание разрешенных преобразований пространств, а также описание хранения этих преобразований в файле и обработки изображения с использованием преобразования без машины управления цветом ICC.

I.3.3 Использование составных методов

Архитектурно формат допускает, чтобы в файл были включены составные методы, а также допускает определение дополнительных и расширенных методов. Согласованным с этими расширениями считывателям это дает возможность выбора, каким образом обрабатывать изображение для интерпретации его пространства цвета. Однако самым первым методом в этом файле (в первом блоке спецификации пространства цвета в блоке заголовка JP2) должен быть один из тех методов, которые определены и ограничены данной Рекомендацией | Международным стандартом. Согласованный считыватель может использовать этот первый метод и игнорировать остальные методы в этом файле (в дополнительных блоках спецификации пространства цвета).

I.3.4 Палитры изображений

В дополнение к определенным интерпретациям изображения в терминах пространства цвета в данной Рекомендации | Международном стандарте допускается декодирование единственного компонента, в котором значение этого компонента представляет индекс в виде палитры цветов. Вход декомпрессированного образца палитры превращает единственное значение в многокомпонентный кортеж. Значение этого кортежа представляет цвет образца. Затем этот кортеж должен быть интерпретирован в соответствии с другими методами спецификации цвета (нумерованными или ограниченными ICC) так, как будто этот многокомпонентный образец был непосредственно извлечен из многих компонентов кодированного потока.

I.3.5 Взаимодействия с декоррелированным преобразованием множества компонентов

Спецификация цвета в файловом формате JP2 не зависит от использования преобразования множества компонентов в кодированном потоке (параметр CSSiz сегмента маркера SIZ, как указано в A.5.1 и в Приложении G). Преобразование пространств цвета, описанное последовательностью блоков спецификации цвета, должно быть использовано для образцов изображения после того, как к декомпрессированным образцам будет применено обратное преобразование множества компонентов. Несмотря на то, что использование преобразований этих некоррелированных компонентов раздельное, применение преобразования множества компонентов, основанное на кодировщике, часто улучшает сжатие данных цветного изображения.

I.3.6 Ключ к графическим описаниям (информативное)

Каждый блок описывают в терминах его функции, использования и длины. Функция описывает информацию, которая содержится в блоке. Использование описывает информацию, которую содержит блок. Длина описывает параметры, определяющие длину блока.

За этими описаниями следует рисунок, который показывает порядок и взаимоотношения параметров в блоке. На рисунке I.2 показан такой пример. Для обозначения параметров блока использован прямоугольник. Ширина прямоугольника пропорциональна числу байтов параметра. Затененный прямоугольник (с диагональными штрихами) указывает на параметр переменной длины. Два параметра с верхними индексами и серой областью между ними показывают разброс некоторых из этих параметров. Последовательность из двух групп с множеством параметров с верхними индексами, разделенными серой областью, показывают разброс этих групп параметров (один ряд каждого параметра в группе, за которым следует другой ряд). Дополнительные параметры и блоки показаны в прямоугольнике со штриховыми линиями.

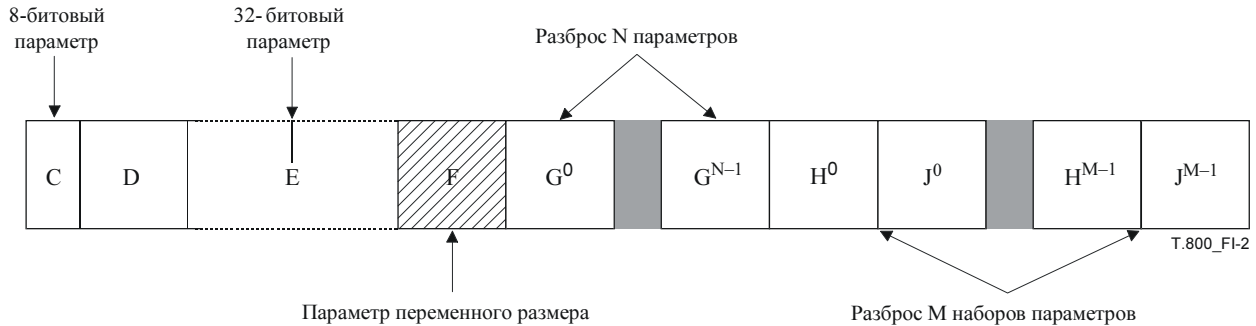


Рисунок I.2 – Пример описания содержимого блока

После рисунка помещен список, в котором указано значение каждого параметра блока. Если параметры повторяются, то объясняется длина d и характер разброса параметров. В качестве примера на рисунке I.2 параметры C, D, E и F имеют по 8, 16 и 32 битов и переменную длину соответственно. Условные изображения G⁰ и G^{N-1} обозначают, что имеется подряд N различных параметров Gⁱ. Группа параметров H⁰ и H^{M-1}, а также J⁰ и J^{M-1} показывает, что блок может содержать параметр H⁰, за которым следует J⁰, далее следуют H¹ и J¹, и продолжается до H^{M-1} и J^{M-1} (в общем, M случаев каждого параметра). Кроме того, поле E является дополнительным и может отсутствовать в блоке.

После этого список превращают в таблицу, которая либо описывает разрешенные значения параметров, либо отсылает к другим таблицам, где описаны эти значения.

Наконец, на рисунке с содержанием суперблока знак (...) использован для указания на то, что содержимое файлов между двумя блоками специально не определено. Каждый блок (или последовательность блоков), если иное не указано в определении этого блока, может находиться в месте, обозначенном как (...).

Например, суперблок, показанный на рисунке I.3, должен содержать блок AA и BB, причем блок BB должен следовать за блоком AA. Однако между блоками AA и BB могут находиться другие блоки. Обращение с неизвестными блоками обсуждается в I.8.



Рисунок I.3 – Пример описания содержимого суперблока

I.4 Определение блока

Физически каждый объект в файле инкапсулируют в бинарную структуру, которую называют блоком. Эта бинарная структура показана на рисунке I.4:

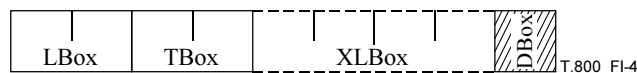


Рисунок I.4 – Организация блока

- LBox:** Длина блока. Это поле описывает длину блока, которая хранится в виде 4-байтовых битов представления целого числа без знака. Это значение включает все поля блока, в том числе длину и тип. Если значение этого поля 1, то поле XLBox должно существовать, а значение этого поля должно быть действительной длиной блока. Если значение этого поля 0 и заполнено поле LBox, то длина блока неизвестна. В этом случае этот блок содержит все байты до конца файла. Если блок с длиной 0 находится внутри другого блока (суперблока), то длина этого суперблока также должна быть 0. Это значит, что данный блок – последний в файле. Значения 2-7 зарезервированы для использования ИСО.
- TBox:** Тип блока. Это поле описывает тип информации, которая находится в поле DBox. Значение этого поля кодируют 4-байтовыми битами представления целого числа без знака. Однако, блоки обычно записывают строкой знаков ИСО/МЭК 646 для целых величин. Для всех типов блоков, которые определены данной Рекомендацией | Международным стандартом, типы блоков будут обозначены как строкой знаков (обязательное условие), так и 4-байтовыми шестнадцатеричными целыми (информативное условие). Кроме того, в трансляции строки знаков типа блока, как например "\040", показан знак пробела. Все значения TBox, не определенные в данной Рекомендации | Международном стандарте, зарезервированы для использования ИСО.
- XLBox:** Расширенная длина блока. Это поле показывает действительную длину блока, если значение поля LBox – это 1. Поле хранят в виде 8-байтовых битов представления целого числа без знака. Это значение включает все поля блока, в том числе поля LBox, TBox и XLBox.
- DBox:** Содержание блока. Это поле содержит действительную информацию в блоке. Формат содержания блока зависит от типа блока и может быть индивидуально определен для каждого типа.

Таблица I.1 – Бинарная структура блока

Наименование поля	Размер (биты)	Значение
LBox	32	0, 1, или $8 - (2^{32} - 1)$
TBox	32	Переменное
XLBox	64 0	$16 - (2^{64} - 1)$; если LBox = 1 Не применимо; если LBox \neq 1
DBox	Переменный	Переменное

Например, рассмотрим иллюстрацию на рисунке I.5 последовательности блоков, включая один блок, в котором содержатся другие блоки:

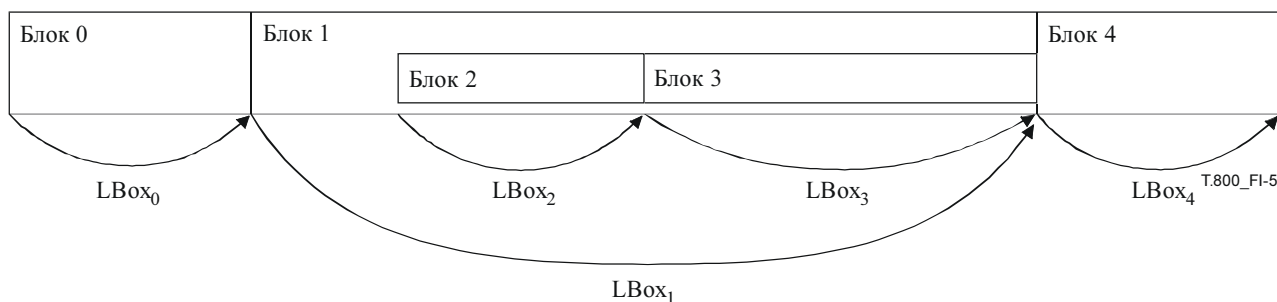


Рисунок I.5 – Иллюстрация длины блоков

Как показано на рисунке I.5, длина каждого блока включает длины любых блоков, которые находятся в этом блоке. Например, длина блока 1 включает длины блока 2 и блока 3 вдобавок к полям LBox и TBox собственно блока 1. В этом случае, если блок 1 не был понят считывателем, он не сможет распознать существование блоков 2 и 3, так как они будут полностью пропущены скачком длины от начала блока 1.

В таблице I.2 перечислены все блоки, определенные данной Рекомендацией | Международным стандартом. Отступ в таблице означает иерархическое включение структуры блока в файл JP2.

Таблица I.2 – Определенные блоки

Наименование блока	Тип	Суперблок	Требуемый?	Комментарии
Блок сигнатуры JPEG 2000	'jP\040\040' (0x6A50 2020)	Нет	Требуемый	Этот блок единственным образом идентифицирует файл как принадлежащий семейству файлов JPEG 2000.
Блок типа файла	'ftyp' (0x6674 7970)	Нет	Требуемый	Этот блок определяет тип файла, версию и информацию о совместимости, включая определение совместимости с файлом JP2 или возможность прочтения считывателем JP2.
Блок заголовка JP2	'jp2h' (0x6A70 3268)	Да	Требуемый	Этот блок содержит ряд блоков с типом заголовков об информации этого файла.
Блок заголовка изображения	'ihdr' (0x6968 6472)	Нет	Требуемый	Этот блок определяет размер изображения и другие связанные с этим поля.
Блок битов на компонент	'bpc' (0x6270 6363)	Нет	Дополнительный	Этот блок определяет глубину битов компонентов в файле в случае, когда глубина битов не постоянна для всех компонентов.
Блок спецификации цвета	'colr' (0x636F 6C72)	Нет	Требуемый	Этот блок определяет пространство цвета изображения.
Блок палитры	'pclr' (0x7063 6C72)	Нет	Дополнительный	Этот блок определяет палитру, которую вмещает единственный компонент в пространственном индексе многокомпонентного изображения.
Блок размещения компонентов	'cmpr' (0x636D 6170)	Нет	Дополнительный	Этот блок определяет размещение между компонентами палитры и кодированного потока.
Блок определения канала	'cdef' (0x6364 6566)	Нет	Дополнительный	Этот блок определяет тип и порядок компонентов в кодированном потоке таким же образом, как определены эти же параметры, созданные приложением палитры.
Блок разрешения	'res\040' (0x7265 7320)	Да	Дополнительный	Этот блок содержит разрешение сетки.
Блок разрешения захвата	'resc' (0x7265 7363)	Нет	Дополнительный	Этот блок определяет разрешение сетки, в которой произошел захват изображения.
Блок разрешения отображения по умолчанию	'resd' (0x7265 7364)	Нет	Дополнительный	Этот блок определяет по умолчанию разрешение сетки, в которой должно быть отображено изображение.
Блок непрерывности кодированного потока	'jp2c' (0x6A70 3263)	Нет	Требуемый	Этот блок содержит кодированный поток, определенный в Приложении А.
Блок интеллектуальной собственности	'jp2i' (0x6A70 3269)	Нет	Дополнительный	Этот блок содержит информацию интеллектуальной собственности на изображение.
Блок XML	'xml\040' (0x786D 6C20)	Нет	Дополнительный	Этот блок обеспечивает инструмент, с помощью которого продавец сможет добавить к файлу JP2 информацию в формате XML.
Блок UUID	'uuid' (0x7575 6964)	Нет	Дополнительный	Этот блок обеспечивает инструмент, с помощью которого продавец сможет добавить дополнительную информацию к файлу без риска конфликта с другими продавцами.
Блок информации UUID	'uinf' (0x7569 6E66)	Да	Дополнительный	Этот блок обеспечивает инструмент, с помощью которого продавец сможет получить доступ к дополнительной информации, связанной с UUID.
Блок списка UUID	'ulst' (0x7563 7374)	Нет	Дополнительный	Этот блок определяет список UUID.
Блок URL	'url\040' (0x7572 6C20)	Нет	Дополнительный	Этот блок определяет URL.

I.5 Определенные блоки

Следующие блоки должны быть правильно интерпретированы всеми согласованными считывателями. Каждый из этих блоков соответствует стандартной структуре блока, определенной в I.4. Последующие разделы определяют значение поля DBox из таблицы I.1 (содержание блока). Предполагается, что поля LBox, TBox и XLBox существуют для каждого блока в файле, как это описано в Приложении I.4.

1.5.1 Блок сигнатуры JPEG 2000

Блок сигнатуры JPEG 2000 устанавливает, что формат этого файла определен Рекомендацией | Международным стандартом JPEG 2000, а также дает немного информации, которая поможет определить достоверность оставшейся части файла. Блок сигнатуры JPEG 2000 должен быть первым блоком в файле, и все файлы должны содержать один и только один блок сигнатуры JPEG 2000.

Тип блока сигнатуры JPEG 2000 должен быть "jp\040\040" (0x6A50 2020). Длина этого блока должна быть 12 байтов. Содержимым блока должна быть 4-байтовая строка знаков "<CR><LF><0x87><LF>" (0x0D0A 870A). Для целей контроля файла этот блок можно рассматривать как 12-байтовую строку фиксированной длины, которая должна иметь значение: 0x0000 000C 6A50 2020 0D0A 870A.

Сочетание особого типа и содержания этого блока позволяют использовать его для обнаружения ошибок в обычном наборе передаваемых файлов. Последовательность CR-LF в содержании обнаруживает передачу поврежденных файлов, которые изменили последовательность. Контрольная длина проверяет проблему инверсии передачи CR-LF. Третий знак содержимого блока имеет установку на перехват поврежденного файла, сбрасывая бит 7.

1.5.2 Блок типа файла

Блок типа файла определяют этой Рекомендацией | Международным стандартом, которая полностью описывает все компоненты этого файла, а также отдельный список считывателей, определенный другими Рекомендациями | Международными стандартами, с которыми совместим этот файл. Таким образом, файл может быть правильно интерпретирован и другими Рекомендациями | Международными стандартами, с которыми он совместим. Этот блок должен непосредственно следовать за блоком сигнатуры JPEG 2000. В этом и заключается разница между стандартом, который полностью описывает файл, и другими стандартами, которые интерпретируют подмножество этих файлов.

Все файлы должны содержать один и только один блок типа файла.

Вид блока типа файла должен быть следующий: 'fup' (0x6674 7970). Содержание этого блока должно соответствовать рисунку I.6:

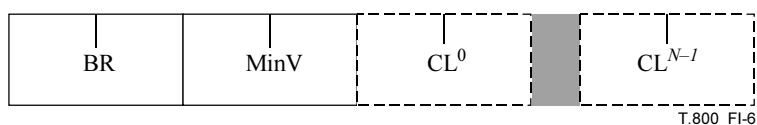


Рисунок I.6 – Организация содержимого блока типа файла

BR: Брэнд (Brand). Это поле указывает на Рекомендацию | Международный стандарт, который полностью определяет этот файл. Это поле определено 4-байтовой строкой знаков ИСО/МЭК 646. Значения этого поля определены в таблице I.3:

Таблица I.3 – Разрешенные значения Brand

Значение	Содержание
"jp2\040"	IS 15444-1, Приложение I (Это – Рекомендация Международный стандарт)
Прочие значения	Зарезервировано для использования ИСО

Кроме того, поле Брэнд следует рассматривать как функциональный эквивалент основного номера версии. Изменение основной версии (если это когда-нибудь происходит), представляющее несовместимое изменение в формате файла JP2, должно определить другое значение в поле Брэнд.

Если значение поля Брэнд не 'jp2\040', то значение 'jp2\040' в списке совместимости показывает, что считыватель JP2 может интерпретировать файл так, как это было сделано изготовителем файла.

MinV: Младшая версия. Этот параметр определяет младшую версию спецификации JP2, с которой совместим файл. Этот параметр определен 4-мя байтами битов представления целого числа без знака. Значение этого поля должно быть нулевым. Однако считыватели могут продолжить анализ и интерпретацию этого файла, даже если значение этого поля ненулевое.

CLⁱ: Список совместимости. Это поле определяет код, представляющий данную Рекомендацию | Международный стандарт, другой стандарт или характеристики другого стандарта, с которым совместим файл. Это поле кодируют 4-байтовой строкой знаков ИСО/МЭК 646. Файл, который соответствует данной Рекомендации | Международному стандарту, должен иметь по крайней мере одно поле CLⁱ в блоке типа файла и должен содержать значение 'jp2\040' в одном из полей CLⁱ этого блока. Все согласованные считыватели должны правильно интерпретировать все файлы со знаком 'jp2\040' в одном из полей CLⁱ.

Если одно из полей CL^i содержит значение "J2P0", то первый кодированный поток в этом файле JP2 file ограничивают, как указано в Profile-0 из таблицы A.45.

Если одно из полей CL^i содержит значение "J2P1", то первый кодированный поток в этом файле JP2 file ограничивают, как указано в Profile-1 из таблицы A.45

Прочие значения полей в списке совместимости зарезервированы для ИСО.

Число полей CL^i определено длиной этого блока.

Таблица I.4 – Формат содержания блока типа файла

Наименование поля	Размер (биты)	Значение
BR	32	$0 - (2^{32} - 1)$
MinV	32	0
CL^i	32	$0 - (2^{32} - 1)$

I.5.3 Блок заголовка JP2 (суперблок)

Блок заголовка JP2 содержит исходную информацию о файле, как например, число компонентов, пространство цвета и разрешение сетки. Этот блок является суперблоком. В файле JP2 должен быть один и только один блок заголовка JP2. Блок заголовка JP2 может находиться в любом месте файла после блока типа файла, но перед блоком непрерывного кодированного потока. Этот блок должен иметь такой же уровень, что блоки сигнатуры JPEG 2000 и типа файла (он должен находиться внутри другого суперблока в файле).

Тип блока заголовка JP2 должен быть "jp2h" (0x6A70 3268).

Этот блок содержит несколько блоков. Другие блоки могут быть определены другими стандартами, и согласованные считыватели могут их игнорировать. Эти блоки, расположенные внутри блока заголовка JP2 и определенные данной Рекомендацией | Международным стандартом, показаны на рисунке I.7:

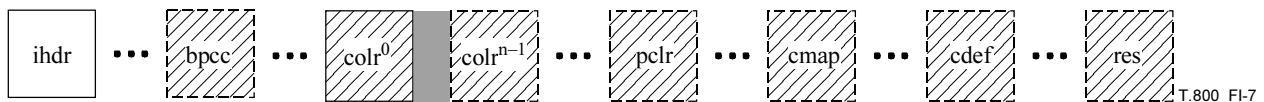


Рисунок I.7 – Организация содержимого блока заголовка JP2

- ihdr:** Блок заголовка изображения. Этот блок описывает информацию об изображении, как например, высоту и ширину. Структура блока определена в I.5.3.1. Этот блок должен быть первым в блоке заголовка JP2.
- bpc:** Блок битов на компонент. Этот блок определяет битовую глубину каждого компонента в кодированном потоке после декомпрессии. Его структура определена в I.5.3.2. Этот блок может находиться в любом месте блока заголовка JP2 при условии, что он будет следовать за блоком заголовка изображения.
- colr:** Блоки спецификации цвета. Эти блоки определяют пространство цвета декомпрессированного изображения. Их структура описана в I.5.3.3. В блоке заголовка JP2 должен быть по крайней мере один блок спецификации цвета. Использование множества блоков спецификации цвета обеспечивает декодеру возможность получить многократную оптимизацию или опции совместимости для обработки цвета. Эти блоки могут находиться в любом месте блока заголовка JP2 при условии, что они будут следовать за блоком заголовка изображения. Все блоки спецификации цвета в блоке заголовка JP2 должны быть смежными.
- pclr:** Блок палитры. Этот блок определяет палитру для создания из единственного компонента множества компонентов. Структура блока описана в I.5.3.4. Этот блок может находиться в любом месте блока заголовка JP2 при условии, что он будет следовать за блоком заголовка изображения.
- cmap:** Блок размещения компонента. Этот блок определяет, какие каналы изображения идентифицированы из действительных компонентов кодированного потока. Структура блока описана в I.5.3.5. Этот блок может находиться в любом месте блока заголовка JP2 при условии, что он будет следовать за блоком заголовка изображения.
- cdef:** Блок определения канала. Этот блок определяет каналы в изображении. Структура блока описана в I.5.3.6. Этот блок может находиться в любом месте блока заголовка JP2 при условии, что он будет следовать за блоком заголовка изображения.
- res:** Блок разрешения. Этот блок описывает захват и отображение по умолчанию разрешений сетки изображения. Структура блока описана в I.5.3.7. Этот блок может находиться в любом месте блока заголовка JP2 при условии, что он будет следовать за блоком заголовка изображения.

1.5.3.1 Блок заголовка изображения

Этот блок содержит исходную информацию фиксированной длины об изображении, как например, размер изображения и число компонентов. Содержимое блока заголовка JP2 должно запускаться вместе с блоком заголовка изображения. Экземпляры этого блока в других местах файла должны игнорироваться. Длина блока заголовка изображения должна быть 22 байта, включая длину и тип полей. Многие из информации в блоке заголовка изображения является дополнительным по сравнению с информацией, хранимой в кодированном потоке.

Все ссылки на "кодированный поток" в описаниях полей блока заголовка изображения применимы к кодированному потоку, который находится в первом блоке непрерывного кодированного потока в этом файле. Файлы, которые содержат противоречивую информацию между блоком заголовка изображения и первым кодированным потоком, не являются согласующимися. Однако считыватели могут сделать попытку прочитать эти файлы, используя значения, найденные в кодированном потоке.

Тип блока заголовка изображения должен быть "ihdr" (0x6968 6472), а содержимое блока должно иметь формат, изображенный на рисунке 1.8:

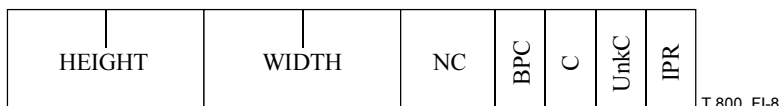


Рисунок 1.8 – Организация содержимого блока заголовка изображения

- HEIGHT:** Высота площади изображения. Значение этого параметра показывает высоту площади изображения. Это поле хранят в виде 4-байтовых битов представления целого числа без знака. Значение этого поля должно быть Ysiz – YOsiz, где Ysiz и YOsiz – значения соответствующих полей в маркере SIZ кодированного потока. Для иллюстрации площади изображения см. рисунок В.1. Однако точки опорной сетки – это не обязательно квадрат. Соотношение геометрических размеров опорной сетки определено блоком разрешения. Если блок разрешения не представлен, то считыватель должен полагать, что точки опорной сетки – это квадрат.
- WIDTH:** Ширина площади изображения. Значение этого параметра показывает ширину площади изображения. Это поле хранят в виде 4-байтовых битов представления целого числа без знака. Значение этого поля должно быть Xsiz – XOsiz, где Xsiz и XOsiz – значения соответствующих полей в маркере SIZ кодированного потока. Для иллюстрации площади изображения см. рисунок В.1. Однако точки опорной сетки – это не обязательно квадрат. Соотношение геометрических размеров опорной сетки определено блоком разрешения. Если блок разрешения не представлен, то считыватель должен полагать, что точки опорной сетки – это квадрат.
- NC:** Число компонентов. Этот параметр определяет число компонентов в кодированном потоке и хранится в виде 2-байтовых битов представления целого числа без знака. Значение этого поля должно быть равным значению поля Csiz в маркере SIZ кодированного потока.
- BPC:** Битов на компонент. Этот параметр определяет глубину битов (битовую глубину) компонентов в кодированном потоке минус 1 и хранится в виде однобайтового поля.
 Если глубина битов и знак одинаковы для всех компонентов, то этот параметр показывает, что глубина битов должна быть эквивалентна значениям полей Ssiz¹ маркера SIZ в кодированном потоке (и все они должны быть одинаковы). Если компоненты варьируются по глубине битов и/или знаку, то значение этого поля должно быть 225, а блок заголовка JP2 должен также содержать блок битов на компонент, определяя глубину битов на каждый компонент (как это указано в 1.5.3.2).
 Низшие 7 битов значения показывают глубину битов компонентов. Высший бит показывает, имеют ли компоненты знаки. Если высший бит – это 0, то компоненты не имеют знаков. Если высший бит – это 1, то компоненты имеют знаки.
- C:** Тип компрессии (сжатия). Этот параметр показывает алгоритм компрессии, использованный для данных сжатого изображения. Величина этого поля должна быть 7. Оно кодировано однобайтовым целым числом без знака. Прочие величины зарезервированы для использования ИСО.
- UnkC:** Неизвестное пространство цвета. Это поле определяет, известны ли действительные данные пространства цвета изображения в кодированном потоке. Это поле кодировано однобайтовым целым без знака. Регулярная величина для этого поля – нуль, если пространство цвета изображения известно и правильно отражено в блоке спецификации пространства цвета внутри файла, или 1, если пространство цвета изображения неизвестно. Значение 1 должно использоваться в таких случаях, как перекодировка регулярных изображений, когда действительное пространство цвета данных изображения неизвестно. В таких случаях, несмотря на то что описанные в файле методы интерпретации пространства цвета могут неточно воспроизводить изображение по сравнению с оригиналом, изображение следует рассматривать так, как будто эти методы корректно воспроизвели изображение. Значения, отличные от 0 и 1, зарезервированы для использования ИСО.

IPR: Интеллектуальная собственность. Этот параметр показывает, содержит ли файл JP2 информацию о праве на интеллектуальную собственность. Если значение этого поля нулевое, этот файл не содержит информацию о праве и, следовательно, файл не содержит блок IPR. Если это значение 1, то этот файл содержит информацию о правах и, следовательно, содержит блок IPR, как это определено в I.6. Прочие значения зарезервированы для использования ИСО.

Таблица I.5 – Формат содержимого блока заголовка изображения

Наименование поля	Размер (биты)	Значение
HEIGHT	32	$1 - (2^{32} - 1)$
WIDTH	32	$1 - (2^{32} - 1)$
NC	16	1 – 16384
BPC	8	см. табл. I.6
C	8	7
Unk	8	0 – 1
IPR	8	0 – 1

Таблица I.6 – Значения BPC

Значения (биты)		Точность образца компонента
MSB	LSB	
x000 до x010	0000 0101	Глубина битов компонента = значение + 1. Соответственно, для глубины битов от 1 до 38 (считая знаковый бит, если это требуется)
0xxx	xxxx	Компоненты без знаков
1xxx	xxxx	Компоненты со знаками
1111	1111	Компоненты с переменной длиной глубины битов
		Все прочие значения зарезервированы для использования ИСО

I.5.3.2 Блок биты на компонент

Блок биты на компонент определяет битовую глубину каждого компонента. Если битовая глубина всех компонентов кодированного потока одинакова (по знаку и точности), тогда этот блок не должен обнаруживаться. В противном случае, этот блок определяет битовую глубину каждого отдельного компонента. Порядок значений битовой глубины в этом блоке соответствует действительному порядку, в котором эти компоненты перечислены в кодированном потоке. Точное расположение этого блока внутри блока заголовка JP2 может меняться при условии, что этот блок будет следовать за блоком заголовка изображения.

Внутри блока заголовка JP2 должен находиться один и только один блок битов на компонент.

Тип блока битов на компонент должен быть 'bpc' (0x6270 6363). Содержимое этого блока показано на рисунке I.9:

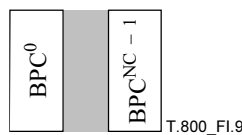


Рисунок I.9 – Организация содержимого блока биты на компонент

BPCⁱ: Биты на компонент. Этот параметр определяет битовую глубину компонента *i*, минус 1 и кодирован как однобайтовое значение. Порядок компонентов в блоке биты на компонент должен быть тем же самым, что и порядок в кодированном потоке. Число полей BPCⁱ должно быть таким же, что и NC из блока заголовка изображения. Значение этого поля должно быть эквивалентно соответствующему полю Ssizⁱ маркера SIZ в кодированном потоке.

Низшие 7 битов значения показывают глубину битов компонентов. Высший бит показывает, имеют ли компоненты знаки. Если высший бит – это 0, то компоненты не имеют знаков. Если высший бит – это 1, то компоненты имеют знаки.

Таблица I.7 – Форма содержимого блока биты на компонент

Наименование поля	Размер (биты)	Значение
ВРС ⁱ	8	см. табл. I.8

Таблица I.8 – Значения ВРСⁱ

Значения (биты)		Точность образца компонента
MSB	LSB	
x000 до x010	0000 0101	Глубина битов компонента = значение + 1. Соответственно, для глубины битов от 1 до 38 (считая знаковый бит, если это требуется)
0xxx	xxxx	Компоненты без знаков
1xxx	xxxx	Компоненты со знаками
		Все прочие значения зарезервированы для использования ИСО

I.5.3.3 Блок спецификации цвета

Каждый блок спецификации цвета определяет один метод, с помощью которого приложение может интерпретировать данные пространства цвета декомпрессированного изображения. Эту спецификацию цвета применяют к данным изображения после декомпрессии и после того, как над данными изображения будет произведено любое обратимое преобразование с декорреляцией компонента.

Файл JP2 может содержать множество блоков спецификации цвета, но в нем должен быть по крайней мере один блок, описывающий различные методы достижения "эквивалентных" результатов. Согласованный считыватель JP2 должен игнорировать все блоки спецификации цвета, кроме первого. Однако считыватель, согласованный с другими стандартами, может использовать такие блоки, как это определено в этих других стандартах.

Тип блока спецификации цвета должен быть "colr" (0x636F 6C72). Содержимое блока спецификации цвета показано на рисунке I.10:

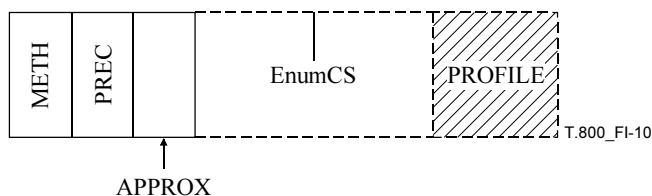


Рисунок I.10 – Организация содержимого блока спецификации цвета

МЕТН: Метод спецификации. Это поле описывает метод, использованный блоком спецификации цвета для определения пространства цвета декомпрессированного изображения. Это поле кодируют однобайтовым целым числом без знака. Значение этого поля должно быть 1 или 2, как это определено в таблице I.9.

Таблица I.9 – Разрешенные значения METH

Значение	Содержание
1	Нумерованное пространство цвета. Этот блок спецификации пространства цвета содержит нумерованную величину пространства цвета данного изображения. Нумерованная величина находится в поле EnumCS этого блока. Если значение поля METH, – это 1, то EnumCS должно следовать в этом блоке непосредственно за полем APPROX, а поле EnumCS должно быть последним полем в этом блоке.
2	Ограниченные характеристики ICC. Этот блок спецификации цвета содержит характеристики ICC в поле PROFILE. Эти характеристики должны описывать преобразование, необходимое для превращения данных декомпрессированного изображения в характеристики PCS _{XYZ} , и должны соответствовать либо входу черного-белого цвета, либо входу класса характеристик, основанных на трехкомпонентной матрице, и содержать все требуемые особые признаки (теги), определенные в ICC.1:1998-09. Значение поля пространственной характеристики соединения в заголовке характеристик должно быть 'XYZ\040' (0x5859 5A20), указывая на то, что выходное пространство цвета является пространством XYZ. Любые собственные теги в характеристиках ICC не должны изменять внешний вид изображения, обработанного этими характеристиками ICC. Компоненты кодированного потока могут иметь диапазон больший, чем входной диапазон кривой воспроизведения тона (TRC) характеристик ICC. Перед обработкой изображения характеристиками ICC любые декодированные значения следует ограничивать в пределах TRC. Например, отрицательные значения образцов со знаками могут быть ограничены до нуля перед обработкой данных изображения характеристиками ICC. См. J.9 для более подробного описания правил преобразований пространства цвета, хранения этих преобразований в файле и обработки изображения с помощью такого преобразования без использования машины управления цветом ICC. Если значение METH – это 2, то поле PROFILE должно непосредственно следовать за полем APPROX, а поле PROFILE должно быть последним полем в этом блоке.
Прочие величины	Зарезервированы для использования ИСО. Если значение METH – это не 1 или 2, то в этом блоке могут быть другие поля, следующие за полем APPROX, и согласованный считыватель JP2 должен игнорировать весь блок спецификации цвета.

PREC: Предпочтение. Это поле зарезервировано для использования ИСО, а его значение должно быть установлено на нуль. Однако согласованные считыватели должны игнорировать значение этого поля. Это поле описано однобайтовым целым числом со знаком.

APPROX: Аппроксимация пространства цвета. Это поле характеризует степень, с которой метод спецификации цвета аппроксимирует "корректность" определения пространства цвета. Значение этого поля должно быть установлено на нуль. Однако согласованные считыватели должны игнорировать значение этого поля. Это поле описано однобайтовым целым числом без знака. Прочие значения зарезервированы для использования ИСО.

EnumCS: Нумерованное пространство цвета. Это поле определяет пространство цвета изображения, используя целочисленные коды. Чтобы правильно интерпретировать цвет изображения с использованием нумерованного пространства цвета, приложение должно внутренне понимать определение пространства цвета. Для индикации пространства цвета изображения это поле содержит биты представления в виде 4-байтового целого числа без знака. Если значение поля METH – это 2, то поле EnumCS не существует. Действительные значения EnumCS для первого блока спецификации пространства цвета в соответствующих файлах ограничены значениями 16, 17 и 18, как это определено в таблице I.10:

Таблица I.10 – Разрешенные значения EnumCS

Значение	Содержание
16	sRGB, как определено в IEC 61966-2-1
17	Оттенки серого: Оттенки серого для случая, когда яркость изображения зависит от значений кода, который использует нелинейная система sRGB в равенстве (2) с помощью (4) спецификации IEC 61966-2-1 (sRGB): $Y' = Y_{8\text{ bit}} / 255 \quad (I-1)$ $\text{для } (Y' \leq 0,04045), Y_{lin} = Y' / 12,92 \quad (I-2)$ $\text{для } (Y' > 0,04045), Y_{lin} = \left(\frac{Y' + 0,055}{1,055} \right)^{2,4},$ <p>где Y_{lin} – значение яркости линейного изображения в диапазоне от 0,0 до 1,0. Значение яркости должно интерпретироваться в зависимости от условий в разделе 2 IEC 61966-2-1.</p>
18	YCC – как это определено в Поправке 1 IEC 61966-2-1 ПРИМЕЧАНИЕ. – Не рекомендуется использование ICT или RCT, определенных в Приложении G с данными изображения YCC. Для руководства обработки кодированного потока YCC см. J.15.
Прочие величины	Зарезервированы для использования ИСО.

PROFILE: Характеристики ICC. Это поле содержит действительные характеристики ICC, как указано в спецификации формата характеристик ICC, которые описывают преобразование данных декомпрессированного изображения в PCS. Это поле не должно существовать, если значение поля METH – 1. Если значение поля METH – 2, то характеристики ICC должны соответствовать классу характеристик монохромного входа или трехкомпонентной матрицы, определенных в ICC.1:1998-09.

Таблица I.11 – Формат содержимого блока спецификации цвета

Наименование поля	Размер (биты)	Значение
METH	8	от 1 до 2
PREC	8	0
APPROX	8	0
EnumCS	32, если METH = 1 0, если METH = 2	0 – $(2^{32} - 1)$ нет значения
PROFILE	Переменный	Переменное; см. Спецификацию формата характеристик ICC, версия ICC.1:1998-09.

I.5.3.4 Блок палитры

Этот блок определяет палитру, которая может быть использована для создания каналов из компонентов. Однако блок палитры не может определять создание никакого конкретного канала. Создание каналов, основанное на применении палитры к компоненту, определено блоком размещения компонентов. Пространство цвета или значение созданного канала определено блоком определения канала (или по умолчанию, как определено в спецификации блока определения канала, если такой блок существует). Если блок заголовка JP2 содержит блок палитры, то в нем не должно быть блока размещения компонента.

В блоке заголовка JP2 должен быть по крайней мере один блок палитры.

Тип блока палитры должен быть "pclr" (0x7063 6C72). Содержимое этого блока показано на рисунке I.11:

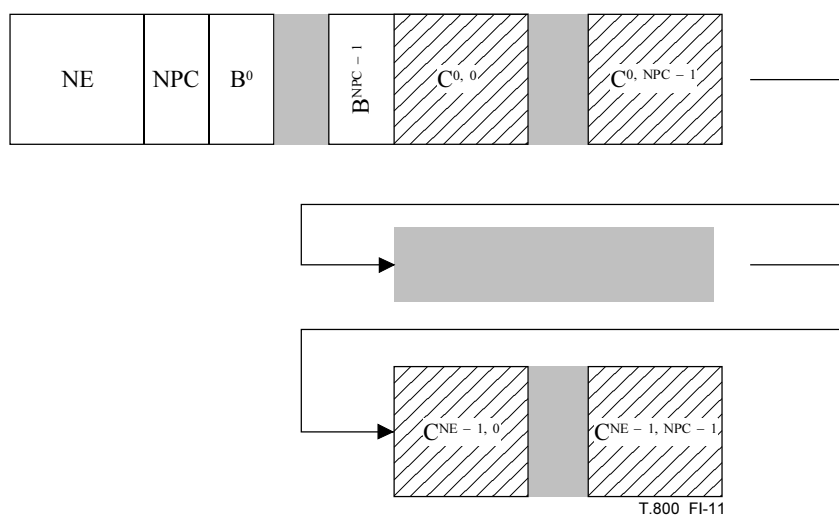


Рисунок I.11 – Организация содержимого блока палитры

- NE:** Число входов в таблице. Эта величина должна находиться в диапазоне от 1 до 1024, и ее кодируют как 2-байтовые биты представления целого без знака.
- NPC:** Число столбцов палитры, определенное в блоке палитры. Например, если палитра должна использоваться для размещения единственного компонента индекса в трехкомпонентном изображении RGB, то значение этого поля должно быть 3. Это поле кодируют однобайтовым целым без знака.
- Bⁱ:** Этот параметр определяет битовую глубину значений, создаваемых столбцом палитры *i*. Параметр кодируют однобайтовыми битами представления целого. Низшие 7 битов этого значения указывают на битовую глубину этого столбца палитры. Высший бит указывает, имеет ли столбец палитры знак. Если высший бит 1, столбец содержит значения со знаками. Если высший бит 0, то эти значения не имеют знака. Значения Bⁱ должны быть такими же, что и в поле NPC.

C^{ji} : Значение для входа j столбца палитры i . Значения C^{ji} организуют в порядке значимости компонента. Все из этих значений для входа j сгруппированы вместе. За ними следуют величины для входа $j + 1$. В вышеприведенном примере в таблице следует читать $R_1, G_1, B_1, R_2, G_2, B_2$ и т. д. Размер C^{ji} – это значение, определенное полем V^i . Число столбцов палитры должно быть таким же, как в поле NPC. Число значений C^{ji} должно равняться числу столбцов палитры (полей NPC), умноженному на число входов в палитре (NE). Если значение V^i не кратно 8, то каждое значение C^{ji} добавляются нулями для кратности 8-ми битам, а действительное значение запоминают в битах низшего разряда добавленного значения. Например, если значение V^i составляет 10 битов, то оригинальное значение C^{ji} должно запоминаться в низших 10 битах 16-битового поля.

Таблица I.12 – Формат содержимого блока палитры

Наименование поля	Размер (биты)	Значение
NE	16	от 1 до 1024
NPC	8	от 1 до 255
V^i	8	см. табл. I.13
C^{ji}	Переменный	Переменное

Таблица I.13 – Значения V^i

Значения (биты) MSB LSB	Точность образца столбца палитры
x000 0000 до x010 0101	Глубина столбца палитры = значение + 1. Соответственно, для глубины битов от 1 до 38 (считая знаковый бит, если это требуется)
0xxx xxxx	Величины столбца палитры без знаков
1xxx xxxx	Величины столбца палитры со знаками
	Все прочие значения зарезервированы для использования ИСО

I.5.3.5 Блок размещения компонента

Блок размещения компонента определяет, какие каналы изображения идентифицированы среди действующих компонентов, декодированных из кодированного потока. Эта абстракция предполагает простую структуру (блок определения канала) для определения цвета или типа как цветных, так и не цветных изображений. Этот блок содержит массив полей $СМР^i$, $МТУР^i$ и $РСОЛ^i$. Каждая группа этих полей представляет определение одного канала в изображении. Каналы нумеруют в порядке, начиная с нуля, а номер канала, описанного в блоке размещения компонентов, определяет длину блока.

В блоке заголовка JP2 должен быть по крайней мере один блок размещения компонентов.

Если блок заголовка JP2 содержит блок палитры, то блок заголовка JP2 должен также содержать блок размещения компонентов, а компоненты должны быть размещены непосредственно в каналы так, чтобы компонент i попадал в канал i .

Тип блока размещения компонентов должен быть "стар" (0x636D 6170). Содержимое этого блока показано на рисунке I.12:

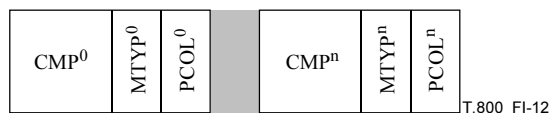


Рисунок I.12 – Организация содержимого блока размещения компонентов

- СМРⁱ:** Это поле определяет индекс компонента кодированного потока, который размещен в данном канале (либо непосредственно, либо с помощью палитры). Поле кодируют как 2-байтовое представление целого без знака.
- МТУРⁱ:** Это поле определяет, каким образом данный канал создают из действующих компонентов файла. Поле кодируют как однобайтовое целое число без знака. Разрешенные значения поля МТУРⁱ показаны в таблице I.14:

Таблица I.14 – Значения поля МТУРⁱ

Значение	Содержание
0	Непосредственное использование. Этот канал создан непосредственно из действующего компонента кодированного потока. Индекс компонента, отображенного в этом канале, описан в поле СМР ⁱ канала.
1	Размещение палитры. Этот канал создан применением палитры к действующему компоненту кодированного потока. Индекс компонента, отображенного в этой палитре, описан в поле СМР ⁱ канала. Столбец используемой палитры определен в поле PCOL ⁱ канала.
от 2 до 255	Зарезервировано для использования ИСО.

PCOLⁱ: Это поле определяет индекс компонента из палитры, который используют для отображения действующего компонента кодированного потока. Поле кодируют как однобайтовое целое число без знака. Если значение поля МТУРⁱ для этого канала 0, то и значение поля должно быть 0.

Таблица I.15 – Формат содержимого блока размещения компонента

Наименование поля	Размер (биты)	Значение
СМР ⁱ	16	от 0 до 16384
МТУР ⁱ	8	от 0 до 1
PCOL ⁱ	8	от 0 до 255

I.5.3.6 Блок определения канала

Блок определения канала описывает значения образцов в каждом канале изображения. Точное расположение этого блока в блоке заголовка JP2 может меняться в зависимости от того, какой блок следует за блоком заголовка изображения. Расположение между действующими компонентами кодированного потока, определено в блоке размещения компонентов. Если блок заголовка JP2 не содержит блок размещения компонентов, считыватель должен для всех компонентов кодированного потока поместить компонент *i* в канал *i*.

В блоке заголовка JP2 должен быть по крайней мере один блок размещения компонентов.

Этот блок содержит массив описаний каналов. Для каждого описания определены три значения: индекс канала, описанного в этом массиве, тип этого канала и соединение этого канала с конкретными цветами. Для одного канала в блоке может быть определено множество описаний.

Если в кодированном потоке определено многократное преобразование компонента, изображение должно находиться в пространстве цвета RGB, а красный, зеленый и синий – соответственно, в каналах 0, 1 и 2.

Тип блока определения канала должен быть "cdef" (0x6364 6566). Содержимое блока показано на рисунке I.13:

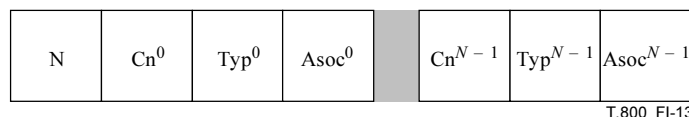


Рисунок I.13 – Организация содержимого блока определения канала

- N:** Число описаний канала. Это поле определяет число описаний канала в блоке. Поле кодируют как 2-байтовый бит представления целого числа без знака.
- Cnⁱ:** Индекс канала. Это поле определяет индекс канала для описания. Значение поля представляет индекс канала, как это определено в блоке размещения компонентов (или действующим компонентом кодированного потока, если в файле нет блока размещения компонентов). Поле кодируют как 2-байтовое представление целого числа без знака.
- Турⁱ:** Тип канала. Это поле определяет тип канала для описания. Значение поля определяет содержание декомпрессированных образцов в этом канале. Поле кодируют как 2-байтовое представление целого числа без знака. Разрешенные значения этого поля показаны в таблице I.16.

Таблица I.16 – Значения поля Турⁱ

Значение	Содержание
0	Это канал данных цвета изображения для соответствующего цвета.
1	Затененность. Значение образца 0 показывает, что образец прозрачен на 100%, а максимальное значение канала (связанное с битовой глубиной компонента кодированного потока или с компонентом палитры, размещенном в этом канале) показывает 100% затенение образца. Все затененные каналы должны быть отображены компонентами без знака.
2	Предварительно умноженное затенение. Канал затенения, как описано выше, за исключением того, что значение канала затенения умножают на канал цвета, с которым этот канал связан. Умножение в обратном порядке определяют следующим образом: $S_p = S \times \frac{\alpha}{\alpha_{max}}, \quad (I-3)$ где S – исходный образец, S_p – предварительно умноженный образец (образец, хранимый в изображении, α – значение затенения канала, а α_{max} – максимальная величина затенения канала, как это определено битовой глубиной затенения канала).
от 3 до ($2^{16}-2$)	Зарезервировано для использования ИСО.
$2^{16}-1$	Тип этого канала не определен.

- Asocⁱ:** Связь канала. Это поле определяет индекс цвета, с которым связан этот канал (или специальное значение для индикации всего изображения, или отсутствие связи). Например, если данный канал – затененный канал для красного в пространстве цвета RGB, то поле покажет индекс красного цвета. В таблице I.17 указаны разрешенные значения связи, а в таблице I.18 – разрешенные индексы цвета. Поле кодируют как 2-байтовое представление целого числа без знака.

Таблица I.17 – Значения поля Asocⁱ

Значение	Содержание
0	Этот канал связан с изображением как с целым (например, независимый затененный канал, который используют для всех каналов цвета).
от 1 до ($2^{16}-2$)	Этот канал связан с конкретным цветом, как указано его значением. Значение используют для связи конкретного канала с конкретным аспектом спецификации пространства цвета изображения. Например, указание, что канал связан с каналом красного в изображении RGB, позволит считывателю связать этот декодированный канал с входом красного в характеристиках ICC, которые содержит блок спецификации цвета. Индикаторы цвета приведены в таблице I.18.
$2^{16}-1$	Этот канал не связан ни с каким конкретным цветом.

Таблица I.18 – Цвета, указанные в поле Asocⁱ

Класс пространства цвета	Цвета, отмеченные следующими значениями в поле Asoc ⁱ			
	1	2	3	4
RGB	R	G	B	
Оттенки серого	Y			
YCbCr	Y	C _b	C _r	

Следующие классы пространств цвета перечислены для будущих применений, а также для помощи в понимании использования поля Asocⁱ :

XYZ	X	Y	Z	
Lab	L	a	b	
Luv	L	u	v	
YCbCr	Y	C _b	C _r	
Yxy	Y	x	y	
HSV	H	S	V	
HLS	H	L	S	
CMYK	C	M	Y	K
CMY	C	M	Y	
Jab	J	a	b	
<i>n</i> пространств цвета	1	2	3	4

Значения в таблице I.18 определяют индексы, которые присвоены для представления особых "цветов" и которые не относят к специальным каналам (или к компонентам кодированного потока, или к палитре). Считыватели должны использовать информацию, которую содержит блок определения канала, для определения того, какой цвет содержит каждый канал.

В этом блоке индексы каналов располагают исходя из конкретных компонентов кодированного потока или палитры. Индекс цвета определяет, как следует интерпретировать конкретный канал, основываясь на спецификации пространства цвета изображения. В этом блоке должно быть одно определение канала для каждого цвета, который затребован спецификацией пространства цвета этого файла, как указано в блоке спецификации пространства цвета.

Например, зеленый цвет изображения RGB определен набором {C_n, Typ, Asoc} со значениями {*i*, 0, 2}, где *i* – индекс этого канала (либо непосредственно, либо созданным с использованием инверсного многократного преобразования компонентов к действующим компонентам кодированного потока). Приложения, которые предназначены только для извлечения каналов цвета, могут обращаться с парой полей Typ/Asoc, как с 4-байтовым значением, где комбинированные значения преобразуют непосредственно в индексы цвета (поскольку тип поля для канала цвета должен быть 0).

В другом примере кодированный поток может содержать канал *i*, который определяет смешанную затененность образцов для красного и зеленого каналов, и канал *j*, который определяет смешанную затененность образцов для канала синего цвета. В этом файле набор {C_n, Typ, Asoc} должен находиться в блоке определения канала для двух затененных каналов: {*i*, 1, 1}, {*i*, 1, 2} и {*j*, 1, 3}.

В файле JP2 должно быть не более одного канала с одинаковой парой значений Typⁱ и Asocⁱ, за исключением этой пары со значениями 2¹⁶ – 1 (не отмечено). Например, файл JP2 в пространстве цвета RGB должен содержать только один зеленый канал, а изображение с оттенками серого – только канал серого цвета. Должен быть либо строго один затененный канал, точнее один затененный канал с предварительным умножением, либо никакой из каналов, связанных с простым каналом цвета в изображении.

Если кодированный поток содержит только каналы цвета и эти каналы расположены в том же самом порядке, что и связанные с ними цвета (например, изображение RGB с тремя каналами в последовательности R, G и затем B), то соответствующий блок не должен существовать. Если имеются любые дополнительные каналы или каналы, расположенные не в том же порядке, что индексы цвета, то блок определения канала (см. таблицу I.19) должен быть расположен в блоке заголовка JP2 с законченным списком определения каналов.

Таблица I.19 – Формат блока определения канала

Параметр	Размер (биты)	Значение
N	16	от 1 до $(2^{16} - 1)$
Cn ⁱ	16	от 0 до $(2^{16} - 1)$
Тип ⁱ	16	от 0 до $(2^{16} - 1)$
Asoc ⁱ	16	от 0 до $(2^{16} - 1)$

I.5.3.7 Блок разрешения (суперблок)

Этот блок определяет захват и отображение по умолчанию сетки разрешения данного изображения. Если такой блок существует, он должен содержать либо блок разрешения захвата, либо блок разрешения отображения по умолчанию, либо оба эти блока.

В блоке заголовка JP2 должен быть по крайней мере один блок разрешения.

Тип блока разрешения должен быть "res\040" (0x7265 7320). Содержимое блока показано на рисунке I.14:

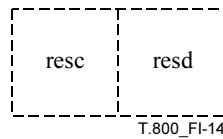


Рисунок I.14 – Организация содержимого блока разрешения

- resc:** Блок разрешения захвата. Этот блок определяет разрешение сетки, на которой происходит захват изображения. Формат этого блока определен в I.5.3.7.1.
- resd:** Блок разрешения отображения по умолчанию. Этот блок определяет разрешение (по умолчанию) сетки, на которую должно отображаться изображение. Формат этого блока определен в I.5.3.7.2.

I.5.3.7.1 Блок разрешения захвата

Этот блок определяет разрешение сетки, с которым источник переводят в цифровую форму, чтобы создать образцы изображения, определенные кодированным потоком. Например, таким образом может быть указан планшетный сканнер, который захватывает страницу книги. Разрешение сетки захвата может также определять разрешение подвесной цифровой камеры или спутниковой камеры.

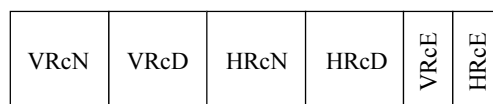
Разрешение сетки захвата по вертикали и горизонтали вычисляют, используя шесть параметров (см. табл. I.20), которые хранят в этом блоке в виде, соответственно, двух равенств:

$$VRc = \frac{VRcN}{VRcD} \times 10^{VRcE}, \quad (I-4)$$

$$HRc = \frac{HRcN}{HRcD} \times 10^{HRcE}. \quad (I-5)$$

Значения VRc и HRc всегда выражены в количестве точек опорной сетки на метр. Если в приложении требуется разрешение сетки в другой размерности, следует сделать соответствующее преобразование.

Тип блока разрешения захвата должен быть "resc" (0x7265 7363). Содержимое блока показано на рисунке I.15:



T.800_FI-15

Рисунок I.15 – Организация содержимого блока разрешения захвата

- VRcN:** Числитель разрешения сетки захвата по вертикали. Этот параметр определяет значение $VRcN$ в равенстве (I-4), которое используют для вычисления разрешения сетки захвата по вертикали. Параметр кодируют как 2-байтовое представление целого числа без знака.
- VRcD:** Знаменатель разрешения сетки захвата по вертикали. Этот параметр определяет значение $VRcD$ в равенстве (I-4), которое используют для вычисления разрешения сетки захвата по вертикали. Этот параметр кодируют как 2-байтовое представление целого числа без знака.
- HRcN:** Числитель разрешения сетки захвата по горизонтали. Этот параметр определяет значение $HRcN$ в равенстве (I-5), которое используют для вычисления разрешения сетки захвата по горизонтали. Этот параметр кодируют как 2-байтовое представление целого числа без знака.
- HRcD:** Знаменатель разрешения сетки захвата по горизонтали. Этот параметр определяет значение $HRcD$ в равенстве (I-5), которое используют для вычисления разрешения сетки захвата по горизонтали. Этот параметр кодируют как 2-байтовое представление целого числа без знака.
- VRcE:** Порядок разрешения сетки захвата по вертикали. Этот параметр определяет значение $VRcE$ в равенстве (I-4), которое используют для вычисления разрешения сетки захвата по вертикали. Этот параметр кодируют как двойное дополнение однобайтового целого числа со знаком.
- HRcE:** Порядок разрешения сетки захвата по горизонтали. Этот параметр определяет значение $HRcE$ в равенстве (I-5), которое используют для вычисления разрешения сетки захвата по горизонтали. Этот параметр кодируют как двойное дополнение однобайтового целого числа со знаком.

Таблица I.20 – Формат содержимого блока разрешения захвата

Наименование поля	Размер (биты)	Значение
VRcN	16	$1 - (2^{16} - 1)$
VRcD	16	$1 - (2^{16} - 1)$
HRcN	16	$1 - (2^{16} - 1)$
HRcD	16	$1 - (2^{16} - 1)$
VRcE	8	от -128 до 127
HRcE	8	от -128 до 127

I.5.3.7.2 Блок разрешения отображения по умолчанию

Это блок определяет требуемое разрешение сетки дисплея. Например, это можно использовать для определения размеров изображения страницы, если изображение поместить в разложенную на страницы программу. Однако это – только значение по умолчанию. Каждое приложение должно определить соответствующий конкретный размер дисплея.

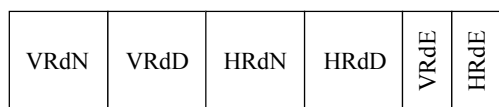
Разрешение сетки отображения по вертикали вычисляют, используя шесть параметров (см. таблицу I.21), которые хранят в этом блоке в виде, соответственно, двух равенств:

$$VRd = \frac{VRdN}{VRdD} \times 10^{VRdE}, \tag{I-6}$$

$$HRd = \frac{HRdN}{HRdD} \times 10^{HRdE}. \tag{I-7}$$

Значения VRd и HRd всегда выражены в количестве точек опорной сетки на метр. Если в приложении требуется разрешение сетки в другой размерности, следует сделать соответствующее преобразование.

Тип блока разрешения отображения по умолчанию должен быть "resd" (0x7265 7364). Содержимое блока показано на рисунке I.16:



T.800_FI-16

Рисунок I.16 – Организация содержимого блока разрешения отображения по умолчанию

- VRdN:** Числитель разрешения сетки отображения по вертикали. Этот параметр определяет значение VRdN в равенстве (I-6), которое используют для вычисления разрешения сетки отображения по вертикали. Этот параметр кодируют как 2- байтовое представление целого числа без знака.
- VRdD:** Знаменатель разрешения сетки отображения по вертикали. Этот параметр определяет значение VRdN в равенстве (I-6), которое используют для вычисления разрешения сетки отображения по вертикали. Этот параметр кодируют как 2- байтовое представление целого числа без знака.
- HRdN:** Числитель разрешения сетки отображения по горизонтали. Этот параметр определяет значение HRdN в равенстве (I-7), которое используют для вычисления разрешения сетки отображения по горизонтали. Этот параметр кодируют как 2- байтовое представление целого числа без знака.
- HRdD:** Знаменатель разрешения сетки отображения по горизонтали. Этот параметр определяет значение HRdD в равенстве (I-7), которое используют для вычисления разрешения сетки отображения по горизонтали. Этот параметр кодируют как 2- байтовое представление целого числа без знака.
- VRdE:** Порядок разрешения сетки отображения по вертикали. Этот параметр определяет значение VRdE в равенстве (I-6), которое используют для вычисления разрешения сетки отображения по вертикали. Этот параметр кодируют как двойное дополнение однобайтового целого числа со знаком.
- HRdE:** Порядок разрешения сетки отображения по горизонтали. Этот параметр определяет значение HRdE в равенстве (I-7), которое используют для вычисления разрешения сетки отображения по горизонтали. Этот параметр кодируют как двойное дополнение однобайтового целого числа со знаком.

Таблица I.21 – Формат содержимого блока разрешения отображения по умолчанию

Наименование поля	Размер (биты)	Значение
VRdN	16	$1 - (2^{16} - 1)$
VRdD	16	$1 - (2^{16} - 1)$
HRdN	16	$1 - (2^{16} - 1)$
HRdD	16	$1 - (2^{16} - 1)$
VRdE	8	от -128 до 127
HRdE	8	от -128 до 127

I.5.4 Блок непрерывного кодированного потока

Блок непрерывного кодированного потока содержит действительный и полный кодированный поток JPEG 2000, определенный в Приложении А. При отображении изображения согласованный считыватель может проигнорировать все потоки после того, как обнаружит первый поток в файле. Блоки непрерывного кодированного потока могут располагаться в файле везде, кроме позиции перед блоком заголовка JP2.

Тип блока непрерывного кодированного потока должен быть "jp2c" (0x6A70 3263). Содержимое блока показано на рисунке I.17:

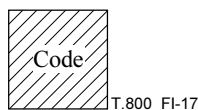


Рисунок I.17 – Организация содержимого блока непрерывного кодированного потока

Code: Это поле содержит действительный и полный кодированный поток JPEG 2000, определенный в Приложении А.

Таблица I.22 – Формат содержимого блока непрерывного кодированного потока

Наименование поля	Размер (биты)	Значение
Code	Переменный	Переменное

I.6 Дополнительная информация в файле JP2 о правах на интеллектуальную собственность

Данная Рекомендация | Международный стандарт определяет тип блока в файле JP2, который предназначен для информации о правах на интеллектуальную собственность. Для файлов согласования включение этой информации в файл JP2 является дополнительным. Определение формата содержания этого блока зарезервировано для ИСО. Однако тип этого блока определен в данной Рекомендации | Международном стандарте как средство, позволяющее приложениям распознавать существование информации IPR. Использование и интерпретация этой информации выходят за рамки данной Рекомендации | Международного стандарта.

В общем случае блок IPR находится на верхнем уровне файла, определяющего в целом IPR для этого файла. Блоки IPR могут находиться и в других местах, включая внутренние части суперблоков, которые определены другими Рекомендациями | Международными стандартами. Для таких блоков эти права относят к объектам, определенным суперблоком, в котором они находятся.

Тип блока интеллектуальной собственности должен быть "jp2i" (0x6A70 3269).

I.7 Дополнение к формату файла JP2 специальной информации для продавцов

Следующие блоки дадут ряд инструментов, с помощью которых в приложения можно добавить к формату файла JP2 информацию специально для продавцов. Все последующие блоки с точки зрения согласованности файлов являются опциональными, и согласованные считыватели могут игнорировать такие блоки.

I.7.1 Блоки XML

Специально для продавцов в блок XML включают информацию (в формате XML), отличную от информации в блоках, определенных данной Рекомендацией | Международным стандартом. В файле может быть много блоков XML, и эти блоки могут находиться в любом месте файла, но не перед блоком типа файла.

Тип блока XML: "xml\040" (0x786D 6C20). Содержимое блока показано на рисунке I.18:



Рисунок I.18 – Организация содержимого блока XML

DATA: Это поле должно содержать правильно оформленный документ XML, определенный в REC-xml-19980210.

Для согласованных файлов существование любых блоков XML является опциональным. Кроме того, любой блок XML не должен содержать никакой информации, необходимой для декодирования изображения в такой степени, как это определено в данной Рекомендации | Международном стандарте, а корректная интерпретация содержимого любого блока XML не должна изменять внешний вид изображения. Все считыватели могут игнорировать любой блок XML в файле.

I.7.2 Блоки UUID

Специально для продавца в блок UUID включают информацию, отличную от информации в блоках, определенных данной Рекомендацией | Международным стандартом. В файле может быть множество блоков UUID, и эти блоки могут находиться в любом месте файла, но не перед блоком типа файла.

Тип блока UUID должен быть "uuid" (0x7575 6964). Содержимое блока показано на рисунке I.19:

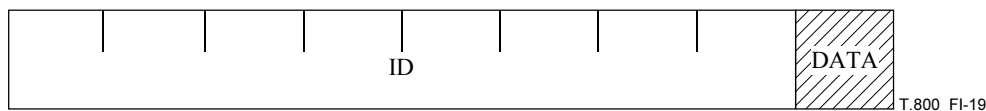


Рисунок I.19 – Организация содержимого блока UUID

ID: Это поле содержит 16-байтовый блок UUID, определенный ИСО/МЭК 11578. Значение UUID определяет формат специальной информации для продавца, которая хранится в поле DATA, а также интерпретацию этой информации.

DATA: Это поле содержит специальную информацию для продавца. Формат этой информации определен вне данной Рекомендации | Международного стандарта, но обозначен значением поля UUID.

Таблица I.23 – Формат содержимого блока UUID

Наименование поля	Размер (биты)	Значение
UUID	128	Переменное
DATA	Переменный	Переменное

Для согласованных файлов существование любых блоков UUID является опциональным. Кроме того, любой блок UUID не должен содержать никакой информации, необходимой для декодирования изображения в такой степени, как это определено в данной Рекомендации | Международном стандарте, а корректная интерпретация содержимого любого блока UUID не должна изменять внешний вид изображения. Все считыватели могут игнорировать любой блок UUID в файле.

I.7.3 Блоки UUID Info (суперблок)

Кроме того, что для продавцов полезно расширение файлов JP2 добавлением информации с помощью блоков, полезно также дать информацию в стандартной форме, которая могла бы использоваться приложениями без расширений, чтобы дать дополнительную информацию о расширениях в файле. Эта информация содержится в блоках UUID Info. Файл JP2 может содержать нуль либо несколько блоков UUID Info. Эти блоки могут находиться в любом месте верхнего уровня файла (суперблок UUID Info должен быть самым файлом JP2), но не перед блоком типа файла.

Эти блоки (если они присутствуют) могут не предоставлять полный индекс для блоков UUID в этом файле, не использовать ссылок на все блоки UUID, но могут многократно давать ссылки на один и тот же блок UUID.

Тип блока UUID Info должен быть "uinfi" (0x7569 6E66). Содержимое блока показано на рисунок I.20:

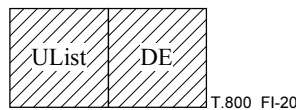


Рисунок I.20 – Организация содержимого блока UUID info

UList: Блок перечня блоков UUID. Этот блок содержит перечень блоков UUID, для которых данный блок UUID Info определяет канал связи к дополнительной информации. Формат блока перечня блоков UUID определен в I.7.3.1.

DE: Блок входных данных URL. Этот блок содержит указатель URL. Приложение может получить больше информации о блоках UUID, которая содержится в блоке перечня блоков UUID. Формат блока входных данных URL определен в I.7.3.2.

I.7.3.1 Блок перечня UUID

Этот блок содержит перечень блоков UUID. Тип блока перечня блоков UUID должен быть 'ulst' (0x756C 7374). Содержимое блока показано на рисунке I.21:



Рисунок I.21 – Организация содержимого блока перечня UUID

NU: Число блоков UUID. Это поле определяет число блоков UUID, которые содержатся в блоке перечня UUID. Это поле кодируют как 2-байтовое представление целого числа без знака.

IDⁱ: ID. Это поле определяет один блок UUID по спецификации ИСО/МЭК 11578, который должен быть связан с URL. Указатель URL находится в блоке URL внутри того же самого блока UUID Info. Число полей UUID_i должно быть тем же самым, что и значение поля NU. Значением этого поля должен быть 16-байтовый блок UUID.

Таблица I.24 – Значения структуры данных содержимого блока перечня UUID

Параметр	Размер (биты)	Значение
NU	16	0 – (2 ¹⁶ – 1)
UUID ⁱ	128	0 – (2 ¹²⁸ – 1)

I.7.3.2 Блок входных данных URL

Этот блок содержит указатель URL, который может использовать приложение для получения дополнительной информации, связанной с продавцом, в виде специального расширения. Формат информации, полученной с помощью URL, не определен в данной Рекомендации | Международном стандарте. Тип указателя URL должен зависеть от службы, которая предоставляет файл (например, файл типа URL, http, ftp и т. п.) и которая наилучшим образом разрешает доступ. Допустимы относительные указатели URL, связанные с файлом, в котором находится этот блок входных данных URL.

Тип блока входных данных URL должен быть "url\040" (0x7572 6C20). Содержимое блока показано на рисунке I.22:



Рисунок I.22 – Организация содержимого блока входных данных URL

- VERS:** Номер версии. Это поле определяет номер версии формата этого блока. Поле кодируют как однобайтовое целое число без знака. Значение этого поля должно быть 0.
- FLAG:** Флаги. Это поле зарезервировано для других применений атрибутов конкретного флага этого блока. Флаг кодируют как трехбайтовое целое число без знака. Значение этого поля должно быть 0.
- LOC:** Местоположение. Это поле определяет указатель URL дополнительной информации, связанной с блоками UUID, которые находятся в блоке перечня UUID внутри того же суперблока UUID Info. Указатель URL кодируют знаками UTF-8 в виде нулевой конечной строки.

Таблица I.25 – Значения структуры данных содержимого блока входных данных URL

Параметр	Размер (биты)	Значение
VERS	8	0
FLAG	24	0
LOC	Переменный	Переменное

I.8 Обращение с неизвестными блоками

Согласованный файл JP2 может содержать блоки, неизвестные приложениям, которые основаны исключительно на данной Рекомендации | Международном стандарте. Если согласованный считыватель обнаружит блок, который не может распознать, этот блок следует пропустить или игнорировать.

Приложение J

Примеры и руководства

(Это Приложение не является формой составной части данной Рекомендации | Международного стандарта)

В это Приложение включен ряд примеров, которые показывают, как работает процесс кодирования и как образуется на выходе кодированный поток. Это Приложение исключительно информативное.

J.1 Условные обозначения в программах адаптивного статистического декодера

В этом Приложении приведены некоторые альтернативные блок-схемы для одного варианта адаптивного статистического декодера. Этот альтернативный вариант может оказаться более эффективным, если его применять в виде программ, поскольку это требует меньшего числа операций по скоростным линиям.

Альтернативный вариант достигается с помощью следующих подстановок. Заменяем блок-схему на рисунке С.20 блок-схемой на рисунке J.1, блок-схему на рисунке С.15 – блок-схемой на рисунке J.2, блок-схему на рисунке С.19 – блок-схемой на рисунке J.3.

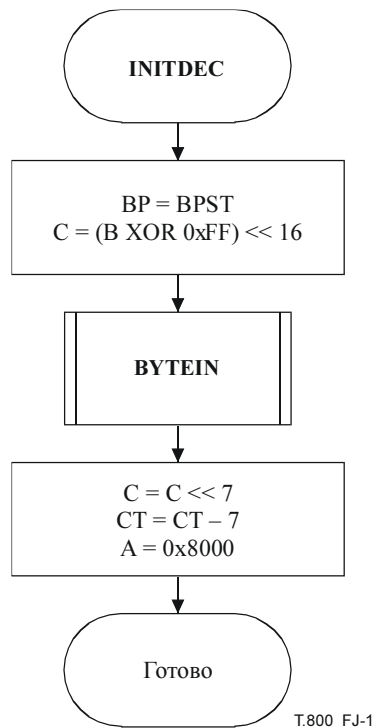


Рисунок J.1 – Инициализация программы условных обозначений декодера

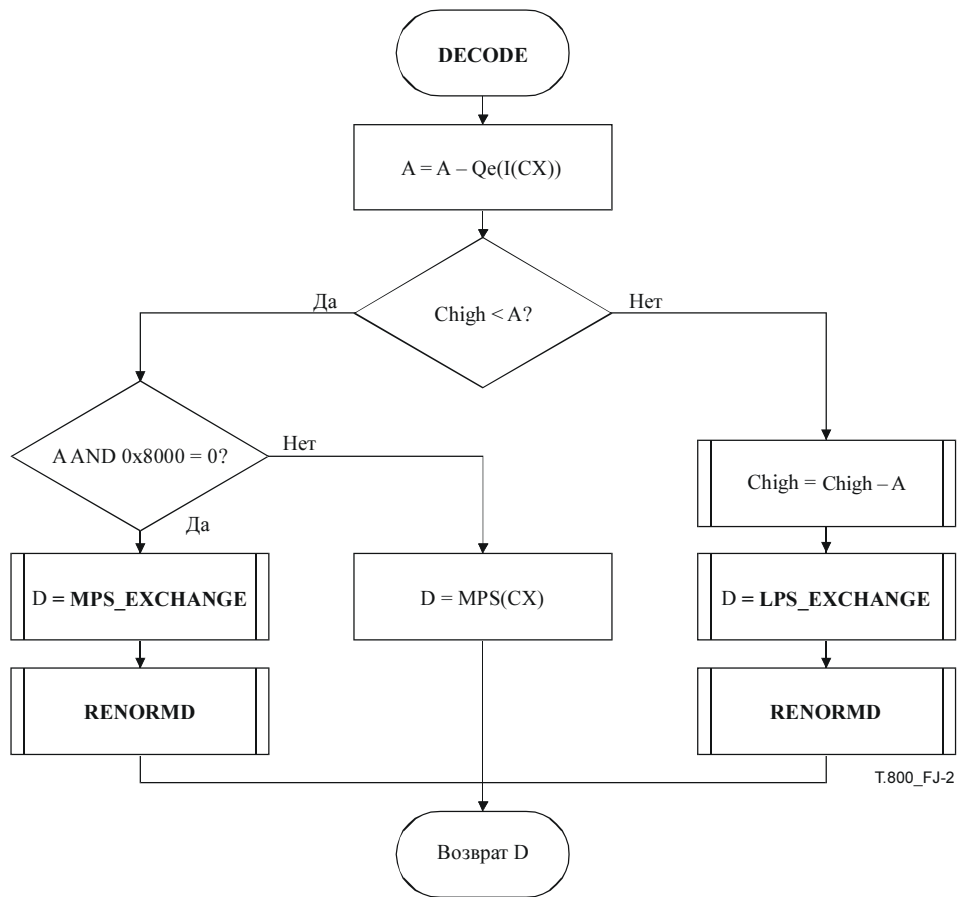


Рисунок J.2 – Декодирование MPS или LPS в программе условных обозначений декодера

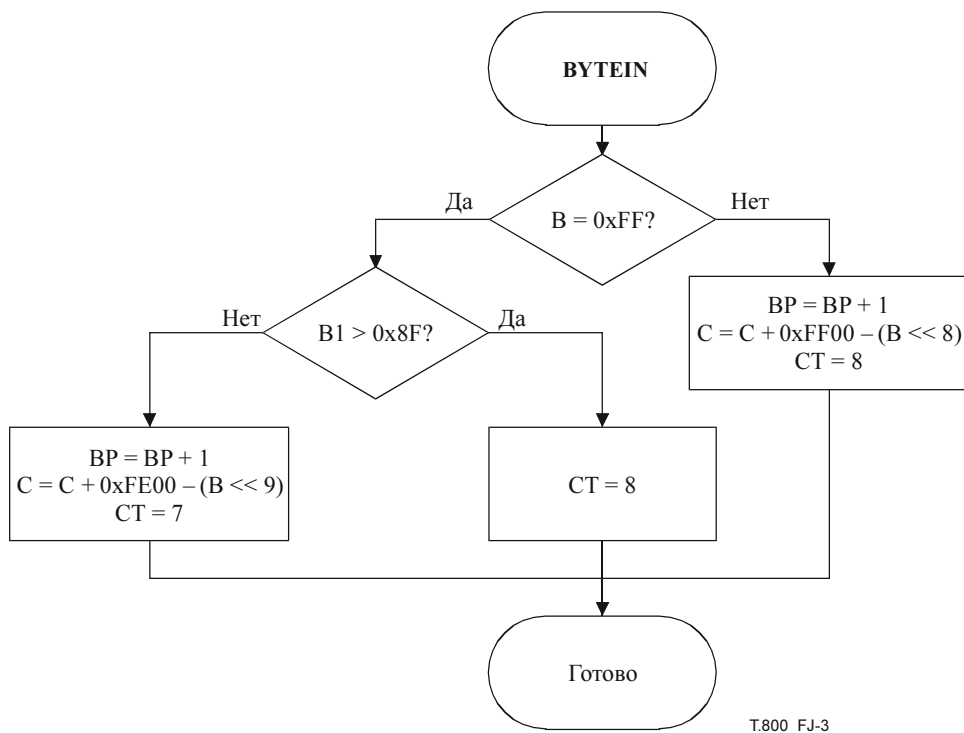


Рисунок J.3 – Введение нового байта в регистр C в программе условных обозначений декодера

J.2 Выбор размера шагов квантования для необратимых преобразований

Для необратимого сжатия в данной Рекомендации | Международном стандарте не требуется никакого выбора размера шагов квантования. Различные приложения могут выбирать шаги квантования в соответствии со спецификой характеристик компонентов тайла. Один из эффективных способов выбора размеров шагов квантования для каждого поддиапазона b заключается в масштабировании по умолчанию размера шага Δ_d , принимая во внимание процедуры горизонтальной и вертикальной фильтрации, с помощью которых получают эти коэффициенты поддиапазонов. Один метод состоит в масштабировании Δ_d с помощью параметра энергетического взвешивания γ_b (величина квадрата ошибок, полученная при введении единичной ошибки в преобразованный коэффициент поддиапазона b) следующим образом [12]:

$$\Delta_b = \frac{\Delta_d}{\sqrt{\gamma_b}}. \quad (J-1)$$

J.3 Импульсная характеристика фильтра, соответствующая процедурам необратимой фильтрации с приподнятым основанием

Процедуры необратимой фильтрации, описанные в Приложении F, используют блок фильтров с 9/7-ю отводами и сверткой по Cohen-Daubechies-Feauveau [20], [21]. Эквивалентные импульсные характеристики для анализа и синтеза фильтров приведены в таблицах J.1 и J.2.

Таблица J.1 – Определение импульсных характеристик для анализа блока необратимых фильтров 9-7

n	Фильтр низкой частоты	Приблизительная величина
0	$-5x_1(48 x_2 ^2 - 16\Re x_2 + 3) / 32$	0,602 949 018 236 360
± 1	$-5x_1(8 x_2 ^2 - \Re x_2) / 8$	0,266 864 118 442 875
± 2	$-5x_1(4 x_2 ^2 - 4\Re x_2 - 1) / 16$	-0,078 223 266 528 990
± 3	$-5x_1(\Re x_2) / 8$	-0,016 864 118 442 875
± 4	$-5x_1 / 64$	0,026 748 757 410 810
n	Фильтр высокой частоты	Приблизительная величина
-1	$(6x_1 - 1) / 8x_1$	1,115 087 052 457 000
-2, 0	$-(16x_1 - 1) / 32x_1$	-0,591 271 763 114 250
-3, 1	$(2x_1 + 1) / 16x_1$	-0,057 543 526 228 500
-4, 2	$-1 / 32x_1$	0,091 271 763 114 250

Таблица J.2 – Определение импульсных характеристик для синтеза блока необратимых фильтров 9-7

<i>n</i>	Фильтр низкой частоты	Приблизительная величина
0	$(6x_1 - 1) / 8x_1$	1,115 087 052 457 000
±1	$(16x_1 - 1) / 32x_1$	0,591 271 763 114 250
±2	$(2x_1 + 1) / 16x_1$	-0,057 543 526 228 500
±3	$1 / 32x_1$	-0,091 271 763 114 250
<i>n</i>	Фильтр высокой частоты	Приблизительная величина
1	$-5x_1(48 x_2 ^2 - 16\Re x_2 + 3) / 32$	0,602 949 018 236 360
0, 2	$5x_1(8 x_2 ^2 - \Re x_2) / 8$	-0,266 864 118 442 875
-1, 3	$-5x_1(4 x_2 ^2 - 4\Re x_2 - 1) / 16$	-0,078 223 266 528 990
-2, 4	$5x_1(\Re x_2) / 8$	0,016 864 118 442 875
-3, 5	$-5x_1 / 64$	0,026 748 757 410 811

J.4 Пример дискретного частотного преобразования

В таблице J.3 приведены целочисленные образцы $I(x, y)$ компонента тайла, у которого 13 образцов по ширине и 17 по высоте.

Таблица J.3 – Образцы компонентов источника тайла

$I(x, y)$	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	2	2	3	4	5	6	7	8	9	10	11	12
3	3	3	3	4	5	5	6	7	8	9	10	11	12
4	4	4	4	5	5	6	7	8	8	9	10	11	12
5	5	5	5	5	6	7	7	8	9	10	11	12	13
6	6	6	6	6	7	7	8	9	10	10	11	12	13
7	7	7	7	7	8	8	9	9	10	11	12	13	13
8	8	8	8	8	8	9	10	10	11	12	12	13	14
9	9	9	9	9	9	10	10	11	12	12	13	14	15
10	10	10	10	10	10	11	11	12	12	13	14	14	15
11	11	11	11	11	11	12	12	13	13	14	14	15	16
12	12	12	12	12	12	13	13	13	14	15	15	16	16
13	13	13	13	13	13	13	14	14	15	15	16	17	17
14	14	14	14	14	14	14	15	15	16	16	17	17	18
15	15	15	15	15	15	15	16	16	17	17	18	18	19
16	16	16	16	16	16	16	17	17	17	18	18	19	20

J.4.1 Пример необратимого частотного преобразования 9-7

В таблицах J.4, J.5, J.6, J.7, J.8, J.9 и J.10 приведены коэффициенты поддиапазонов 2LL, 2HL, 2LH, 2HH, 1HL, 1LH и 1HH, которые получают в результате двухуровневой декомпозиции с необратимым преобразованием 9-7 образцов компонентов источника тайла. Образцы компонентов приведены в таблице J.3 (см. рисунок F.18). Величины показанных в таблицах коэффициентов были округлены до ближайших целых.

**Таблица J.4 – Коэффициенты поддиапазона 2LL
(необратимая частотная трансформация 9-7)**

$a_{2LL}(u, v)$	0	1	2	3
0	1	4	8	11
1	4	5	8	11
2	8	9	11	13
3	12	12	14	16
4	15	15	17	18

**Таблица J.5 – Коэффициенты поддиапазона 2HL
(необратимая частотная трансформация 9-7)**

$a_{2HL}(u, v)$	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

**Таблица J.6 – Коэффициенты поддиапазона 2LN
(необратимая частотная трансформация 9-7)**

$a_{2LN}(u, v)$	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

**Таблица J.7 – Коэффициенты поддиапазона 2NN
(необратимая частотная трансформация 9-7)**

$a_{2NN}(u, v)$	0	1	2
0	-1	0	0
1	0	1	0
2	0	0	0
3	0	0	0

**Таблица J.8 – Коэффициенты поддиапазона 1NL
(необратимая частотная трансформация 9-7)**

$a_{1NL}(u, v)$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	-1	0	0	0	0
4	0	0	0	-1	0	0
5	0	0	1	1	0	-1
6	0	0	0	0	0	0
7	0	0	-1	-1	-1	0
8	0	0	0	0	0	0

**Таблица J.9 – Коэффициенты поддиапазона 1LN
(необратимая частотная трансформация 9-7)**

$a_{1LN}(u, v)$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	0	-1	0	0	0
2	0	0	0	0	0	1	1
3	0	0	0	0	-1	1	0
4	0	0	0	0	0	0	1
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

**Таблица J.10 – Коэффициенты поддиапазона 1NN
(необратимая частотная трансформация 9-7)**

$a_{1NN}(u, v)$	0	1	2	3	4	5
0	-1	0	0	0	0	0
1	0	0	-1	0	0	0
2	0	-1	1	0	0	0
3	0	0	0	0	0	1
4	0	0	0	0	-1	0
5	0	0	0	0	0	0
6	0	0	-1	0	-1	1
7	0	0	0	0	-1	0

J.4.2 Пример обратимого частотного преобразования 5-3

В таблицах J.11, J.12, J.13, J.14, J.15, J.16 и J.17 показаны коэффициенты поддиапазонов 2LL, 2HL, 2LN, 2NN, 1NL, 1LN и 1NN. Коэффициенты получают в результате двухуровневой декомпозиции с обратимым преобразованием 5-3 образцов компонентов источника тайла, приведенных в таблице J.3 (см. рисунок F.18). Величины показанных в таблицах коэффициентов были округлены до ближайших целых.

**Таблица J.11 – Коэффициенты поддиапазона 2LL
(обратимое частотное преобразование 5-3)**

$a_{2LL}(u, v)$	0	1	2	3
0	0	4	8	12
1	4	5	8	12
2	8	8	11	15
3	12	12	14	18
4	16	16	18	20

**Таблица J.12 – Коэффициенты поддиапазона 2HL
(обратимое частотное преобразование 5-3)**

$a_{2HL}(u, v)$	0	1	2
0	0	0	0
1	0	1	0
2	0	1	0
3	0	0	1
4	0	0	0

**Таблица J.13 – Коэффициенты поддиапазона 2LN
(обратимое частотное преобразование 5-3)**

$a_{2LN}(u, v)$	0	1	2	3
0	0	0	0	0
1	0	1	1	1
2	0	0	0	0
3	0	0	0	0

**Таблица J.14 – Коэффициенты поддиапазона 2NN
(обратимое частотное преобразование 5-3)**

$a_{2NN}(u, v)$	0	1	2
0	-1	0	0
1	0	-1	0
2	0	1	0
3	0	0	0

**Таблица J.15 – Коэффициенты поддиапазона 1NL
(обратимое частотное преобразование 5-3)**

$a_{1NL}(u, v)$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	1	0	1	0	0
3	0	0	0	0	-1	1
4	0	0	0	0	1	1
5	0	0	1	1	0	-1
6	0	0	1	0	1	1
7	0	0	0	0	0	0
8	0	0	0	0	0	0

**Таблица J.16 – Коэффициенты поддиапазона 1LN
(обратимое частотное преобразование 5-3)**

$a_{1LN}(u, v)$	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
2	0	0	0	0	0	1	1
3	0	0	1	0	0	1	1
4	0	0	0	0	1	0	2
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	1
7	0	0	0	0	1	1	0

Таблица J.17 – Коэффициенты поддиапазона 1НН
(обратимое частотное преобразование 5-3)

$a_{1НН}(u, v)$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	1	0	1	0
3	0	0	0	0	0	1
4	0	0	0	0	0	1
5	0	0	0	1	0	0
6	0	0	0	0	0	1
7	0	0	0	0	-1	0

J.5 Частотное преобразование по рядам

Описанное здесь частотное преобразование по рядам для необратимого фильтра 9-7 хорошо подходит для устройств сжатия, которые получают и передают данные изображения последовательным образом. Использование традиционных частотных преобразований требует буферизации всего изображения и поэтому необходима фильтрация в горизонтальном и вертикальном направлениях. В то время как фильтрация в горизонтальном направлении весьма проста, фильтрация в вертикальном направлении более сложная. Для фильтрации ряда требуется прочитать один ряд; для фильтрации по столбцу требуется прочитать все изображение. Это объясняет требования к большой полосе при использовании традиционного частотного преобразования. Частотное преобразование по рядам позволяет преодолеть прежние ограничения, в то же время получая точно такие же коэффициенты преобразования, что и при использовании традиционного частотного преобразования. Однако одно лишь частотное преобразование по рядам не обеспечивает полной системы кодирования по рядам. Законченный кодер для кодирования по рядам должен, кроме того, учитывать все последующие стадии кодирования, вплоть до стадий статистического кодирования и распределения скоростей.

J.5.1 Процедура FDWT_ROW

Процедура FDWT_ROW для необратимого фильтра 9-7 использует один буфер $buf(i, j)$ из пяти рядов $0 \leq i \leq 4$, чтобы выполнить процедуру, эквивалентную описанной в F.4.2 процедуре 2D_SD, за исключением процедуры 2D_DEINTERLEAVE. Предполагается, что диапазон образцов входного компонента тайла $I(x, y)$ определен равенством $(F-1)$. Каждый ряд буфера $buf(i, j)$ имеет размер $tcx_1 - tcx_0 + 1$. Общее описание процедуры FDWT_ROW над одним компонентом тайла для первого уровня декомпозиции иллюстрируется на рисунке J.4. Процедура FDWT_ROW принимает на входе ряд образцов компонентов тайла изображения со смещенным уровнем и выдает на выходе ряд коэффициентов преобразования. В этом разделе предполагается, что компонент тайла изображения занимает, по крайней мере, пять рядов.

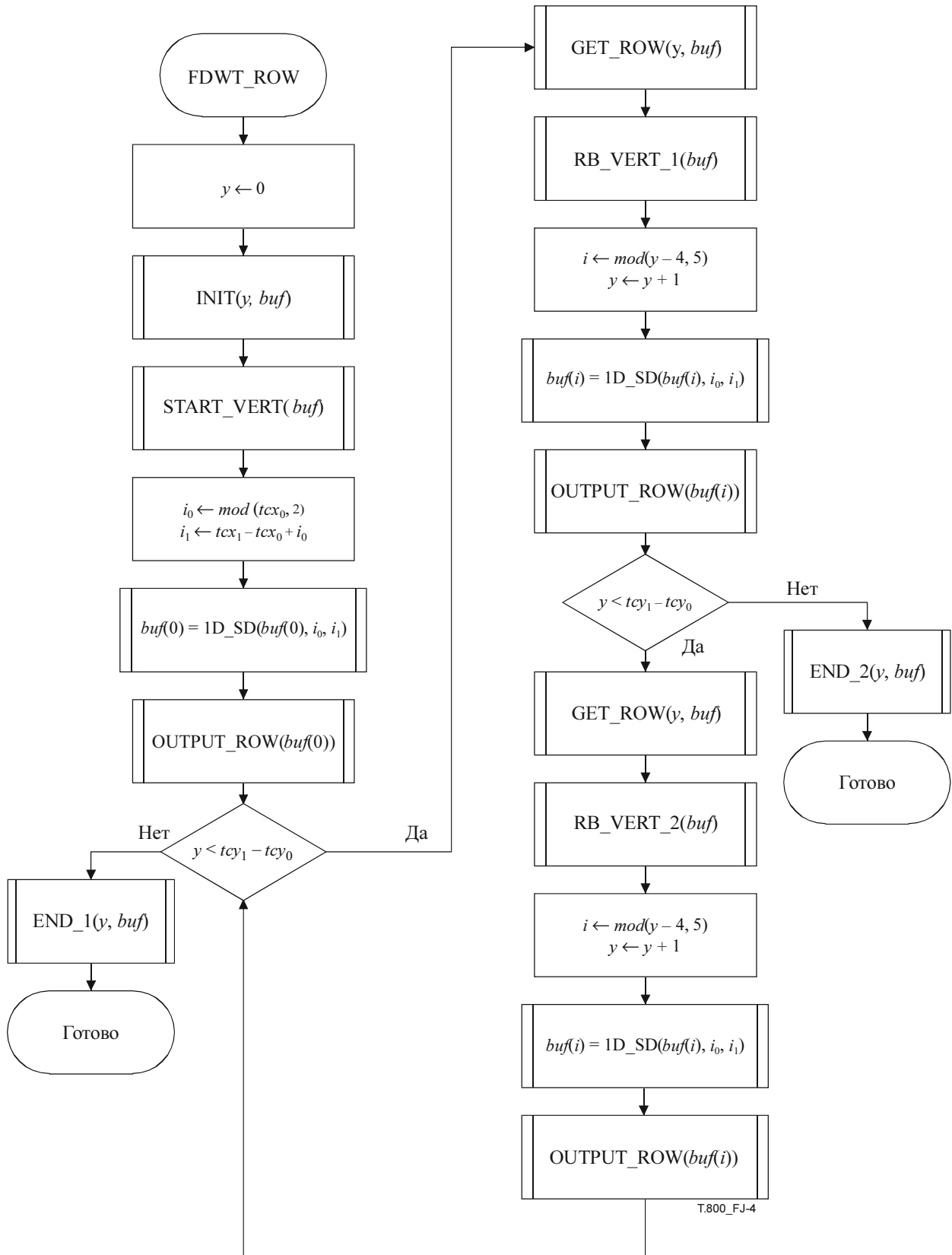


Рисунок J.4 – Процедура FDWT_ROW

J.5.1.1 Процедура GET_ROW

В этом описании предполагается, что компонент тайла изображения со смещенным уровнем хранят во внешней памяти $I(x, y)$. Как показано на рисунке J.5, процедура GET_ROW считывает один ряд образцов компонента тайла изображения со смещенным уровнем и передает этот ряд образцов в буфер buf .

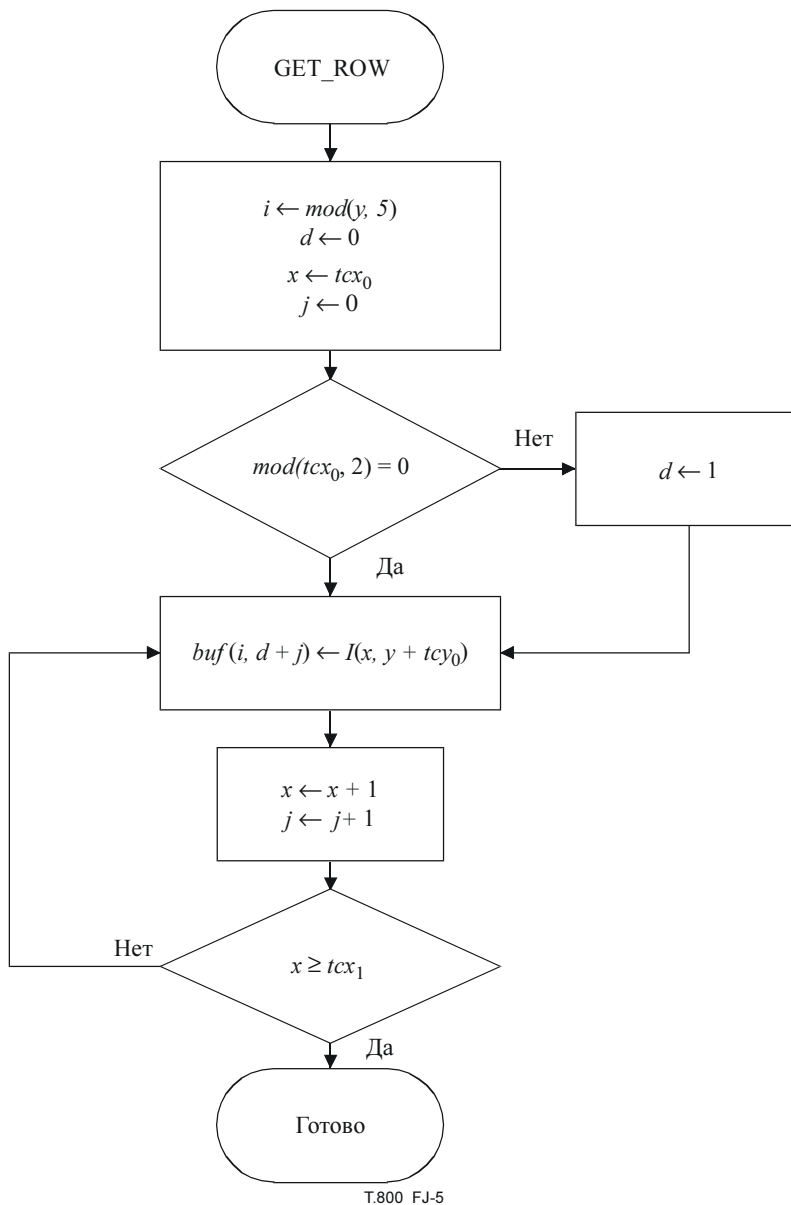


Рисунок J.5 – Процедура GET_ROW

J.5.2 Процедура INIT

Как показано на рисунке J.6, процедура INIT считывает пять рядов образцов компонента тайла изображения со смещенным уровнем и посылает эти ряды образцов в буфер buf .

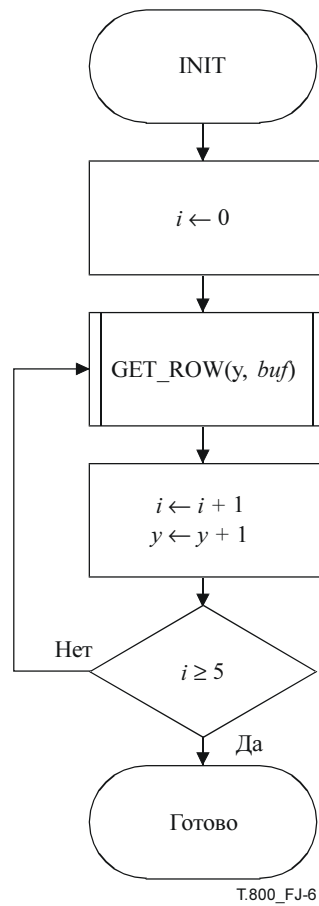


Рисунок J.6 – Процедура INIT

J.5.3 Процедура START_VERT

Как показано на рисунке J.7, процедура START_VERT модифицирует коэффициенты в буфере $buf(i, j)$. На этом рисунке, также как и на всех последующих в этом разделе, выражение $buf(i) \leftarrow buf(i) + \alpha \cdot buf(i_2)$ эквивалентно $buf(i, j) \leftarrow buf(i, j) + \alpha \cdot buf(i_2, j)$ для $d \leq j < tex_0 - tex_1 + d$.

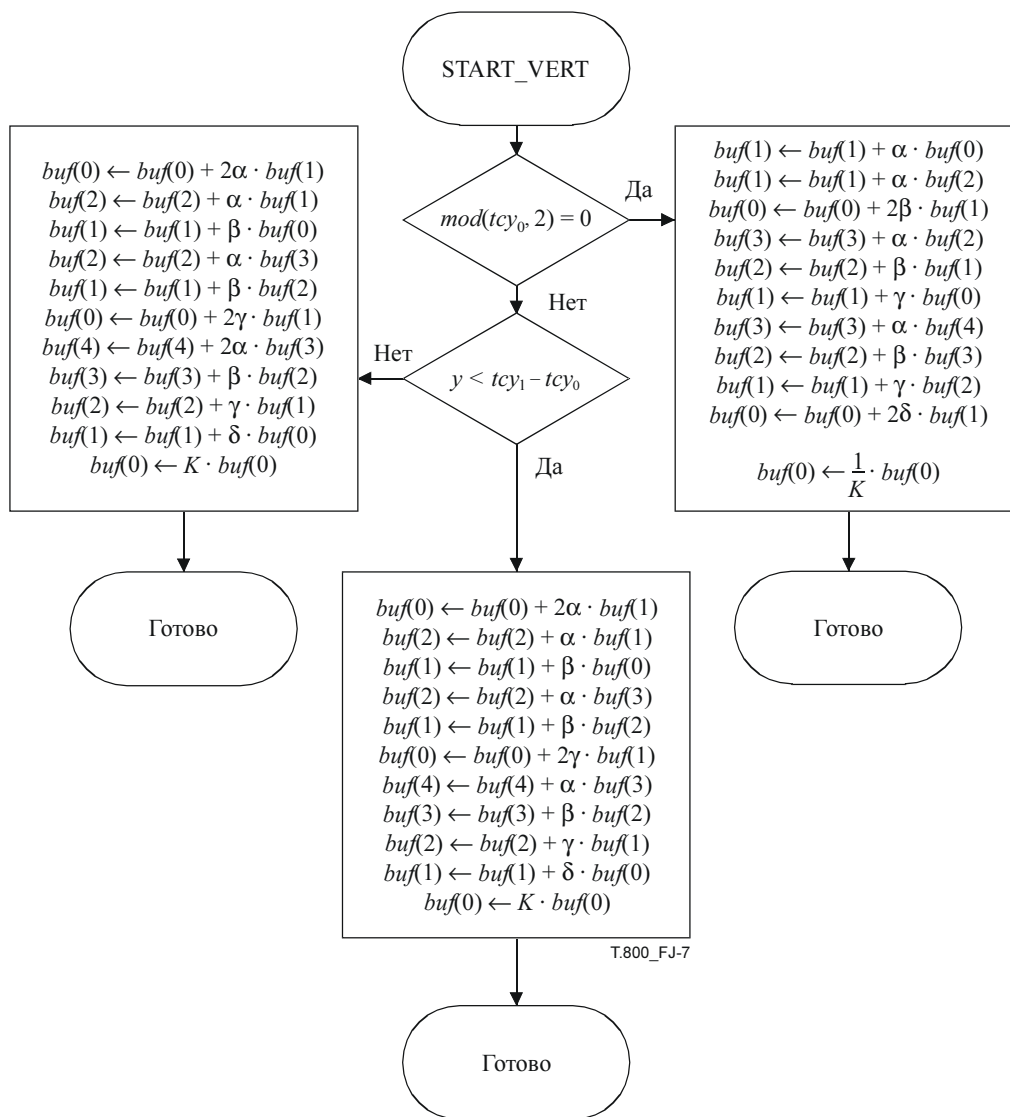


Рисунок J.7 – Процедура START_VERT

J.5.3.1 Процедура RB_VERT_1

Как показано на рисунке J.8, процедура RB_VERT_1 модифицирует коэффициенты в буфере $buf(i, j)$.

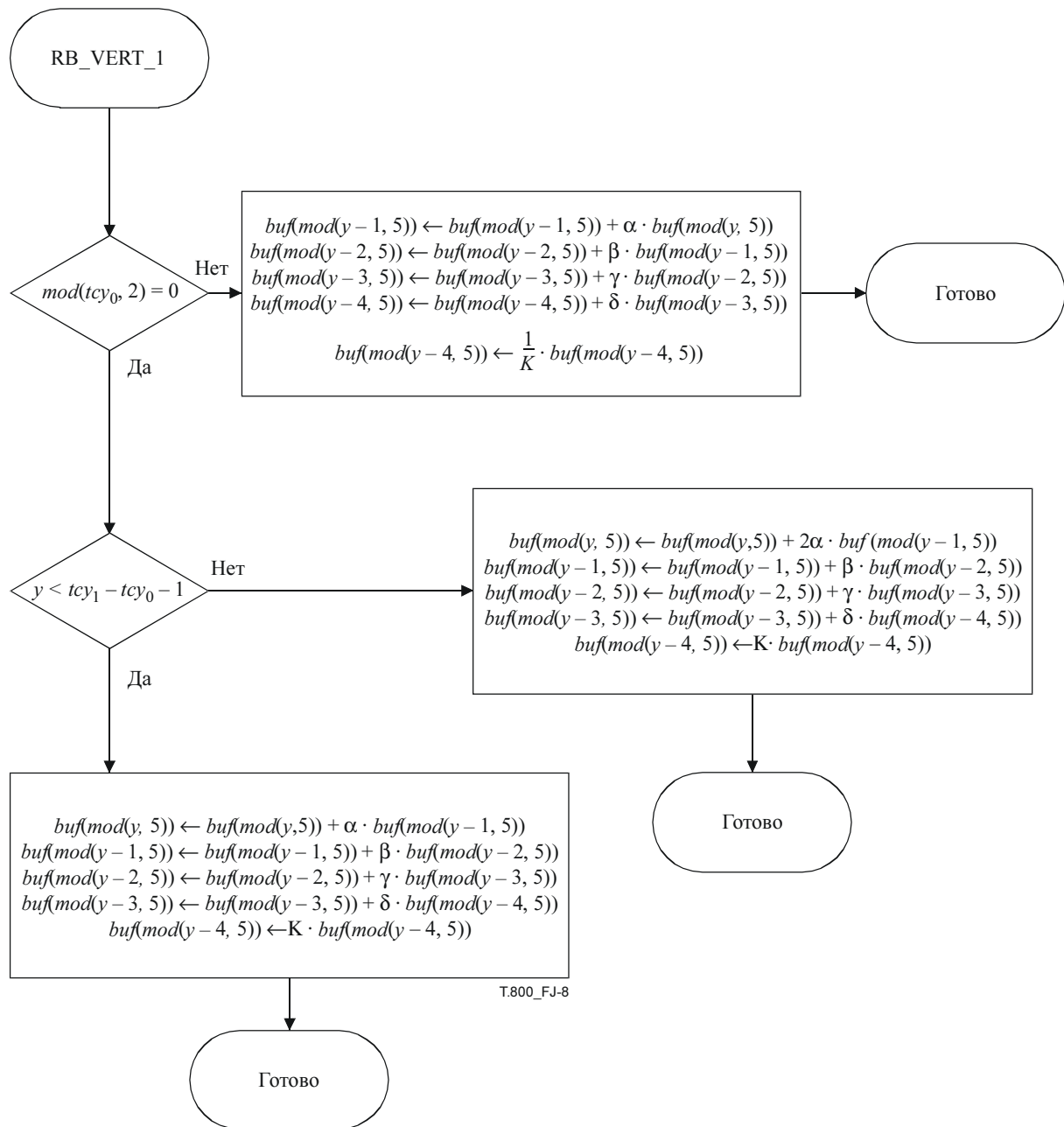


Рисунок J.8 – Процедура RB_VERT_1

J.5.3.2 Процедура RB_VERT_2

Как показано на рисунке J.9, процедура RB_VERT_2 модифицирует коэффициенты в буфере $buf(i, j)$.

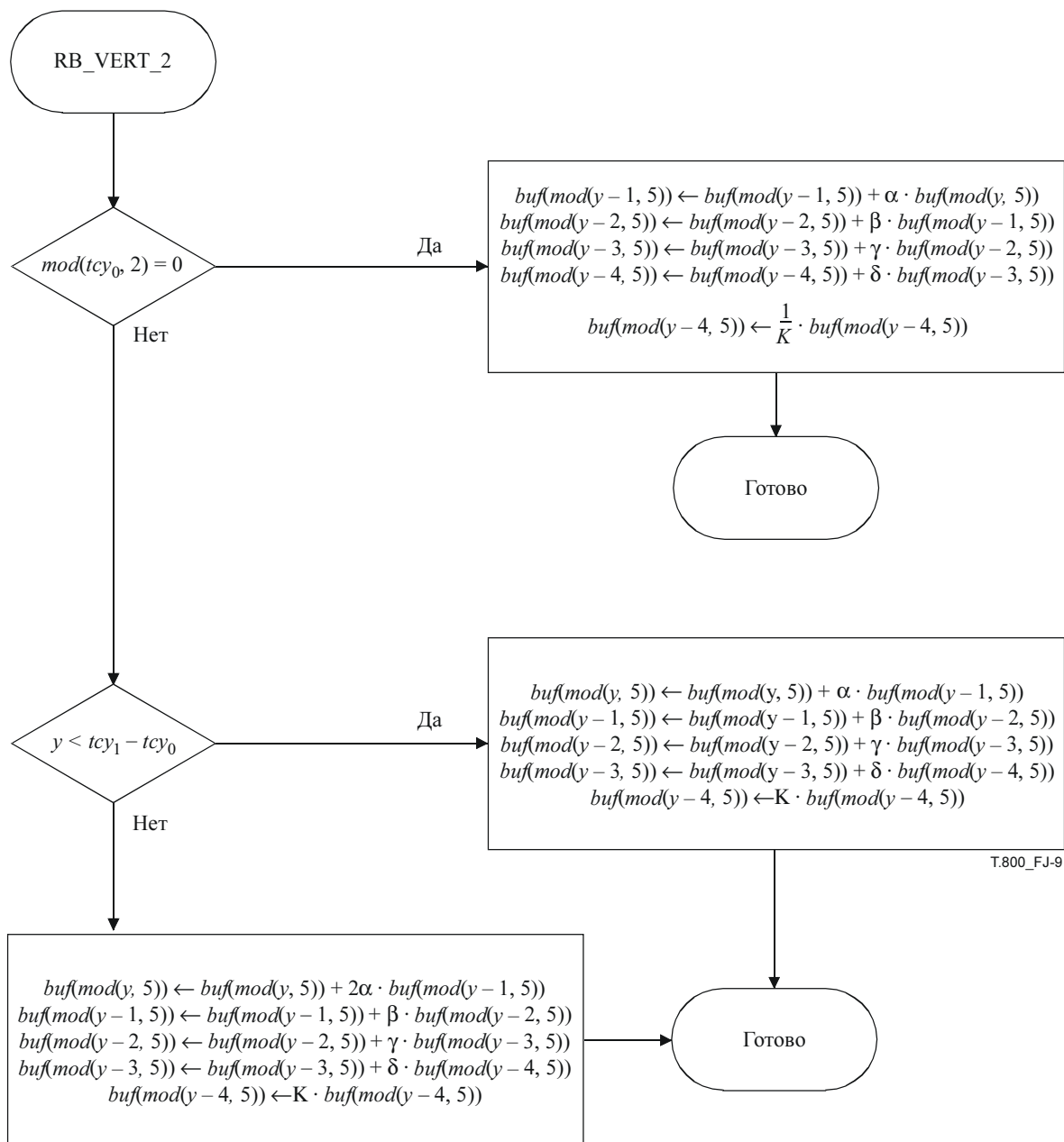
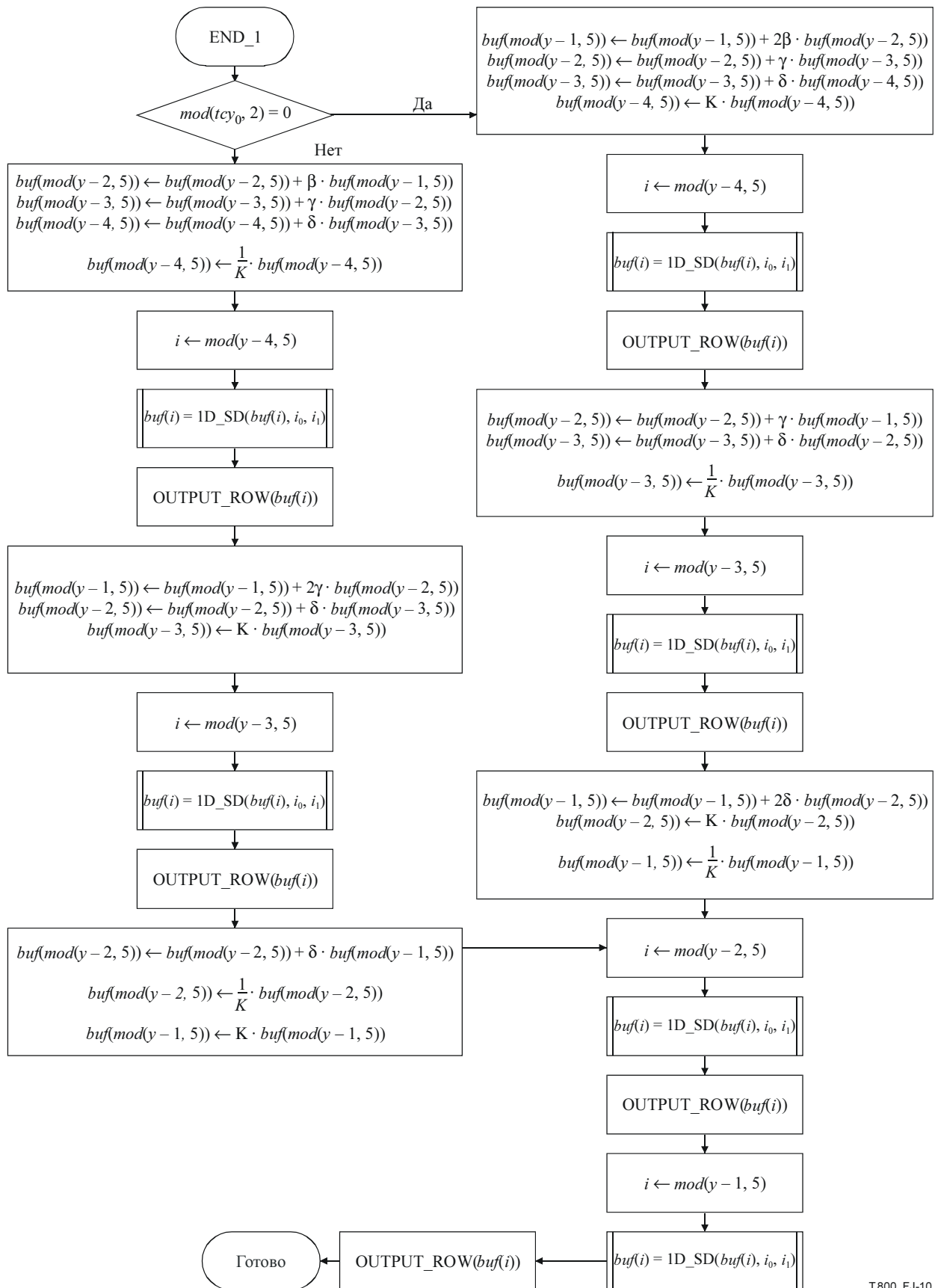


Рисунок J.9 – Процедура RB_VERT_2

J.5.3.3 Процедура END_1

Процедура END_1 подробно показана на рисунке J.10.



T.800_FJ-10

Рисунок J.10 – Процедура END_1

Ј.5.3.4 Процедура END_2

Процедура END_2 подробно показана на рисунке Ј.11

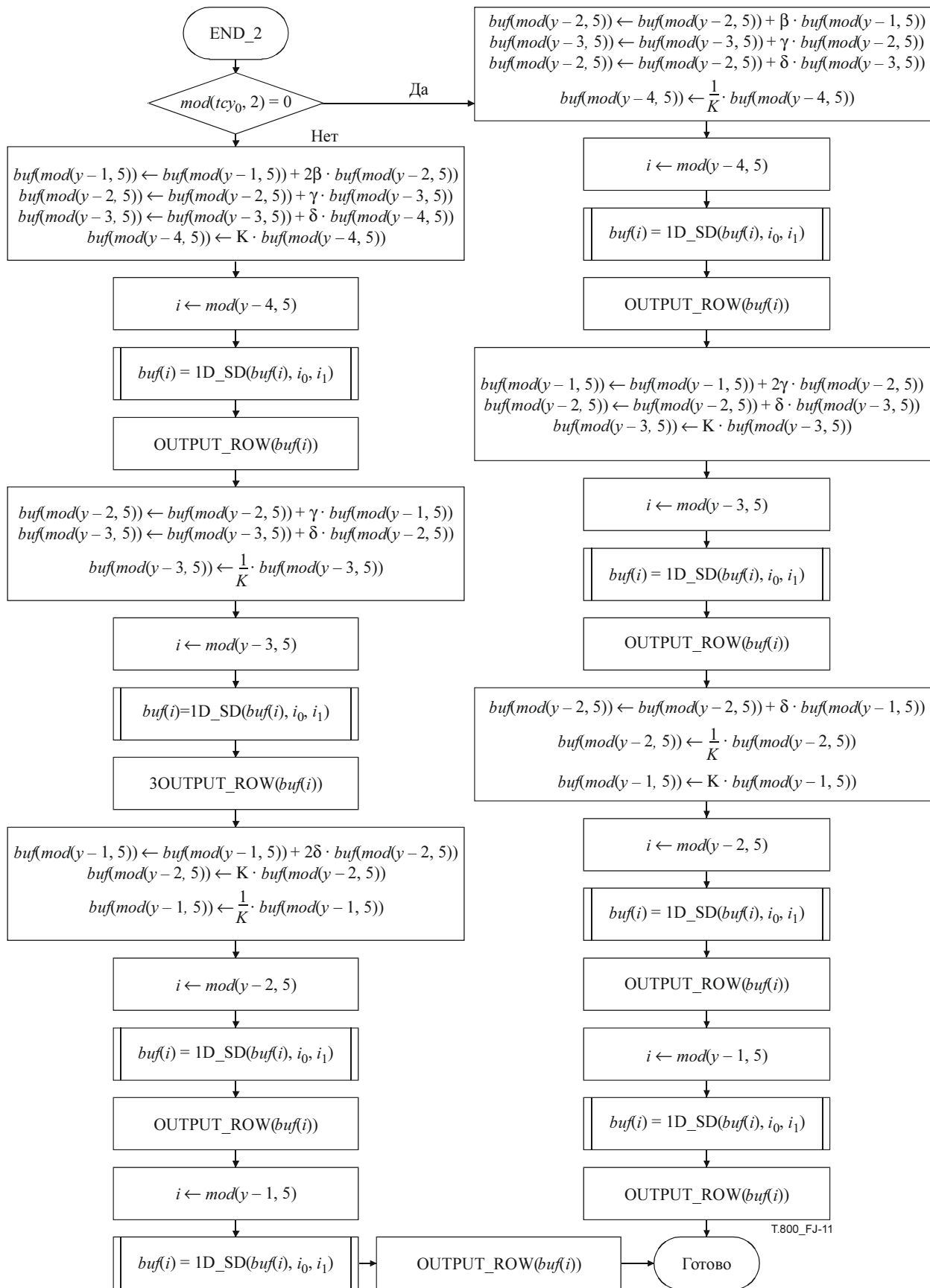


Рисунок Ј.11 – Процедура END_2

J.5.4 Процедура OUTPUT_ROW

Эта процедура возвращает ряд $buf(i)$ преобразованных коэффициентов, которые соответствуют либо поддиапазонам 1LL и 1HL, либо 1LH и 1HH. Этот ряд преобразованных коэффициентов можно либо сохранить во внешней памяти, либо сразу же обработать.

J.6 Кодирование, основанное на сканировании

В некоторых приложениях используют сенсоры сканирования, которые ряд за рядом создают изображения (возможно, и неограниченной длины), но имеют ограниченный размер памяти для обработки. Эти приложения требуют полного кодирования на основе сканирования, при котором в данный момент времени требуется удерживать только ограниченное количество байтов без существенного ухудшения качества. Примеры использования таких систем кодирования на основе сканирования приведены в [39], [40]. Ниже описаны рекомендуемые процедуры.

Пример управления скоростью, приведенный в J.14, требует буферизации всего сжатого кодированного потока со скоростью выше, чем заключительная скорость чередования. С другой стороны, подход, который основан на сканировании, можно использовать там, где частотное преобразование "по рядам" (см. J.5) следует за распределением скорости и процедурой кодирования, которые основаны на сканировании. Это позволит обеспечить передачу сжатых коэффициентов преобразования вскоре после того, как они были получены. С этой целью после частотного преобразования вводят ограниченную буферную память (буфер сканирования). Элемент сканирования состоит из локализованного набора частотных коэффициентов. Это может быть тайл или участок, которые соответствуют небольшому числу рядов пространства изображения. Буфер сканирования может содержать более одного элемента сканирования.

Алгоритм управления скоростью применяют в буфере сканирования к сжатым данным, а первый элемент сканирования отправляют в поток битов. Если буфер сканирования имеет более одного элемента сканирования, включают механизм "скользящего окна" для управления скоростью. Такой подход может дать лучший результат в сжатии ценой незначительного увеличения требований к сложности и памяти.

Подход, основанный на сканировании, не влияет на процесс декодирования JPEG2000.

J.7 Защита от ошибок

В этом разделе описан метод декодирования изображений, которые были кодированы с использованием синтаксиса защиты от ошибок.

Для многих приложений требуется доставка данных изображения по каналам связи разного типа. Обычные каналы радиосвязи подвержены воздействию случайных и взрывных битовых ошибок. Связи по Интернет свойственны потери из-за перегрузки трафика. Для улучшения качества передачи сжатых изображений по каналам, подверженным ошибкам, в эту Рекомендацию | Международный стандарт включены синтаксис и инструментальные средства для защиты потока битов от ошибок.

Инструментальные средства защиты от ошибок (см. таблицу J.18) в данной Рекомендации | Международном стандарте, которые связаны с ошибками в канале, используют следующие подходы: ресинхронизация и разбиение на части сжатых данных, обнаружение и маскирование ошибок, а также приоритет передачи, основанный на качестве обслуживания (QoS). Инструментальные средства защиты от ошибок описаны для каждой категории.

Таблица J.18 – Инструментальные средства защиты от ошибок

Тип инструмента	Наименование	Ссылка
Уровень статистического кодирования	Блоки кодирования Завершение арифметического кодирования для каждого прохода Возврат контекста в исходное положение после каждого кодирующего прохода Избирательный обход арифметического кодирования Сегментация символов	Приложение D
Уровень пакетов	Короткий формат пакета Пакет с ресинхронизацией маркера (SOP)	Приложение B

Статистическое кодирование квантованных коэффициентов выполняют в блоках кодирования. Поскольку кодирование и декодирование блоков независимо, битовые ошибки в потоке битов блока будут сохраняться только в этом блоке кодирования (см. Приложение D).

Завершение арифметического кодирования допустимо после каждого кодирующего прохода. Кроме того, контекст может быть возвращен в исходное положение после каждого кодирующего прохода. Это позволяет арифметическому кодеру продолжать декодировать кодирующие проходы после появления ошибок (см. D.4).

Оптимальный стиль обхода арифметического кодирования вводит необработанные биты без арифметического кодирования в поток битов. Это предотвращает размножение ошибок, к которому чувствительна переменная длина кодирования (см. D.6).

ИСО/МЭК 15444-1:2002 (R)

Короткие пакеты создают перемещением заголовков пакетов в сегменты маркеров PPM или PPT (см. А.7.4 и А.7.5). Если имеются ошибки, заголовки пакетов в сегментах маркеров PPM или PPT могут быть все же объединены с исправным пакетом, используя номер последовательности в маркере SOP.

Символ сегментации – это специальный символ. Правильное декодирование этого символа подтверждает правильность декодирования и данной битовой матрицы, что позволяет обнаружить ошибку (см. D.5).

Пакет с ресинхронизацией маркера (см. А.8.1) допускает пространственное разделение и ресинхронизацию. Этот маркер помещают перед каждым пакетом в тайле с номером последовательности, который начинается с нуля и возрастает с каждым пакетом. Порядок пакетов описан в В.10.

J.8 Требование совместимости с файлами JFIF/SPIFF

Этот раздел относится только к расширениям к спецификации Рекомендации МСЭ-Т Т.81 | ИСО/МЭК 10918-1 и никаким образом не затрагивает эту Рекомендацию | Международный стандарт, а также любое ее расширение. Не существует требований поддержки этих характеристик в декодере JPEG 2000.

J.8.1 Методология совместимости

Чтобы избежать любой модификации файлового формата, описанного в нормативной части этой Рекомендации | Международном стандарте, в Рекомендации МСЭ-Т Т.81 | ИСО/МЭК 10918-1 определены новые характеристики, соответствующие расширенным методологиям, описанным в Рекомендации МСЭ-Т Т.86 | ИСО/МЭК 10918-4. Эти характеристики ускорят переход от использованных в Рекомендации МСЭ-Т Т.81 | ИСО/МЭК 10918-1 к характеристикам данной Рекомендации | Международного стандарта.

Эти характеристики доступны через управление регистрации, установленной для целей расширений и определенной в спецификациях Рекомендации МСЭ-Т Т.84 | ИСО/МЭК 10918-3 и Рекомендации МСЭ-Т Т.86 | ИСО/МЭК 10918-4. Следовательно, они появляются на сайте JURA (JPEG Utilities Registration Authority): <http://jura.jpeg.org>. Статус этих характеристик такой же, как и всех зарегистрированных параметров. Они являются расширением Рекомендации МСЭ-Т Т.81 | ИСО/МЭК 10918-1.

J.8.2 Проектирование параметров совместимости

- 1) Прозрачность для декодера JFIF (совместимо с Рекомендацией МСЭ-Т Т.81 | ИСО/МЭК 10918-1 и Рекомендацией МСЭ-Т Т.83 | ИСО/МЭК 10918-2).
- 2) Прозрачность для декодера JFIF (совместимо с Рекомендацией МСЭ-Т Т.84 | ИСО/МЭК 10918-3).
- 3) Структура основного блока JP2/JPX.
- 4) Нет необходимости в транскодировании DCT в частотный кодированный поток.
- 5) Сохранять целостность и доступ к любой информации, связанной с IPR.
- 6) Использовать расширения ".jpg" или ".spfi", соответствующие происхождению исходного файла.
- 7) Давать преимуществ любому зарегистрированному параметру, как это описано в Рекомендации МСЭ-Т Т.86 | ИСО/МЭК 10918-4.

J.9 Использование ограниченного метода консорциума ICC за пределами полной машины управления цветом ICC

В этом Приложении описан ограниченный метод ICC для спецификации пространства цвета файла JP2 с использованием характеристик ICC, основанных на версии ICC.1:1998-09 спецификации формата характеристик [41]. Это Приложение рассчитано специально на разработчиков, которые не используют полностью машину управления цветом ICC и, следовательно, должны извлекать параметры преобразования из характеристик ICC и обрабатывать изображение, используя специальный код приложения.

J.9.1 Выделение преобразования цвета из характеристик консорциума ICC

J.9.1.1 Формат характеристик ICC

Характеристики ICC используют формат данных тегов для организации информации. Это описано в разделе 6 спецификации формата характеристик ICC. Формат состоит из 128-байтового заголовка, таблицы тегов и данных тегов. Каждый тег идентифицирован 32-битовой сигнатурой, которая обычно соответствует четырем знакам ASCII. Данные каждого тега хранят в формате, который описывает различные элементы данных. Каждый формат идентифицирован сигнатурой типа данных, которая составляет первые 32 бита данных тега. Чтобы получить данные тега, следует сначала найти в таблице тегов место для сигнатуры данного тега, а затем отыскать данные, основанные на местоположении и размере в данных тега. Как только эта информация найдена, сигнатура типа данных тега описывает, как следует интерпретировать данные тега.

Существенные теги, используемые при обработке изображения с ограниченными характеристиками ICC, приведены в таблице J.19.

Таблица J.19 – Теги обработки, которые используют ограниченные характеристики ICC

Наименование тега	Сигнатура тега	Тип данных тега	Сигнатура типа данных тега
redTRCTag	'rTRC'	curveType	"curv"
greenTRCTag	'gTRC'	curveType	"curv"
blueTRCTag	'bTRC'	curveType	"curv"
redColorantTag	'rXYZ'	XYZType	"XYZ\040"
greenColorantTag	'gXYZ'	XYZType	"XYZ\040"
blueColorantTag	'bXYZ'	XYZType	"XYZ\040"
grayTRCTag	'kTRC'	curveType	"curv"

Отметим, что характеристики ICC и, следовательно, ограниченные характеристики ICC, могут содержать другие теги, как например, *mediaWhitePoint*. Хотя эти теги не используют в канале обработки по умолчанию ограниченных характеристик ICC, как описано в J.9.2, в более сложных сценариях это может дать преимущество в получении такой информации, которая позволит получить более точное и оптимизированное изображение.

Формат характеристик ICC и, следовательно, ограниченных характеристик ICC, определяет модель обработки, производит преобразования между значениями кода устройства и пространственными характеристиками соединения (PCS). Эта модель состоит из двух частей, набора из трех одномерных таблиц интерполяции и матрицы 3×3 . Таблицы интерполяции формируют из тегов redTRCTag, greenTRCTag и blueTRCTag. Матрицу формируют из тегов redColorantTag, greenColorantTag и blueColorantTag. Основная модель обработки использует эти элементы, определенные равенством (4), раздела 6.3.1.2 спецификации формата характеристик ICC и описанные в J.9.2. Отметим, что эти характеристики используются для преобразования значений устройств в PCS. Матрицы и таблицы интерполяции следует инвертировать для преобразования из пространства PCS в значения устройств.

Полная спецификация формата характеристик (и, следовательно, ограниченных характеристик ICC) содержится в спецификации формата характеристик ICC, версии ICC.1:1998-09.

J.9.1.2 Таблицы интерполяции

Таблицы интерполяции используют формат curveType. Тег типа curveType состоит из счета, за которым следует номер в виде таблицы входов 16-битовых значений без знака.

Если значение счета 1, единичный вход таблицы – это не кодированное значение гамма (степени контрастности). В этом случае одномерную таблицу формируют по формуле:

$$linear = \left(\frac{dVal}{dMax} \right)^{256} \times 65535, \quad (J-2)$$

где *dVal* – значение компонента устройства с максимальной величиной *dMax*, а *tVal* – значение входа таблицы.

Если значение счета больше единицы, входы – это значения из таблицы интерполяции. Первый вход соответствует значению устройства 0, а последний – максимальному значению устройства (например, 255 для 8-битовых данных и 65535 для 16-ти битовых данных). Оставшиеся входы равномерно распределены между этими двумя. Например, если имеется 8-битовые данные и 6 входов, то четвертый вход соответствует значению $255 * ((4 - 1)/(6 - 1)) = 153$. Заметим, что 1 вычитают из позиции входа в таблице, чтобы преобразовать эту величину в отсчитываемый от нуля индекс, а 1 вычитают из числа входов, чтобы преобразовать эту величину в число интервалов между входами.

Чтобы преобразовать таблицу интерполяции в справочную таблицу, число входов в интерполяции должно точно соответствовать числу возможных значений устройств. В случае 8-битовых данных – это 256 возможных величин. Если в таблице интерполяции 6 входов, это число следует увеличить до 256 входов. Существует несколько методов выполнения этой операции. Часто используют линейную интерполяцию. При этом входы, представленные таблицей интерполяции, вычисляют линейной интерполяцией между соседними входами таблицы интерполяции, используя значения устройств в качестве "интерполянтов". Например, рассмотрим таблицу интерполяции с 6-ью входами со значением T3 на третьем входе и T4 – на четвертом. Индекс 128 справочной таблицы интерполяции получит позицию таблицы интерполяции $(128/255) * 5 = 2,5098$. значение справочной таблицы с индексом 128 вычислено интерполяцией между значениями входа 3 и входа 4, используя "интерполянт" 0,5098: значение с индексом $128 = T3 + ((T4 - T3) * 0,5098)$.

Каждый вход в таблице делением на 65535 можно преобразовать в значение с плавающей точкой.

При обработке пикселей каждый компонент пикселя сверяют с соответствующей ему справочной таблицей. Величины с плавающей запятой получают из вектора 3-элементного столбца, приведенного в равенстве (4) раздела 6.3.1.2 спецификации формата характеристик ICC.

Ј.9.1.3 Матрица

Матрицу формируют из значений тегов *redTRCTag*, *greenTRCTag* и *blueTRCTag*. Эти теги используют тип *XYZType*, который содержит три *XYZNumberTypes*. Первым является компонент X, вторым – Y, а третьим – Z. Каждый тип *XYZNumberType* содержит 32-битовые целые числа со знаком, которые можно преобразовать делением на 65 536 в числа с плавающей запятой. Значения XYZ каждого тега соответствуют ряду матрицы, как показано в равенстве (4) раздела 6.3.1.2 спецификации формата характеристик ICC. Эту матрицу умножают на вектор полученного в таблице столбца, чтобы получить значения XYZ PCS.

Ј.9.1.4 Объединенные характеристики источника и пункта назначения

Характеристики в файле JP2 описывают, как преобразовать данные изображения в пространство PCS. Это относят к характеристикам "источника". Обычно данные изображения требуется преобразовать в данные другого устройства, например, дисплея. Это устройство относят к устройству "пункта назначения", а характеристики называют характеристиками пункта назначения. Преобразование выполняют комбинацией моделей обработки источника и пункта назначения. Полагая, что преобразование устройства назначения ограничено матрицей 3×3, за которой следует ID, процедуру можно представить следующим образом:

- 1) Получить таблицы интерполяции и матрицу от характеристик источника.
- 2) Получить таблицу интерполяции и матрицу от характеристик пункта назначения.
- 3) Преобразовать таблицы интерполяции и матрицу характеристик пункта назначения.
- 4) Используя матричное умножение, объединить эти две матрицы.

Так получают общую модель обработки, которая представляет набор одномерных таблиц, матрицы и второй набор одномерных таблиц. Это можно использовать для преобразования пикселей источника изображения в пиксели изображений пункта назначения.

Ј.9.2 Уравнения обработки цвета для трехкомпонентных изображений RGB

Цель метода ограниченных характеристик ICC заключается в ограничении набора всех характеристик ICC до такого набора, который можно описать простым набором уравнений обработки. Спецификации ICC определяют этот класс характеристик как входные характеристики трех цветов на основе матриц (определены в разделе 6.3.1.2 спецификации формата характеристик ICC) и входные характеристики черно-белого цвета (определены в разделе 6.3.1.1 спецификации формата характеристик ICC). Класс характеристик трех цветов на основе матриц можно описать следующими уравнениями:

$$\begin{aligned} linear_r &= redTRC [decompressed_r] \\ linear_g &= greenTRC [decompressed_g] \\ linear_b &= blueTRC [decompressed_b] \end{aligned} \tag{J-3}$$

$$\begin{bmatrix} connection_x \\ connection_y \\ connection_z \end{bmatrix} = \begin{bmatrix} redColorant_x & greenColorant_x & blueColorant_x \\ redColorant_y & greenColorant_y & blueColorant_y \\ redColorant_z & greenColorant_z & blueColorant_z \end{bmatrix} \begin{bmatrix} linear_r \\ linear_y \\ linear_z \end{bmatrix}, \tag{J-4}$$

где *decompressed_{rgb}* – это пиксели декомпрессированного оригинала, а *connection_{xyz}* – пиксели, преобразованные в форму XYZ пространственных характеристик соединения (*XYZ_{PCS}*). В равенстве (J-3) три справочные таблицы загружают из ограниченных характеристик ICC, соответственно, из тегов *redTRCTag*, *greenTRCTag* и *blueTRCTag*, как это описано в разделах 6.4.38, 6.4.18 и 6.4.4 спецификации формата характеристик ICC. Формат общих данных этих тегов определен в разделе 6.5.25 спецификации характеристик. В равенстве (J-4) ряды матриц загружают соответственно из тегов *redColorantTag*, *greenColorantTag* и *blueColorantTag*, как это определено в разделах 6.4.39, 6.4.19 и 6.4.5 спецификации формата характеристик ICC. Формат общих данных этих тегов определен в разделе 6.5.2 спецификации характеристик.

Класс входных характеристик черно-белого цвета можно описать следующим равенством:

$$connection = grayTRC [устройство], \tag{J-5}$$

где устройство (device) – это пиксели декомпрессированного оригинала, а соединение (connection) – это канал серого цвета характеристик пространства соединения. В равенстве (J-5) справочную таблицу загружают из ограниченных характеристик ICC, из тега grayTRCTag, как указано в разделе 6.3.17. Формат данных этого тега определен в разделе 6.5.2 спецификации характеристик.

J.9.3 Преобразование изображений в sRGB

Один из наиболее общих сценариев приложения приводит к ситуации, когда описываемое изображение, в котором использован метод ограниченных характеристик ICC, должен быть преобразован в пространство цвета sRGB для вывода на программируемый дисплей (например, настольной издательской системы и браузеров сети) [42].

Это преобразование используют вместе с методом ограниченных характеристик ICC для получения результирующих значений sRGB из значений цвета источника оригинала [47]. Там, где это возможно, такие преобразования следует объединить (справочные таблицы 1D или матрицы), чтобы улучшить характеристики обработки. Для этого случая будет показано только преобразование из пространства характеристик соединения (XYZ_{PCS}). Далее его можно объединить с преобразованиями по равенствам (J-3) и (J-4).

Для завершения преобразования и передачи цветов, кодированных в системе XYZ_{PCS}, в цвета, кодированные в системе sRGB, необходимы три составляющие. Эти составляющие, представленные двумя матрицами 3 × 3 на канал и уравнением преобразования линейной части в нелинейную, на практике можно применить с помощью трех одномерных справочных таблиц.

Первая матрица в преобразовании требуется для выполнения преобразования цветовой адаптации между пространственными характеристиками соединения ICC (цвета из CIE D50) и определенной точкой адаптации белого sRGB (цвета из CIE D65). Можно использовать несколько разных преобразований. Первый пример – это преобразование цветовой адаптации по Брэдфорду (BFD) [43]. Преобразование Брэдфорда было продемонстрировано для получения точных результатов и было одобрено как часть рекомендованной CIE модели возникновения цвета (CIECAM97) [44], [45]. Преобразование BFD обычно включает линейную и нелинейную части. В данном примере преобразования нелинейная часть преобразования Брэдфорда была опущена, чтобы показать простую обработку матрицы 3 × 3. Можно показать, что даже при таком упрощении преобразование Брэдфорда дает хороший результат [46].

Вторая матрица в преобразовании – это основное преобразование матрицы, которое требуется для перемещения цветов из исходного пространства XYZ_{PCS} в исходный набор по Рекомендации МСЭ-Р ВТ.709-2, как это определяет стандарт sRGB, IEC/TC100/PT61966-2.1.

Разделенное преобразование выглядит так, как это следует из исходного преобразования, отмеченного знаком PT, и матрицей цветовой адаптации Брэдфорда, отмеченной знаком BFD:

$$\begin{bmatrix} slinear_r \\ slinear_g \\ slinear_b \end{bmatrix} = \begin{bmatrix} 3,2406_{PT} & -1,5372_{PT} & -0,4986_{PT} \\ -0,9689_{PT} & 1,8758_{PT} & 0,0415_{PT} \\ 0,0557_{PT} & -0,2040_{PT} & 1,0570_{PT} \end{bmatrix} \begin{bmatrix} 0,9554_{BDF} & -0,0231_{BDF} & 0,0633_{BDF} \\ -0,0284_{BDF} & 1,0100_{BDF} & 0,0211_{BDF} \\ 0,0123_{BDF} & -0,0205_{BDF} & 1,3305_{BDF} \end{bmatrix} \begin{bmatrix} connection_x \\ connection_y \\ connection_z \end{bmatrix}. \quad (J-6)$$

Однако эти матрицы можно объединить для формирования единой матрицы, как показано в следующем равенстве:

$$\begin{bmatrix} slinear_r \\ slinear_g \\ slinear_b \end{bmatrix} = \begin{bmatrix} 3,1337 & -1,6173 & -0,4907 \\ -0,9785 & 1,9162 & 0,0334 \\ 0,0720 & -0,2290 & 1,4056 \end{bmatrix} \begin{bmatrix} connection_x \\ connection_y \\ connection_z \end{bmatrix}. \quad (J-7)$$

Далее необходимо преобразовать $slinear_{rgb}$ в нелинейное значение sRGB. Для каждого красного, зеленого и синего каналов значения $slinear_{rgb}$ (обозначенного как $slinear_X$), соответствующий канал нелинейного значения sRGB (обозначенного как $sRGB_X$) вычисляют по равенству (J-8).

$$sRGB_X = \begin{pmatrix} 12,92 \cdot slinear_X & slinear_X \leq 0,0031308 \\ 1,055 \cdot slinear_X^{(1,0 / 2,4)} - 0,055 & slinear_X > 0,0031308 \end{pmatrix}. \quad (J-8)$$

Отметим, что преобразование из декомпрессированных пикселей в sRGB можно оптимизировать, объединив следующим образом матрицу цветов, описанную равенством (J-4), с матрицей преобразования XYZ в sRGB, описанной равенством (J-7):

$$\begin{bmatrix} slinear_r \\ slinear_g \\ slinear_b \end{bmatrix} = \begin{bmatrix} 3,1337 & -1,6173 & -0,4907 \\ -0,9785 & 1,9162 & 0,0334 \\ 0,0720 & -0,2290 & 1,4056 \end{bmatrix} \begin{bmatrix} redColorant_x & greenColorant_x & blueColorant_x \\ redColorant_y & greenColorant_y & blueColorant_y \\ redColorant_z & greenColorant_z & blueColorant_z \end{bmatrix} \begin{bmatrix} linear_r \\ linear_g \\ linear_z \end{bmatrix}. \quad (J-9)$$

Такая оптимизация уменьшает обработку пространства цвета из декомпрессированных пикселей в sRGB при применении справочной таблицы 1D, единой матрицы 3×3 и другой справочной таблицы 1D.

Показанные выше преобразования для sRGB можно обобщить для использования в преобразовании во многие заданные пространства цвета, отличные от sRGB. Во многих случаях предпринятые шаги будут совпадать с теми, которые требовались для преобразований в sRGB. Однако в других случаях потребуются меньше шагов. Например, если точка адаптации белого цвета целевого объекта пространства цвета совпадает с такой же точкой пространства PCS XYZ, то устраняется необходимость преобразования цветовой адаптации. В некоторых случаях, возможно, потребуются дополнительные шаги для компенсации различных факторов, например, условий просмотра. Преобразования действующих условий просмотра выходят за рамки данного Приложения, но содержатся в других публикация [41], [42], [46], [48].

Следует заметить, что в зависимости от хранилища пространства цвета возможна потеря информации, которая связана с преобразованием в sRGB или другие ограниченные цветовые гаммы вне пространства цвета. Например, рассмотрим случай, в котором хранилище пространства цвета – это расширенное пространство цвета цветовых гамм. Преобразование в sRGB приведет к ограничению цветов, которые расположены вне гаммы цветов sRGB. Поскольку это необходимый шаг при выводе изображения на дисплей, преобразования цвета (например, смещение цвета) могут допустить использование таких ограниченных данных (смещением в гамму sRGB). Таким образом, часто предпочтительнее выполнить больше преобразований цвета над оригиналом хранимых данных, прежде чем преобразовывать их в пространство цвета sRGB. Аналогично, если затем изображение должно быть распечатано на выходном устройстве, которое может печатать ограниченные цвета, то предпочтительнее вернуться к изображению в хранилище пространстве цвета, чем распечатывать изображение в sRGB. В качестве другого примера рассмотрим случай, в котором изображение в хранилище пространства цвета представляет собой кодированное цветовое место действия с расширенным диапазоном. Преобразование в sRGB по необходимости включит шаг формирования изображения, в котором информация о свете и/или тени будет ограничена до динамического диапазона дисплея на выходе. (Шаг формирования изображения обычно используют, применяя функцию градационной шкалы в качестве части TRCs, как показано в равенстве (J-3).) Информацию о потере шага формирования изображения нельзя использовать позже для модификации изображения. Аналогично, преобразование в sRGB может привести к ошибкам квантования изображения, которые могут ограничить качество этого изображения. Возможно, в обоих этих примерах более желательно сохранить изображение в хранилище пространства цвета и там совершать над ним различные преобразования. С другой стороны, изображение можно преобразовать в кодированную промежуточную обширную гамму. Такое преобразование в sRGB можно выполнить только для целей предварительного просмотра или в качестве заключительного шага в цепи создания изображения.

J.9.4 Преобразование изображений в другие пространства цвета

В противоположность этому, может оказаться желательным преобразовывать изображения, помимо sRGB, в другие пространства цвета с целью их отображения на специальных выходных устройствах или для манипуляций в пространстве цвета приложения. Например, если желательно вывести изображение на устройство CRT с характеристиками, которые отличаются от базовых характеристик sRGB дисплея, то матрица и нелинейности, описанные равенствами от (J-6) до (J-9), можно заменить соответствующими матрицей и нелинейностью конкретного устройства CRT. В общем, могут быть использованы: матрица, связанная с люминофорными цветами и точкой белого цвета для преобразования трехцветных значений PCS в соответствующие линейные значения RGB, а нелинейности – для связи линейных значений RGB с соответствующими кодовыми значениями. Аналогично, другие дополнительные пространства цвета RGB, например ROMM RGB, могут быть также вычислены подстановкой соответствующей матрицы и нелинейности [49].

В некоторых случаях желательно преобразовать изображения в другие пространства цвета, которые не могут быть описаны простым преобразованием матрицы/нелинейности. Такую операцию можно выполнить заменой преобразования, описанного в J.9.3, соответствующим преобразованием в желаемое пространство цвета. Во многих случаях это удобно выполнить, используя для желаемого пространства цвета характеристики ICC.

J.9.5 Диапазоны входа и выхода и квантование

Входные кодовые значения в справочных таблицах равенства (J-3) (redTRC, greenTRC и blueTRC) должны быть также объединены по одинаковым точностям, как и декомпрессированные кодовые значения, и индексированы так, чтобы $TRC[i]$ создавал верные значения линейной интенсивности для входного кодового значения i . Входные кодовые значения, которые превышают число элементов справочной таблицы минус 1, должны быть ограничены числом элементов справочной таблицы минус 1.

Выходные пиксели из равенства (J-3) должны быть действительными значениями линейной интенсивности, номинально в диапазоне (0,0, 1,0).

Вход в матрице цвета в равенстве (J-4) также должен иметь действительные значения линейной интенсивности в диапазоне (0,0; 1,0). Выход этого равенства (значения XYZ_{PCS}) следует масштабировать таким образом, чтобы значение Y находилось в диапазоне (0,0; 1,0). Нейтральные значения в изображении следует отображать значениями XYZ, которые имеют цветность точки белого в PCS (используют при $X/Y = 0,9642$, и $Z/Y = 0,8250$). Если приложение преобразует во входные кодовые значения пространства цвета sRGB, то диапазон выхода допускает непосредственное объединение матриц, как указано в равенстве (J-8).

Диапазоны и квантование из пространства XYZ_{PCS} в пространство sRGB подобны. Входы и выходы в равенстве (J-6) и то же в равенстве (J-8) – это также действительные величины в диапазоне (0,0; 1,0).

Выходы в равенстве (J-8) – это значения в диапазоне (0,0; 1,0). Однако в общем эти значения следует масштабировать до 255, чтобы получить 8-битовые значения в пространстве sRGB. Это крайне чувствительная зависимость, в частности, от того, была ли сделана дополнительная обработка (если такая произведена). Однако строго

рекомендуется, чтобы любая обработка цвета выполнялась над данными источника изображения (decompressed, decompressed_g, decompressed_b), прежде чем выполнялось преобразование в пространство sRGB, поскольку существует большая вероятность ухудшения квантования.

J.9.6 Преимущества спецификации множества пространств цвета

Формат JP2 позволяет файлу определять методы множественности для интерпретации пространства цвета изображения. Например, некоторое приложение может записать изображения, в которых значения пикселей уже преобразованы в сигналы, необходимые для управления конкретным выходным устройством. В этой ситуации в приложении полезно создать для этого устройства простой механизм определения, что устройству не требуется дополнительная обработка цвета. Это можно выполнить, определив имя устройства пространства цвета, используя метод нумерованных пространств цвета в одном из блоков спецификации цвета в этом файле.

Однако другие приложения, такие как браузеры сети, должны преобразовать изображение в сигналы, подходящие для отображения на других устройствах. Очень вероятно, что эти приложения не смогут распознать пространство цвета, установленное продавцом. Поэтому весьма полезно для автора оригинального файла записать в этом файле второй блок спецификации цвета, который использует метод ограниченных характеристик ICC или метод исходных характеристик ICC. Включение такого второго механизма значительно увеличит число приложений, которые будут в состоянии правильно интерпретировать пространство цвета изображения.

Отметим, что метод выбора из многих методов спецификации цвета, определенный в единственном файле, не определен данной Рекомендацией | Международным стандартом. Каждое приложение должно выбрать метод, который наилучшим образом отвечает требованиям конкретного приложения.

J.10 Пример интерполяции множества компонентов

Примером нетрадиционной интерполяции служит кодирование областей интересов (ROI) в комплекте набора данных SAR. Каждую область ROI можно рассматривать как набор из двух чипов изображения, представляющих реальную (I) и мнимую (Q) части данных. Ансамбль из чипов I и Q можно собрать в набор "множества компонентов", даже если отдельные чипы не соединены и имеют разные объемные размеры. Исключая преобразование пространства цвета, этот ансамбль чипов может далее стать предметом сжатия с потерями или без потерь. Такая процедура имеет два преимущества: все области ROI в таком наборе данных могут быть сжаты за единственный проход, и распределение битов может быть оптимизировано в ансамбле областей ROI быстрее, чем по очереди чип за чипом.

J.11 Пример декодирования с показом промежуточных шагов

Рассмотрим следующий сжатый поток битов, в котором сдвиги от начала файла приведены слева в восьмеричной форме, а значения в файле – в шестнадцатеричной форме.

```

00000 FF4F FF51 0029 0000 0000 0001 0000 0009
00020 0000 0000 0000 0000 0000 0001 0000 0009
00040 0000 0000 0000 0000 0001 0701 01FF 5C00
00060 0740 4048 4850 FF52 000C 0000 0001 0001
00100 0404 0001 FF90 000A 0000 0000 001E 0001
00120 FF93 C7d4 0C01 8F0D C875 5DC0 7C21 800F
00140 B176 FFD9
    
```

Различные части этого потока битов можно декодировать следующими способами.

J.11.1 Основной заголовок

Основной заголовок запускают с байта 0, как указано маркером SOC, и заканчивают байтом 0104 (восьмеричным), который известен по маркеру SOT.

```

00000 FF4F Маркер SOC
00002 FF51 Маркер SIZ
00004 0029 Длина маркера Lsiz SIZ
00006 0000 Rsiz
00010 0000 0001 Xsiz
00014 0000 0009 Ysiz
00020 0000 0000 XOsiz
00024 0000 0000 YOsiz
00030 0000 0001 XTsiz
00034 0000 0009 YTtiz
00040 0000 0000 XTOsiz
00044 0000 0000 YTOsiz
    
```

ИСО/МЭК 15444-1:2002 (R)

00050	0001	Csiz
00052	07	Ssiz
00053	01	XRsiz
00054	01	YRsiz

Таким образом, "изображение" – это один компонент с плотностью 8 битов | образец, 1 образец по горизонтали и 9 по вертикали. Все образцы относятся к одному тайлу.

00055	FF5C	Маркер QCD
00057	0007	Длина маркера Lqcd QCD
00061	40	Sqcd
00062	4048 4850	SPqcd

Имеются 2 защитные бита, никакого квантования не производят (иного, кроме возможного усечения), а расширения размеров шагов квантователя ϵ_b составляют {8,9,9, 10}.

00066	FF52	Маркер COD
00070	000C	Длина маркера Lcod COD
00072	00	Scod (PPx = PPy = 15, No SOP, No EPH)
00073	00	Порядок последовательности действий
00074	0001	Число слоев
00076	00	Множественное преобразование компонента
00077	01	Число уровней декомпозиции
00100	04	Величина сдвига показателя ширины блока кодирования
00101	04	Величина сдвига показателя высоты блока кодирования
00102	00	Тип кодирующих проходов блока кодирования
00103	01	Преобразование

Не используют никакие граничные области. Существует один уровень частотного преобразования. Порядок последовательности действий: разрешение, слой, уровень, компонент, позиция, но реально существует только один слой. Блоки кодирования состоят из образцов 64×64 (отметим, что размер составляет 2^6 , в то время как значение потока битов составляет 4). Отсутствует избирательный обход арифметического кодирования, при каждом проходе кодирования отсутствует перезагрузка контекстов вероятностей или завершения, нет причинных контекстов вертикальных полос, нет предсказуемого завершения и нет сегментации символов. Используют обратимый фильтр 5-3.

J.11.2 Заголовок части тайла

Первый и единственный заголовок части тайла начинается на восьмеричном байте 0104 с маркера SOT и заканчивается на восьмеричном байте 0120 с маркером SOD.

00104	FF90	Маркер SOT
00106	000A	Длина маркера Lsot SOT
00110	0000	Isot
00112	0000 001E	Psot
00116	00	TPsot
00117	01	TNsot

Номер тайла – 0. Длина части тайла – 30 байтов. Следовательно, следующая часть тайла, или конец закодированного потока: $0104 + 036 = 0142$. Это – часть тайла 0. Для этого тайла имеется только одна часть тайла.

00120	FF93	Маркер SOD
-------	------	------------

В этом заголовке части тайла не существовало никаких маркеров COD или QCD, а также маркеров комментариев. Следовательно, все параметры кодирования определяют из основного заголовка. Следующие 16 байтов представляют сжатые данные (30-ти байтовой длины и с 14 байтовыми сегментами маркеров).

Сжатые данные (заголовки и тела пакетов)

00122	C7D4 0C01 8F0D C875 5DC0 7C21 800F
00140	B176

Конец изображения

00142 FFD9

Маркер EOC

J.11.3 Заголовки пакетов

Так как изображение представляет собой 1×9 и имеется один уровень преобразования (а блоки кодирования, граничные области и тайлы слишком велики, чтобы оказывать влияние), должно быть 5 низкочастотных коэффициентов и 4 коэффициента горизонтально низкочастотных и вертикально высокочастотных. Сжатые данные начинаются с заголовка пакета, который декодируют, как показано в таблице J.20.

Таблица J.20 – Декодирование заголовка первого пакета

Байты кодированного потока	Использованные биты	Комментарии
0xC7	1	Ненулевая длина пакета
	1	Включен только один блок кодирования
	0001	3 нулевых битовых матрицы ^{a)}
0xD4	11 1101010	16 кодирующих проходов этого блока ^{b)}
	0	Длина <i>LBlock</i> остается 3
0x0C	0000110	6 байтов сжатых данных ^{c)}
	0	Неиспользуемый добавочный бит
<p>a) Максимальное число битовых матриц согласно равенству (E-2) составляет 9, из которых 3 – нулевые, поэтому первым будет декодирован бит в битовой матрице 6.</p> <p>b) Число битов, которое требуется считать, зависит от сжатых данных, см. табл. B.2.</p> <p>c) Число битов, которое требуется считать, равно <i>LBlock</i> плюс минимальный уровень от $2 \log$ числа проходов: $3 + \lfloor \log 16 \rfloor = 7$.</p>		

Декодирование заголовка первого пакета требует 3 байтов и означает, что 6 байтов арифметически кодированных сжатых данных используют только для блока кодирования в этом пакете. Следовательно, заголовок следующего пакета начнется со сдвига 0134. Декодирование этого заголовка пакета показано в таблице J.21.

Таблица J.21 – Декодирование заголовка второго пакета

Байты кодированного потока	Использованные биты	Комментарии
0xC0	1	Ненулевая длина пакета
	1	Включен только один блок кодирования
0x7C	000000 01	7 нулевых битовых матриц ^{a)}
0x21	111100 001	7 кодирующих проходов этого блока
	0	Длина <i>LBlock</i> остается 3
0x80	0001 1	3 байта сжатых данных ^{b)}
	0000000	Неиспользуемый добавочный бит
<p>a) Максимум битовых матриц составляет 10, из которых 7 – нулевые, поэтому первым будет декодирован бит в битовой матрице 3.</p> <p>b) Число битов, которое требуется считать, равно $3 + \lfloor \log 7 \rfloor = 5$.</p>		

J.11.4 Арифметически кодированные сжатые данные

Шесть байтов сжатых данных первого блока кодирования (из первого пакета) могут быть декодированы, как показано в таблице J.22. Первый пункт – это метка контекста из Приложения D (которая может полностью отличаться для разных применений). Второй пункт – это тип контекста. Наконец, бит, возвращенный из арифметического кодера. Эти биты используют для определения низкочастотных горизонтальных и низкочастотных вертикальных коэффициентов. В арифметический кодер поступают байты, которые начинаются со сдвига 0125.

0000125 01 8F0D C875 5D

Таблица J.22 – Арифметическое декодирование первого блока кодирования

СТХ	Тип контекста	Бит	Комментарий
17	C4(ZERO_RUN)	1	Ненулевой проход.
18	C5(UNIFORM)	1	Четвертый коэффициент – это первый ненулевой (нумерация от 1).
18	C5(UNIFORM)	1	
9	C2(SIGN)	1	Отрицательный.
3	C1 (NEW_SIGNIFICANT)	0	Пятый коэффициент – не значимый.
3	C1 (NEW_SIGNIFICANT)	1	Третий коэффициент – значимый (первый коэффициент в проходе значимости).
10	C2 (SIGN)	0	Отрицательный (бит XOR равен 1).
3	C1 (NEW_SIGNIFICANT)	1	Пятый значимый коэффициент в этом кодирующем проходе.
10	C2 (SIGN)	0	Отрицательный(бит XOR равен 1).
15	C3 (REFINE)	0	Следующий бит четвертого коэффициента – это 0.
0	C1 (NEW_SIGNIFICANT)	1	Первый коэффициент – значимый.
9	C2 (SIGN)	1	Отрицательный.
4	C1 (NEW_SIGNIFICANT)	1	Второй коэффициент – значимый.
10	C2 (SIGN)	0	Отрицательный.
15	C3 (REFINE)	1	Все коэффициенты получены в проходе подстройки. Декодированный бит – это следующий бит коэффициента в последовательности от первого до пятого.
15	C3 (REFINE)	0	
15	C3 (REFINE)	1	
16	C3 (REFINE)	0	
15	C3 (REFINE)	0	
16	C3 (REFINE)	0	Следующая битовая матрица.
16	C3 (REFINE)	1	
16	C3 (REFINE)	1	
16	C3 (REFINE)	0	
16	C3 (REFINE)	0	
16	C3 (REFINE)	1	Следующая битовая матрица.
16	C3 (REFINE)	1	
16	C3 (REFINE)	1	
16	C3 (REFINE)	0	
16	C3 (REFINE)	1	
16	C3 (REFINE)	0	Последняя битовая матрица.
16	C3 (REFINE)	0	
16	C3 (REFINE)	0	
16	C3 (REFINE)	0	
16	C3 (REFINE)	1	

Следовательно, декодированные коэффициенты:

-26, -22, -30, -32, -19

Сжатые данные для единственного блока кодирования во втором пакете, представляющем высокочастотный вертикальный и низкочастотный горизонтальный поддиапазон, начинаются с восьмеричного сдвига 0137.

0000137 0F B176

Процесс декодирования описан в таблице J.23.

Таблица J.23 – Арифметическое декодирование второго блока кодирования

CTX	Тип контекста	Бит	Комментарий
17	C4 (ZERO_RUN)	1	Ненулевой проход.
18	C5 (UNIFORM)	0	Второй коэффициент – это первый ненулевой.
18	C5 (UNIFORM)	1	
9	C2 (SIGN)	0	Положительный.
3	C1 (NEW_SIGNIFICANT)	0	3-й и 4-й коэффициенты – в проходе очистки.
0	C1 (NEW_SIGNIFICANT)	0	
3	C1 (NEW_SIGNIFICANT)	0	1-й и 3-й коэффициенты – в проходе значимости.
3	C1 (NEW_SIGNIFICANT)	0	
14	C3 (REFINE)	0	2-й коэффициент.
0	C1 (NEW_SIGNIFICANT)	0	4-й коэффициент – в проходе очистки.
3	C1 (NEW_SIGNIFICANT)	1	1-й коэффициент – в проходе значимости.
10	C2 (SIGN)	0	Положительный.
3	C1 (NEW_SIGNIFICANT)	1	3-й коэффициент – в проходе значимости.
10	C2 (SIGN)	0	Положительный.
3	C1 (NEW_SIGNIFICANT)	0	4-й коэффициент – в проходе значимости.
16	C3 (REFINE)	1	2-й коэффициент – в проходе подстройки.

Декодированные высокочастотные вертикальные и низкочастотные горизонтальные коэффициенты:

1, 5, 1, 0.

J.11.5 Частота и сдвиг уровня

После инверсии обратимого фильтра 5-3 и сдвига уровня образцы компонентов в десятичной форме приобретают следующий вид:

101, 103, 104, 105, 96, 97, 96, 102, 109

J.12 Визуальное частотное взвешивание

Визуальная (зрительная) система человека играет важную роль в восприятии качества сжатых изображений. Поэтому пользователям и проектировщикам систем желательно воспользоваться преимуществами современных знаний о визуальном восприятии и т. п., чтобы использовать модели изменения чувствительности визуальных систем к пространственным частотам в качестве меры функции контрастной чувствительности (CSF). Поскольку вес CSF определяют визуальной частотой коэффициента преобразования, при частотном преобразовании существует одна весовая функция CSF на поддиапазон. Расчет весов CSF возложен на кодировщик и зависит от особенностей условий, при которых собираются просматривать декодированное изображение. Для подробностей расчета весов функции CSF просим обратиться к литературе [34] и [35].

Во многих случаях выбирают и используют только один набор весов CSF в зависимости от условий просмотра. Такое применение визуального частотного взвешивания считают фиксированным визуальным взвешиванием. В случае участия в этом процессе кодеров, учитывают, что кодированный поток битов может быть усечен, а условия просмотра на разных стадиях процесса могут сильно отличаться. При низких битовых скоростях качество сжатого изображения плохое, а подробности изображения не доступны. Обычно изображение просматривают на довольно большом расстоянии, и зрители больше заинтересованы в общем виде. Как только поступает все больше и больше битов, качество изображения улучшается, а его детали раскрываются. Оценивают изображение обычно с близкого расстояния или для лучшей оценки даже с увеличением, что эквивалентно уменьшению расстояния. Таким образом, на разных стадиях процесса используют разные веса функции CSF. Применение визуального частотного взвешивания с подстройкой считают последовательным визуальным кодированием. Ясно, что фиксированное визуальное взвешивание можно рассматривать как частный случай последовательного визуального кодирования.

При фиксированном визуальном взвешивании набор весов функции CSF, $\{w_i\}$, выбирают в соответствии с конечными условиями просмотра, где w_i – это вес i -ого поддиапазона. Набор весов CSF можно встроить двумя следующими способами.

J.12.1 Модификация размера шага квантования

Размер шага квантования кодировщика, q_i , коэффициентов преобразования i -ого поддиапазона настраивают обратно пропорционально весу функции CSF, w_i . Чем меньше вес CSF, тем больше размер шага квантования. Нормализованные по функции CSF индексы квантования обрабатывают обычным образом по процессу оптимизации R-D, который не следует модифицировать, чтобы учесть изменение размера шага квантования. Нет необходимости передавать декодеру веса функции CSF. Эта информация включена в размеры шагов квантования, которые в явной

форме передают для каждого диапазона. Такой подход нуждается в четком определении квантователя. Поэтому он особенно подходит для встроенного кодирования различными методами: от кодирования с потерями до кодирования без потерь.

J.12.2 Модификация порядка встроенного кодирования

Размеры шагов квантования не модифицируют, но вместо этого изменяют отклонения весов, вводимых в процесс оптимизации R-D, что позволяет эффективно управлять относительной значимостью включения различного числа битовых матриц из потока битов каждого кодированного блока. Нет необходимости передавать таблицу частотного взвешивания в явной форме. Такой подход рекомендуют, поскольку он дает результаты, аналогичные изложенным в J.12.1, и совместим со сжатием без потерь. Этот подход действует только на компрессор и совместим со всеми стратегиями квантования, включая неявное квантование.

J.12.3 Последовательное визуальное кодирование (VIP)

Если во время встроенного процесса кодирования требуется изменить веса визуальных частот, то очень неудобно менять величины коэффициентов или размеры шагов квантования. Более того, характеристики последующего статистического кодера могут деградировать из-за изменения статистики бинарного представления. Элегантный способ введения последовательного визуального кодирования (VIP) состоит в изменении "на лету" порядка, в котором блок кодирования частей битовых матриц будет появляться в общем встроенном потоке битов, основываясь на визуальных весах, вместо изменения величин коэффициентов или размеров шагов квантования. Другими словами, визуальный вес воздействует скорее на порядок кодирования, чем на содержание кодирования.

Последовательность наборов визуального взвешивания для различных диапазонов битовых скоростей выражают следующим образом:

$$\begin{aligned} \text{Набор взвешивания } 0: r(0), c W(0) &= \{w_0(0), w_1(0), \dots, w_n(0)\}; \\ \text{Набор взвешивания } 1: r(1), c W(1) &= \{w_0(1), w_1(1), \dots, w_n(1)\}; \\ &\dots \\ \text{Набор взвешивания } m: r(m), c W(m) &= \{w_0(m), w_1(m), \dots, w_n(m)\}, \end{aligned} \quad (J-10)$$

где $r(j)$ представляет битовую матрицу, в которой изменяют коэффициенты взвешивания, $r(0) < r(1) < \dots < r(m)$ и $w_i(j)$ – это вес, используемый для поддиапазона i при диапазоне битовой скорости от $r(j)$ до $r(j+1)$. Каждый набор визуальных весов должен действовать внутри определенного диапазона битовой скорости. Если $m=0$, т. е. имеется только один набор визуальных весов, то есть это случай фиксированного визуального взвешивания. Для определения порядка встраивания в соответствующие диапазоны битовых скоростей следует использовать наборы визуальных весов от $W(0)$ до $W(m)$. При внедрении на высоких битовых скоростях, особенно при внедрении с кодированием во всем диапазоне – от кодирования с потерями до кодирования без потерь, – последние визуальные веса $W(m)$ все должны быть единицами (поскольку не может быть взвешивания при кодировании без потерь). Последовательное визуальное кодирование должно подстраивать визуальные веса, чтобы обеспечить высокое качество при всех битовых скоростях.

Взвешивание VIP влияет только на кодировщик, а декодеру никаких сигналов посылать не требуется.

Предполагается, что кодировщик вычислит порядок, в котором части битовых матриц блока кодирования должны появляться в иерархии слоев потока битов, основываясь на критерии искажений в зависимости от скорости передачи. Простое использование последовательного визуального взвешивания последовательно изменяет и метрику искажений, основанную на визуальном взвешивании во время формирования потока битов. Поскольку формирование потока битов приводит в действие оптимизация R-D после компрессии, изменение последовательного визуального взвешивания эффективно управляет порядком внедрения "на лету" частей битовых матриц блока кодирования.

J.12.4 Таблицы рекомендованных частотных взвешиваний

В таблице J.24 приведены три набора весов CSF, которые спроектированы для компонента яркости, основываясь на величине CSF на средней частоте каждого поддиапазона. Предполагается, что расстояние просмотра составляет, соответственно, 1000, 2000 и 4000 образцов (что соответствует 10 дюймам для плотностей 100, 200 и 400 точек на дюйм при печати или на экране). Отметим, что эти таблицы предназначены для 5-уровневой частотной декомпозиции.

В таблице J.24 не включен вес для самого нижнего частотного поддиапазона nLL, который всегда равен 1. Уровни 1, 2, ..., 5 обозначают уровни поддиапазонов в порядке от нижней до верхней частоты. (HL, LH, HH) обозначают три частотные ориентации в каждом поддиапазоне.

Таблица J.24 – Рекомендованные частотные взвешивания

Уровень	Расстояние просмотра 1000			Расстояние просмотра 2000			Расстояние просмотра 4000		
	HL	LH	HN	HL	LH	HN	HL	LH	HN
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	0,731 668
3	1	1	1	1	1	0,727 203	0,564 344	0,564 344	0,285 968
4	1	1	0,727 172	0,560 841	0,560 841	0,284 193	0,179 609	0,179 609	0,043 903
5	0,560 805	0,560 805	0,284 173	0,178 494	0,178 494	0,043 631	0,014 774	0,014 774	0,000 573

Для цветных изображений таблицы частотных взвешиваний компонентов Y, Cr и Cb должны отличаться порядком, чтобы создать преимущество свойствам визуальной системы человека. Например, обычно желательно подчеркнуть компонент яркости в большей степени, чем хроматические компоненты. В таблице J.25 приведены три набора весов CSF для хроматических компонентов и яркости.

Таблица J.25 – Рекомендованные частотные взвешивания для (цветных) изображений с множеством компонентов

Комп	Уровень	Расстояние просмотра 1000			Расстояние просмотра 1700			Расстояние просмотра 3000		
		HL	LH	HN	HL	LH	HN	HL	LH	HN
Y (Y0)	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1
	3	1	1	1	1	1	1	0,921 045	0,921 045	0,848 324
	4	0,998 276	0,998 276	0,996 555	0,861 593	0,861 593	0,742 342	0,410 628	0,410 628	0,182 760
	5	0,756 353	0,756 353	0,573 057	0,307 191	0,307 191	0,108 920	0,038 487	0,038 487	0,003 075
Cb (Y1)	1	0,883 196	0,883 196	0,833 582	0,818 766	0,818 766	0,745 875	0,717 086	0,717 086	0,613 777
	2	0,793 487	0,793 487	0,712 295	0,689 404	0,689 404	0,579 220	0,539 437	0,539 437	0,403 353
	3	0,650 482	0,650 482	0,531 700	0,501 652	0,501 652	0,362 279	0,319 773	0,319 773	0,185 609
	4	0,450 739	0,450 739	0,309 177	0,280 068	0,280 068	0,152 290	0,124 021	0,124 021	0,044 711
	5	0,230 503	0,230 503	0,113 786	0,097 816	0,097 816	0,031 179	0,023 308	0,023 308	0,003 413
Cr (Y2)	1	0,910 877	0,910 877	0,872 378	0,860 885	0,860 885	0,803 172	0,780 091	0,780 091	0,695 128
	2	0,841 032	0,841 032	0,776 180	0,757 626	0,757 626	0,665 951	0,631 632	0,631 632	0,509 729
	3	0,725 657	0,725 657	0,625 103	0,598 537	0,598 537	0,470 893	0,428 659	0,428 659	0,287 593
	4	0,552 901	0,552 901	0,418 938	0,388 492	0,388 492	0,248 566	0,211 871	0,211 871	0,100 658
	5	0,336 166	0,336 166	0,200 507	0,177 435	0,177 435	0,077 130	0,060 277	0,060 277	0,014 977

J.13 Вторичный отбор компонентов устройством кодирования

В некоторых применениях сжатия становится общепринятой практикой использовать вторичный отбор компонентов в сочетании с определенными декоррелирующими преобразованиями. Типичный пример – использование декоррелирующего преобразования RGB в YCrCb, за которым следует вторичный отбор компонентов хроматических данных (Cr, Cb). Несмотря на эффективность уменьшения размеров данных изображения для кодирования по алгоритму компрессии, основанному на DCT (Рекомендация МСЭ-Т Т.81 | ИСО/МЭК 10918-1), этот способ не рекомендован для использования в данной Рекомендации | Международном стандарте.

Природу частотного преобразования с многократным разрешением, описанную в этой Рекомендации | Международном стандарте, можно использовать, чтобы получить такой же эффект, какой дает вторичный отбор компонентов. Например, если поддиапазоны 1HL, 1LH и 1HN частотной декомпозиции компонентов отвергнуты, а все другие поддиапазоны сохранены, то в горизонтальном и вертикальном измерениях компонентов получают вторичный отбор с соотношением 2:1. Эта методика дает такие же преимущества, что и явный вторичный отбор компонентов до любого частотного преобразования.

Далее, с точки зрения качества изображения часто оказывается выгодным сохранить несколько частотных коэффициентов в поддиапазонах 1HL, 1LH, 1HN, в то же время удалив их значительную часть. В таких случаях число коэффициентов уменьшается примерно в соотношении 2:1, но результирующее декодирование изображения демонстрирует лучшее качество с меньшими погрешностями от сжатия. Использование методики вторичного отбора

не позволяет кодирующим устройствам сделать такой выбор и может ухудшить качество декодированного изображения.

J.14 Управление скоростью

Управление скоростью для выполнения особых задач по битовой скорости или времени передачи. Управление скоростью позволяет использовать в кодированном потоке требуемое число байтов, обеспечивая в то же время наивысшее достижимое качество изображения.

J.14.1 Введение в ключевые концепции управления скоростью

Разделять каждый поддиапазон на блоки кодирования образцов, которые следует кодировать независимо. Поскольку каждый блок кодирования кодируют полностью независимо, используя один и тот же алгоритм для всех поддиапазонов, в определенный момент времени связью между поддиапазонами и блоками кодирования можно пренебречь и полагать, что множество $\{B_i\}_{i=1,2,\dots}$ описывает набор всех блоков кодирования, которые представляют изображение. Для каждого блока B_i создают отдельный поток битов, используя любую информацию из любых других блоков. Кроме того, поток битов обладает свойством, позволяющим сократить его до множеств переменной длины $R_i^1, R_i^2, R_i^3, \dots$, а искажения при восстановлении каждого из этих подмножеств оценить и записать в виде $D_i^1, D_i^2, D_i^3, \dots$. Часто используют среднеквадратичную ошибку искажений, но в этом нет необходимости. Во время процесса кодирования множества длин R_i^n и искажений D_i^n вычисляют и временно хранят в компактной форме вместе со сжатым потоком битов.

Как только сжато все изображение, производят операцию последующей обработки над всеми сжатыми блоками кодирования и определяют степень, до которой каждый включенный в битовый поток блок кодирования может быть сокращен (усечен) с целью обеспечить заданную скорость, искажения и другие показатели качества. В общем виде, окончательно поток битов формируют из собрания так называемых "слоев", причем каждый слой интерпретируют в терминах общего качества изображения. Первый, самый нижний слой качества, формируют из оптимально сокращенных блоков кодирования потока битов описанным выше образом. Каждый последующий слой формируют из оптимально сокращенных блоков кодирования потока битов для последовательного достижения все более высоких битовых скоростей, уменьшения искажений и других возможных показателей качества, и включают дополнительные кодовые слова, которые требуются для дополнительной информации, представленной в предыдущих слоях для новых точек сокращения. Концепция формирования битового потока по слоям обсуждается далее в J.14.2.

J.14.2 Выделение слоев из потока битов

Важный аспект представляет способ, которым кодировщик формирует окончательный поток битов из независимо встроенных потоков битов, созданных для каждого блока кодирования. Проблема формирования потока битов значительно упрощается, если кодер обрабатывает сразу все поддиапазоны, поскольку отпадает необходимость в организации дополнительного пространства, которое требуется для независимых блоков кодирования.

В основном, поток битов организован как последовательность слоев, в которой каждый слой содержит дополнительные вклады от каждого блока (некоторые вклады могут быть пустыми), как показано на рисунке J.12. Точки сокращений блоков кодирования, связанные с каждым слоем, оптимальны в смысле соотношения скорость/искажения. Это означает, что поток битов, образованный отбрасыванием ряда менее важных слоев, всегда будет оптимальным по соотношению скорость/искажения. Если поток битов сокращают за счет слоя, то, строго говоря, поток не будет оптимальным, но отклонение от оптимальности будет мало, если велико число слоев. Так как число слоев возрастает, число байтов кода в каждом слое уменьшается. При этом наклоны характеристик скорость/искажения, которые связаны со всеми точками усечения (сокращения) блоков кодирования в слое, становятся все более подобными. Однако возрастает также число блоков, которые не вносят вклад в слой. Поэтому увеличивается заголовок, связанный с определением блоков, которые действительно вносят вклад в слой. Практика показала, что оптимальные по критерию ОСШ характеристики сжатия для современных приложений получают в тех случаях, когда число слоев примерно вдвое больше числа проходов по частям битовых матриц, который выполняет статистический кодер. Границы проходов по частям битовых матриц являются также точками сокращения блоков кодирования, включаемых в поток битов. Следовательно, в среднем каждый слой содержит вклады примерно от половины блоков кодирования, так что затраты на определение того, внес ли блок вклад в любой данный слой, гораздо меньше затрат на распознавание строгого порядка вкладов блоков кодирования. Кроме того, относительный вклад заголовка в общую битовую скорость не зависит от размеров изображения.

На рисунке J.12 проиллюстрированы вклады блоков кодирования в слои потока битов. Для упрощения показаны только пять слоев с семью блоками кодирования. Заметим, что не от всех блоков требуют вклад в каждый слой и что число байтов, вносимых блоками в данный слой, обычно колеблется в широких пределах. Заметим также, что операцию кодирования блока производят для каждого блока независимо и вертикально, несмотря на то что организация потока битов по слоям – горизонтальная с распределением кодирующих проходов блоков по разным слоям.

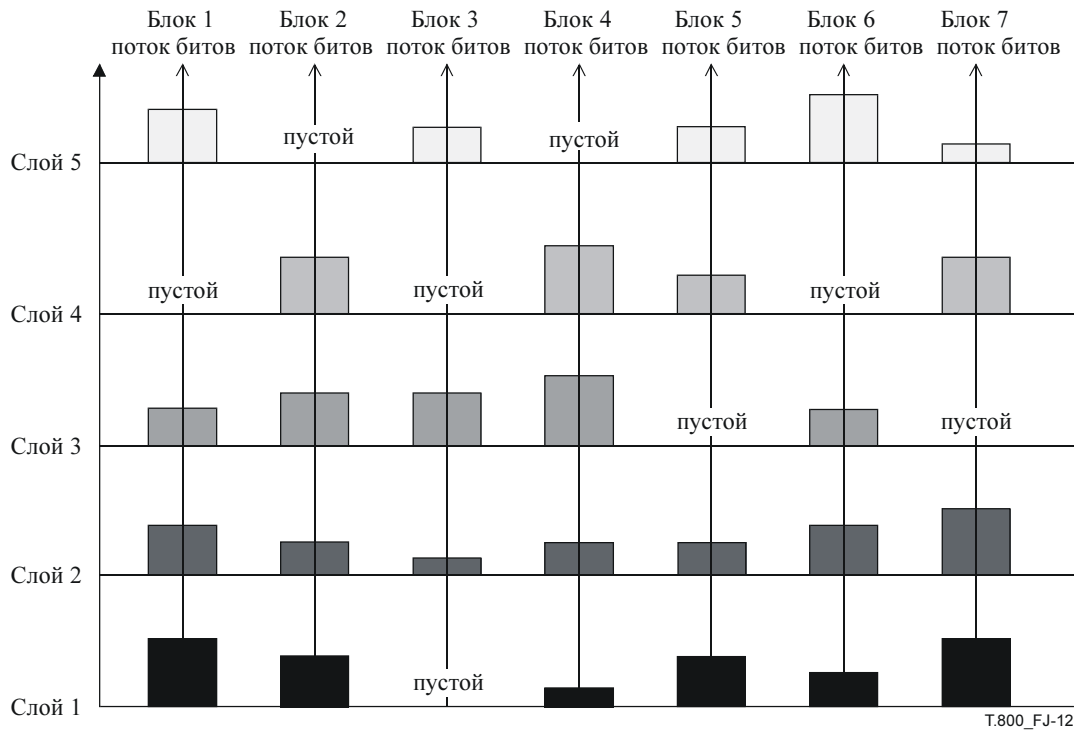


Рисунок J.12 – Иллюстрация вкладов блоков кодирования в слои потока битов

J.14.3 Оптимизация по критерию скорость/искажения

Описанный здесь алгоритм скорость/искажения справедлив только при условии, что для блоков принята аддитивная система сложения искажений. Так, искажение D в полностью восстановленном изображении должно удовлетворять равенству:

$$D = \sum_i D_i^{n_i}, \quad (J-11)$$

где n_i – точки сокращения (усечения) блока кодирования B_i . При соответствующей нормализации свойство аддитивности удовлетворяется среднеквадратической ошибкой (MSE) и взвешенной MSE (например, визуально взвешенной MSE), при условии ортогонального частотного преобразования. Аддитивность также сохраняется, если ошибки квантования для значений индивидуальных образцов не коррелированы, независимо от того, является ли преобразование ортогональным. Практически преобразование лишь приблизительно ортогонально, а ошибки квантования не коррелированы полностью. Поэтому даже среднеквадратичные показатели только приблизительно аддитивны, но обычно этого вполне достаточно. Предположим, что R – число байтов кода, связанных с некоторым слоем в потоке битов (и всеми предшествующими слоями). Тогда для некоторого набора точек усечения n_i :

$$R = \sum_i R_i^{n_i}. \quad (J-12)$$

Задача состоит в том, чтобы найти множество значений n_i , которое минимизирует D при ограничении $R \leq R_{max}$. Задача ограниченной оптимизации методом коэффициентов Лагранжа имеет хорошо известное решение. В частности, эта проблема эквивалентна минимизации

$$\sum (R_i^{n_i} - \lambda D_i^{n_i}), \quad (J-13)$$

где величину λ следует подстраивать в точках усечения до скорости, которая минимизирует соотношение (J-13) при $R = R_{max}$. Не существует простого алгоритма, который в общем виде может найти глобально оптимальное множество точек усечения. Однако можно гарантировать, что любое множество точек усечения n_i , которое минимизирует (J-13) для некоторого λ , будет оптимальным в том смысле, что при соответствующей скорости будет достигнут минимум искажений. Если для наибольшей величины λ , такой что множество точек усечения n_i , полученных оптимизацией суммы (J-13), приводит к скорости $R \leq R_{max}$, то невозможно найти никакое множество точек усечения, которое обеспечит меньшее общее искажение и скорость, которая будет меньше или равна R . Практика показала, что, как

правило, можно найти значения λ такие, что скорость R будет очень близка R_{max} (почти всегда в пределах 100 байтов), поэтому не совсем полная оптимальность не играет роли.

Теперь вернемся к проблеме минимизации выражения (J-13). Это отдельная проблема оптимизации для каждого отдельного блока кодирования. Конкретно для каждого блока B_i требуется найти такую точку усечения n_i , которая минимизирует $(R_i^{n_i} + \lambda D_i^{n_i})$. Это производят с помощью простого алгоритма следующим образом:

Положим $n_i = 0$ (то есть в блок не включена никакая информация)

Для $k = 1, 2, 3, \dots$

Положим $\Delta R_i^k = R_i^k - R_i^{n_i}$ и $\Delta D_i^k = D_i^k - D_i^{n_i}$

Если $(\Delta D_i^k / \Delta R_i^k) > \lambda^{-1}$, то положим $n_i = k$

Поскольку этот алгоритм следует выполнять для большого числа разных значений λ , имеет смысл предварительно идентифицировать подмножество пороговых значений N_i таких, что значения в соотношении скорость/искажения $S_i^k = \Delta D_i^k / \Delta R_i^k$ будут монотонно уменьшаться для всех k в N_i . Более конкретно подходящий алгоритм для определения N_i выглядит следующим образом:

- 1) Положим $N_i = \{n\}$, то есть множество всех точек усечения.
- 2) Положим $p = 0$
- 3) Для $k = 1, 2, 3, 4, \dots$

Если k принадлежит N_i

Положим $\Delta R_i^k = R_i^k - R_i^p$ и $\Delta D_i^k = D_i^k - D_i^p$

Положим $S_i^k = \Delta D_i^k / \Delta R_i^k$

Если $p \neq 0$ и $S_i^k > S_i^p$, то удалить p из N_i и перейти к шагу (2).

В противном случае положить $p = k$.

Как только эта информация предварительно вычислена, задача оптимизации для любой заданной величины λ состоит в том, чтобы просто установить значение p равным наибольшей величине k в N_i , такой, что $S_i^k > \lambda^{-1}$. Очевидно, величину λ можно интерпретировать как параметр качества, поскольку большие величины λ соответствуют менее жестким требованиям к усечению блока кодирования в потоке битов. По-другому это можно идентифицировать как порог наклона характеристики скорость/искажения.

Множество N_i и наклоны S_i^k вычисляются сразу после кодирования блока B_i . Этой информации достаточно, чтобы далее определить точки усечения, принадлежащие N_i и соответствующие значения R_i^k и S_i^k , которые хранят во время фазы оптимизации соотношения скорость/искажения. Обычно эта информация много меньше собственно потока битов, который хранят в блоке кодирования.

J.14.4 Эффективная оценка искажений для оптимального усечения R-D

Предполагаемые точки усечения для введения в поток битов, представляющий каждый блок кодирования, поступают на заключение в каждом кодирующем проходе. Во время сжатия следует оценить число байтов R_i^n , которые требуются для представления всех символов кода вплоть до каждой точки усечения n , а также искажение D_i^k как следствие усечения потока битов в каждой точке n . В действительности оценка искажений не строго обязательна для создания разрешенного способного к декомпрессии потока битов, но важно оценить алгоритм оптимизации, описанный в J.14.3.

J.14.4.1 Анализ необратимых преобразований

Алгоритм оптимизации скорость/искажения, описанный в J.14.3, зависит только от результата снижения искажений при каждом кодирующем проходе. В частности, если D_i^0 обозначает искажения вследствие полного игнорирования (пропуска) блока кодирования (например, если положить все образцы нулями), то для вычисления искажений при $n = 1, 2, 3, \dots$ необходима только разность $D_i^n - D_i^{n-1}$. Это вычисление, в свою очередь, можно выполнить с помощью двух небольших справочных таблиц, которые не зависят от кодирующего прохода, битовой матрицы или

поддиапазона. Чтобы рассмотреть этот вопрос, положим $\omega_i \Delta_i^2$ – вклад в искажение восстановленного изображения, который явился результатом ошибки в точности равной размеру одного шага образца блока кодирования B_i . Здесь ω_i имеет положительный вес, который вычисляются из нормы L2 частотного синтеза формы сигнала соответствующего поддиапазона и который могут дополнительно модифицировать, чтобы учесть визуальное взвешивание или другие критерии. Теперь определим:

$$v_i^p [m, n] = 2^{-p} v_i [m, n] - 2 \left[\frac{2^{-p} v_i [m, n]}{2} \right]. \quad (J-14)$$

Таким образом, $v_i^p [m, n]$ выражает нормализованную разность между размером образца $s_i[m, n]$ и наибольшим порогом квантования в предшествующей битовой матрице, у которой размер образца был не больше указанного. Легко проверить, что $0 \leq v_i^p [m, n] \leq 2$. Хотя в действительности $s_i[m, n]$ – квантованная целая величина, для общности будем считать, что квантователь может выдавать дробные биты для $s_i[m, n]$ и, следовательно, для $v_i [m, n]$, которые можно использовать в равенстве (J-14) для точного вычисления искажений, связанных с кодирующими проходами в менее значимых битовых матрицах. Теперь, когда единственный образец первым становится значимым в данной битовой матрице p , можно получить $v_i [m, n] \geq 2^p$ и, следовательно, $v_i^p [m, n] \geq 1$, а уменьшение искажений можно выразить как:

$$2^{2p} \omega_i \Delta_i^2 \left[v_i^p [m, n]^2 - (v_i^p [m, n] - 1,5)^2 \right] = 2^{2p} \omega_i \Delta_i^2 f_s (v_i^p [m, n]) \quad (J-15)$$

при условии, что уровни представления, использованные во время обратного квантования, расположены посередине между порогами квантования. Кроме того, уменьшение искажений, которое происходит за счет подстройки размеров образца в битовой матрице p , можно выразить как:

$$2^{2p} \omega_i \Delta_i^2 \left[(v_i^p [m, n] - 1)^2 - (v_i^p [m, n] - 0,5 - v)^2 \right] = 2^{2p} \omega_i \Delta_i^2 f_m (v_i^p [m, n]) \quad (J-16)$$

Таким образом, уменьшение искажений в результате одного кодирующего прохода можно вычислить, суммируя выходы, соответственно, одной из двух разных функций, $f_s(\cdot)$ или $f_m(\cdot)$, всякий раз, когда образец становится значимым или подстроены его размеры. Далее, результат в конце кодирующего прохода масштабируют с помощью постоянного коэффициента, который легко вычисляют из индекса битовой матрицы и значения $\omega_i \Delta_i^2$. Аргумент этих функций $v_i^p [m, n]$ в бинарной форме представлен как $v.xxxxx$, где v , единственный бит перед бинарной точкой, выражает битовый размер p , т. е. $v_i^p [m, n]$. Ровно 6 дополнительных битов после бинарной точки используют для индексации 7-битовой справочной таблицы для $f_m(\cdot)$ и 6-битовой справочной таблицы для $f_s(\cdot)$ (напомним, что мы должны получить $1 \leq v_i^p [m, n] < 2$, если образец первым становится значимым). Каждый вход этих справочных таблиц хранит 16-битовое представление с фиксированной запятой $2^{13} f_s (v_i^p [m, n])$ или $2^{13} f_m (v_i^p [m, n])$, соответственно. Это означает, что уменьшение общих искажений, связанное с любым заданным кодирующим проходом, может быть вычислено наращиванием этих целых величин в 32-битовом сумматоре без всякого риска переполнения.

J.14.4.2 Анализ обратимых преобразований

Обычно процесс оценки искажений для случая кодирования коэффициентов с помощью обратимого преобразования не отличается от оценки при необратимом преобразовании. Однако существуют два тонкие различия, которые должны быть здесь рассмотрены. Равенства (J-15) и (J-16) основаны на допущении, что деквантователь представит каждый коэффициент в средней точке соответствующего интервала квантования. Это наиболее вероятная для квантователя ситуация в течение наибольшей части времени, за исключением обработки наименее значимой битовой матрицы в обратимом режиме. В этом случае $\Delta_i = 1$, то есть без ошибок квантования. В данном случае восстановление средней точки не имеет смысла, и деквантователь представляет коэффициенты преобразования, используя нижний (по величине) порог соответствующего интервала квантования. Соответственно, равенства (J-15) и (J-16) можно модифицировать к следующему виду:

$$2^{2p} \omega_i \Delta_i^2 v_i^p [m, n]^2 = 2^{2p} \omega_i \Delta_i^2 f'_m (v_i^p [m, n]) \quad (J-17)$$

и

$$2^{2p} \omega_i \Delta_i^2 (v_i^p [m, n] - 1)^2 = 2^{2p} \omega_i \Delta_i^2 f'_s (v_i^p [m, n]), \quad (J-18)$$

соответственно.

J.15 Руководства по оперированию с кодированным потоком YCC

Существует множество приложений и устройств по использованию подвижных и неподвижных изображений, которые нельзя рассматривать без поддержки YCC и непосредственного вторичного отбора хроматических данных. В таких случаях может не потребоваться сигнализация о многократных преобразованиях компонентов в кодированном потоке. Этот раздел посвящен руководствам по обработке данных YCC.

Однако не предполагается, что данные YCC должны касаться вторичного отбора.

J.15.1 Использование многократного преобразования компонентов

Нет необходимости использовать многократное преобразование компонентов для поддержки данных YCC, поскольку компоненты уже декоррелированы. Поэтому сигнал параметра SGcod о многократном преобразовании компонентов, определенный в таблице A.17, всегда должен быть "0000 0000".

J.15.2 Использование формата JP2

Существуют устройства, у которых на выходе появляются данные YCC, автоматически преобразованные в пространство цвета sYCC. Формат JP2 поддерживает такие устройства, определяя значение EnumCS числом "18", как показано в таблице I.10.

J.15.3 Смещение хроматических данных

Смещение хроматических данных (общий термин для вторичного отбора хроматических компонентов и их относительных смещений) определяют в кодированном потоке JPEG 2000, используя маркер CRG (см. A.9.1). На рисунках J.13, J.14, J.15 и J.16 показаны примеры смещения хроматических данных хорошо известных рисунков. Для каждого рисунка в таблице J.26 приведены примеры параметров сегментов маркеров SIZ (см. A.5.1) и CRG.

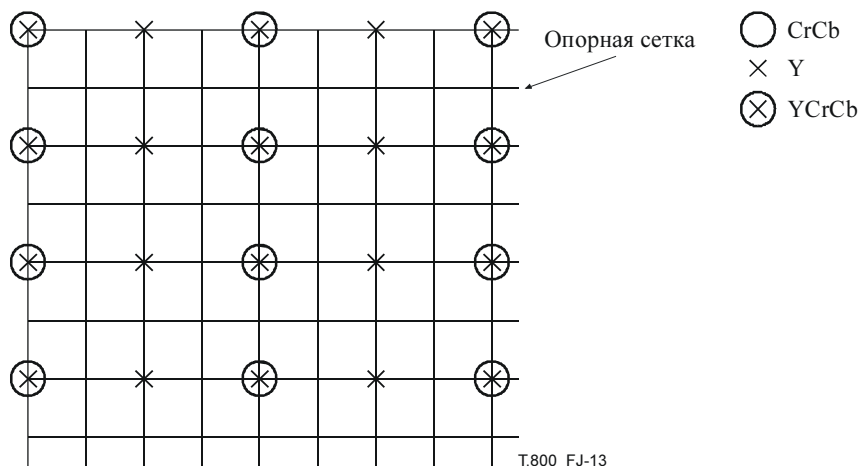


Рисунок J.13 – Формат 4:2:2 (взаимно смещенный)

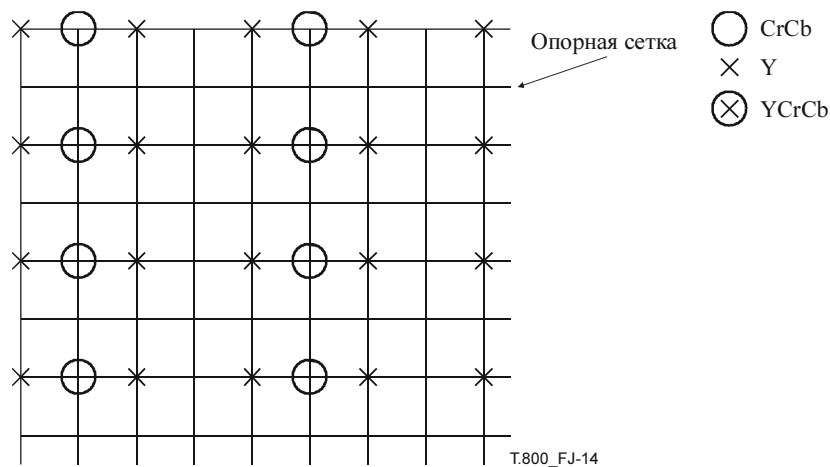


Рисунок J.14 – Формат 4:2:2 (центрированный)

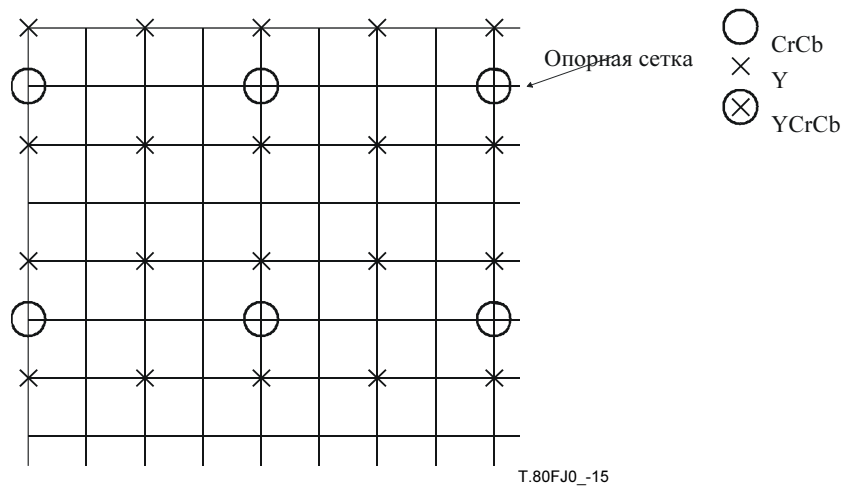


Рисунок J.15 – Формат 4:2:0 (взаимно смещенный)

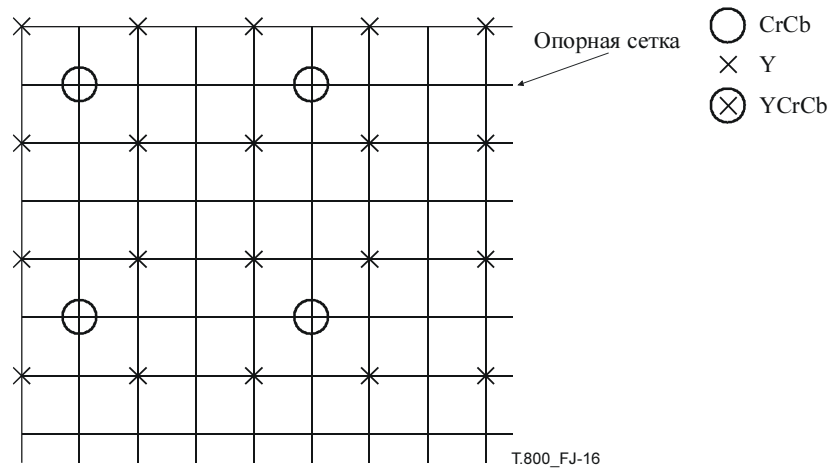


Рисунок J.16 – Формат 4:2:0 (центрированный)

Таблица J.26 – Значения CRG (регистрация компонентов)

		Рисунок J.13	Рисунок J.14	Рисунок J.15	Рисунок J.16
Y	(XRsiz, YRsiz)	(2, 2)	(2, 2)	(2, 2)	(2, 2)
	(XOsiz, YOsiz)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
	(Xcrg, Ycrg)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
Cb	(XRsiz, YRsiz)	(4, 2)	(4, 2)	(4, 4)	4, 4)
	(XOsiz, YOsiz)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
	(Xcrg, Ycrg)	(0, 0)	(16 384, 0)	(0, 16 384)	(16 384, 16 384)
Cr	(XRsiz, YRsiz)	(4, 2)	(4, 2)	4, 4)	4, 4)
	(XOsiz, YOsiz)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
	(Xcrg, Ycrg)	(0, 0)	(16 384, 0)	(0, 16 384)	(16 384, 16 384)

ПРИМЕЧАНИЕ. – Значение CRG определяют таким образом, чтобы образцы всех компонентов располагались на точках опорной сетки. Поэтому равенство $(XRsiz, YRsiz) = (2, 2)$ для Y не означает, что образцы Y также являются вторично отобранными.

Приложение К

Библиография

(Это Приложение не является составной частью данной Рекомендации | Международного стандарта)

К.1 Общее

- [1] BOLIEK (M.), GORMISH (M.J.), SCHWARTZ (E.L.), KEITH (A.): Decoding compression with reversible embedded wavelets (CREW) codestreams, *Electronic Imaging*, Vol. 7, No. 3, July 1998.
- [2] EBRAHIMI (T.), CHRISTOPOULOS (C.): JPEG 2000 standard: Still image compression scheme of 21st century, Tampere, Finland, 5-8 September 2000.
- [3] MARCELLIN (M.W.), GORMISH (M.), BILGIN (A.), BOLIEK (M.): An Overview of JPEG-2000, *Proc. of IEEE Data Compression Conference*, Snowbird, Utah, March 2000.
- [4] SKODRAS (A.), CHRISTOPOULOS (C.), EBRAHIMI (T.): JPEG2000: The upcoming still image compression standard, (*Invited paper*) *Proceedings of the 11th Portuguese Conference on Pattern Recognition (RECPAD 2000)*, pp. 359-366, Porto, Portugal, 11-12 May 2000.
- [5] ZANDI (A.), ALLEN (J.D.), SCWHARTZ (E.L.), BOLIEK (M.): CREW: Compression with reversible embedded wavelets, *Proceedings of Data Compression Conference*, Snowbird, Utah, pp. 212-221, March 1995.

К.2 Квантование и статистическое кодирование

- [6] ORDENTLICH (E.), WEINBERGER (M.J.), SEROUSSI (G.): A low-complexity modelling approach for embedded coding of wavelet coefficients, *Proceedings of Data Compression Conference*, pp. 408-417, Snowbird, Utah, March 29-April 1, 1998.
- [7] ORDENTLICH (E.), TAUBMAN (D.), WEINBERGER (M.J.), SEROUSSI (G.), MARCELLIN (M.): Memory Efficient Scalable Line-based Image Coding, *Proceedings of Data Compression Conference*, pp. 218-227, Snowbird, Utah, March 29-31, 1999.
- [8] PENNEBAKER (W.), MITCHELL (J.): JPEG Still Image Data Compression Standard, *Van Nostrand Reinhold*, New York, 1993.
- [9] SEMENTILLI (P.J.), BILGIN (A.), KASNER (J.H.), MARCELLIN (M.W.): Wavelet TCQ: Submission to JPEG 2000, (invited paper), *Proceedings, Applications of Digital Image Processing, SPIE*, San Diego, California, July 1998.
- [10] TAUBMAN (D.), ZAKHOR (A.): Multirate 3-D subband coding of video, *IEEE Trans. Image Processing*, Vol. 3, pp. 572-88, September 1994.
- [11] TAUBMAN (D.): High performance scalable image compression with EBCOT, *IEEE Trans. on Image Processing*, July 2000.
- [12] WOODS (J.W.), NAVEEN (J.): A filter based bit allocation scheme for subband compression of HDTV, *IEEE Trans. on Image Processing*, July 1992.

К.3 Частотное преобразование

- [13] ADAMS (M.D.), KOSENTINI (F.): Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis, *IEEE Trans. on Image Processing*, 2000.
- [14] ANTONINI (M.), BARLAUD (M.), MATHIEU (P.), DAUBECHIES (I.): Image coding using wavelet transform, *IEEE Trans. on Image Processing* Vol. 1, pp. 205-220, April 1992.
- [15] BRISLAWN (C.M.): Classification of nonexpansive symmetric extension transforms for multirate filter banks, *Applied and Computational Harmonic Analysis*, Vol. 3, pp. 337-57, 1996.
- [16] CALDERBANK (A.R.), DAUBECHIES (I.), SWELDENS (W.), YEO (B.-L.): Wavelet transforms that map integers to integers, *Applied and Computational Harmonic Analysis*, Vol. 5, No. 3, pp. 332-369, July 1998.

- [17] CHRYSAFIS (C.), ORTEGA (A.), Line-Based, Reduced Memory, Wavelet Image Compression, *IEEE Trans. on Image Processing*, Vol. 9, No 3, pp. 378-389, March 2000.
- [18] CHUI (C.K.): An Introduction to Wavelets, *Academic Press*, Boston, 1992.
- [19] CHUI (C.K.): Wavelets: A Mathematical Tool for Signal Analysis, *SIAM Publ.*, Philadelphia, 1997.
- [20] DAUBECHIES (I.): Ten Lectures on Wavelets, *SIAM Publ.*, Philadelphia, 1992.
- [21] DAUBECHIES (I.), SWELDENS (W.): Factoring wavelet transforms into lifting steps, *Journal of Fourier Analysis and Applications*, Vol. 4, No. 3, pp. 247-269, 1999.
- [22] LEGALL (D.), TABATABAI (A.): Subband coding of digital images using symmetric short kernel filters and arithmetic coding techniques, *IEEE International Conference on Acoustics, Speech and Signal Processing*, New York, NY, pp. 761-765, 1988.
- [23] MALLAT (S.): A Wavelet Tour of Signal Processing, *Academic Press*, 2nd Edition, San Diego, 1999.
- [24] SHENG (F.), BILGIN (A.), SEMENTILLI (P.J.), MARCELLIN (M.W.): Lossy and lossless image compression using reversible integer wavelet transforms, *Proceedings, IEEE International Conference on Image Processing*, Chicago, Illinois, October 1998.
- [25] SWELDENS (W.): The lifting scheme: a custom-design construction of biorthogonal wavelets, *Applied and Computational Harmonic Analysis*, Vol. 3, No. 2, pp 186-200, 1996.
- [26] SWELDENS (W.): The lifting scheme: A construction of second generation wavelets, *SIAM Journal on Mathematical Analysis*, Vol. 29, No. 2, pp 511-546, 1997.

К.4 Кодирование области интересов

- [27] CHRISTOPOULOS (C.), ASKELOF (J.), LARSSON (M.): Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard, *IEEE Signal Processing letters*, September 2000.
- [28] ATSUMI (E.), FARVARDIN (N.): Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees, *Proc. IEEE International Conference on Image Processing (ICIP-98)*, pp. 87-91, 4-7 October 1998, Chicago, Illinois.
- [29] NISTER (D.), CHRISTOPOULOS (C.): Lossless Region of Interest with a naturally progressive still image coding algorithm, *Proceedings of IEEE International Conference on Image Processing (ICIP 98)*, pp. 856-860, 4-7 October 1998, Chicago, Illinois,
- [30] NISTER (D.), CHRISTOPOULOS (C.): Lossless region of interest with embedded wavelet image coding, *Signal Processing*, Vol. 78, No. 1, pp. 1-17, October 1999.
- [31] SANTA CRUZ (D.), LARSSON (M.), ASKELOF (J.), EBRAHIMI (T.), CHRISTOPOULOS (C.): Region of Interest coding in JPEG2000 for interactive client/server applications, *IEEE International Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, 13-15 September 1999.

К.5 Визуальное частотное взвешивание

- [32] ALBANESI (M.), BERTOLUZZA (S.): Human vision model and wavelets for high-quality image compression, *Proc. of 5th Int. Conference in Image Processing and its Applications*, Edinburgh, UK, No. 410, pp. 311-315, 4-6 July 1995.
- [33] ECKERT (M.): Lossy compression using wavelets, block DCT, and lapped orthogonal transforms optimised with a perceptual model, *SPIE*, Vol. 3031, pp. 339-350, 1997.
- [34] O'ROURKE (T.), STEVENSON (R.): Human visual system based wavelet decomposition for image compression, *J. VCIP V. 6*, pp. 109-121, 1995.
- [35] WATSON (A.B.), YANG (G.), SOLOMON (J.), VILLASENOR (J.): Visibility of wavelet quantization noise, *IEEE Trans. on Image Processing*, Vol. 6, pp. 1164-1175, 1997.

К.6 Защита от ошибок

- [36] MAN (H.), KOSENTINI (F.), SMITH (M.): A Family of Efficient and Channel Error Resilient Wavelet/Subband Image Coders, *Special issue of the IEEE Transactions on Circuits and Systems for Video Technology on Interactive Multimedia*, 9(2), February 1999.

- [37] LIANG (J.), TALLURI (R.): Tools for Robust Image and Video Coding in JPEG2000 and MPEG-4 Standards, *Proceedings of SPIE Visual Communications and Image Processing Conference (VCIP)*, January 1999, San Jose, CA.
- [38] MOCCAGATA (I.), SODAGAR (S.), LIANG (J.), CHEN (H.): Error Resilient Coding in JPEG-2000 and MPEG-4, *IEEE Journal of Selected Areas in Communications (JSAC)*, June 2000.

К.7 Кодирование, основанное на сканировании

- [39] LAMBERT-NEBOUT (C.), LATRY (C.), MOURY (G.), ANTONINI (M.), BARLAUD (M.), PARISOT (C.): On-Board Optical image compression for future high resolution space remote sensing systems, *Proc. SPIE*, Vol. 4115, 31 July-3 August 2000.
- [40] FLOHR (T.J.), MARCELLIN (M.W.), ROUNTREE (J.C.): Scan-Based Processing with JPEG 2000, *Proc. SPIE*, Vol. 4115, 31 July-3 August 2000.

К.8 Цвет

- [41] International Color Consortium (ICC), *ICC Profile Format Specification 1:1998-09*, (1998).
- [42] IEC 61966-2-1 (1999), *Multimedia systems and equipment – Colour measurement and management – Part 2-1: Colour management – Default RGB colour space – sRGB*.
- [43] LAM (K.M.): Metamerism and Colour Constancy, *University of Bradford*, 1985.
- [44] LUO (M.R.), LO (M.C.), KUO (W.G.): The LLab(l:c) Colour Model, *Color Research and Application*, Vol. 21, pp. 412-429 (1996).
- [45] LUO (M.R.), HUNT (R.W.G.): A Chromatic Adaptation Transform and a Colour Constancy Index, *Color Research and Application*, Vol. 23, pp. 154-158, 1997.
- [46] LUO (M.R.), HUNT (R.W.G.): The Structure of the CIE 1997 Colour Appearance Model (CIECAM97s), *Color Research and Application*, Vol. 23, 1997.
- [47] NIELSEN (M.), STOKES (M.): The Creation of the sRGB ICC Profile, *Proceedings of the IS&T/SID 6th Color Imaging Conference*, pp. 253-257, 1998.
- [48] GIORGIANNI (E.), MADDEN (T.): Digital Color Management: Encoding Solution, *Addison-Wesley*, 1998.
- [49] SPAULDING (K.E.), WOOLFE (G.J.), GIORGIANNI (E.J.): Reference Input/Output Medium Metric RGB color encodings (RIMM/ ROMM RGB), *Proc. PICS 2000 Conference*, March 26-29, 2000, Portland, OR.

Приложение L

Заявление о патентных правах

(Это Приложение не является составной частью данной Рекомендации | Международного стандарта)

Международная организация по стандартизации (ИСО) и Международная электротехническая комиссия (МЭК) обращают внимание на то, что соответствие настоящей Рекомендации | Международному стандарту может стать причиной использования патентов, указанных в таблице L.1.

Таблица L.1 – Заявление о правах на интеллектуальную собственность

Номер	Компания
1	Algo Vision plc
2	Canon Incorporated
3	Digital Accelerator Corporation
4	Telefonaktiebolaget LM Ericsson
5	Hewlett Packard Company
6	International Business Machines, Inc.
7	LizardTech, Incorporated
8	LuraTech
9	Mitsubishi Electric Corporation
10	PrimaComp Incorporated
11	Ricoh Company, Limited
12	Sarnoff Corporation
13	Sharp Corporation
14	TeraLogic Incorporated
15	University of Arizona
16	Washington State University
17	Sony Corporation

ИСО и МЭК не несут ответственность за очевидность, законность и границы прав этих патентов.

Владельцы прав на эти патенты заверили ИСО и МЭК, что они готовы вести переговоры с претендентами по всему миру о лицензировании на разумных и не дискриминационных условиях и сроках. В связи с этим заявления держателей прав на эти патенты зарегистрированы в ИСО и МЭК.

Обращаем внимание на возможность того, что некоторые элементы данной Рекомендации | Международного стандарта могут стать предметом иных патентных прав, кроме указанных выше. ИСО и МЭК не несут ответственность за идентификацию любых или всех таких прав.

Индекс

C

CME 56

COC 34

COD 29

CRG 54

E

EOC 25

EPH 53

H

HH 59

определение 4

HL 59

определение 4

I

ICC 146, 195

J

JP2 143

определение 4

L

LH 59

определение 4

LL

определение 4

P

PLM 46

PLT 48

POC 42

PPM 49

PPT 51

Q

QCC 40

QCD 38

R

ROI 36

S

SIZ 26

SOC 21

SOD 24

SOP 52

SOT 22

Т

TLM 44

А

арифметический кодер 82
определение 2
пример 203

Б

битовая матрица
определение 2

блок 2, 143, 148

блок кодирования 59, 67
определение 3
сигнализация 31

В

визуальное частотное взвешивание 206

Г

границная область 59, 66
определение 5
сигнализация 31

Д

декодер 9

дерево тегов 71

дискретное частотное преобразование 116

дискретное частотное преобразование
определение 3
пример 206
"по рядам" 187
сигнализация 31

З

завершение
защита от ошибок 194

заголовок 13
определение 4

заголовок пакета 70
определение 5
пример 203

заголовок части тайла 20, 22
определение 6
пример 202

запуск маркера данных 24

ИСО/МЭК 15444-1:2002 (R)

запуск маркера кодированного потока 21
запуск сегмента маркера граничной области 30
запуск сегмента маркера пакета 52
запуск сегмента маркера тайла 22
защита от ошибок 194
знаковый бит 103
определение 6

И

избирательный обход арифметического кодирования 108
определение 6
изображение
определение 4

К

квантование 113, 115
определение 5
сигнализация 38, 40
кодирование, основанное на сканировании 194
кодированный поток 16
определить 3
кодировщик 9
кодирующий проход 102
определение 3
компонент тайла 59
определение 6

М

маркер 13
определение 5
маркер конца изображения 25
многократно преобразовать компонент 147
сигнализация 30
многократное преобразование компонента 136

Н

необратимый
определение 4
многократно преобразовать компонент 136
преобразовать по частоте 124, 133

О

область интересов
определение 5
сигнализация 36
обратимый
определение 6
многократно преобразовать компонент 136
преобразовать по частоте 124, 133

окончание кодированного потока 107

опорная сетка 26, 59

определение 5

основной заголовок 19, 21

определение 5

пример 201

П

пакет 68

определение 5

поддиапазон 59, 65

определение 6

порядок 117

положение значимости 102

порядок последовательности 77, 79

сигнализация 30, 42

поток битов

определение 2

преобразование

определение 6

проход значимости воспроизведения 102

определение 6

проход очистки 105

определение 2

проход подстройки величины 104

определение 5

Р

размер шага квантования

визуальное частотное взвешивание 206

разрешение 59, 64

определение 5

С

сдвиг уровня DC 135

сегмент маркера 13

определение 5

сегмент маркера длин частей тайла 44

сегмент маркера длины пакета 46

сегмент маркера длины пакетов 48

сегмент маркера изменения порядка последовательности 42

сегмент маркера квантования компонента 40

сегмент маркера квантования по умолчанию 38

сегмент маркера комментария и расширения 56

сегмент маркера конца заголовка пакета 30, 53

сегмент маркера области интересов 36

ИСО/МЭК 15444-1:2002 (R)

сегмент маркера размеров изображения и тайла 26

сегмент маркера регистрации компонента 54

сегмент маркера типа кодирования компонента 34

сегмент маркера упакованных заголовков пакетов 49, 51

символ сегментации 108, 195

определение 6

слой 67

определение 5

сигнализация 30

суперблок 154, 167, 175

Т

тайл 26, 60, 76

определение 6

тип кодирования по умолчанию 29

У

управление скоростью 209

уровень декомпозиции 59, 116

определение 3

сигнализация 31

Ф

файловый формат 143

определение 3

формирование контекста причины по вертикали 110

Ч

часть тайла 76

определение 6

СЕРИИ РЕКОМЕНДАЦИЙ МСЭ-Т

Серия А	Организация работы МСЭ-Т
Серия В	Средства выражения: определения, символы, классификация
Серия С	Общая статистика электросвязи
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	TMN и техническое обслуживание сетей: международные системы передачи, телефонные, телеграфные, факсимильные и арендованные каналы
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 и взаимосвязь открытых систем
Серия Y	Глобальная информационная инфраструктура и аспекты межсетевого протокола (IP)
Серия Z	Языки и общие аспекты программного обеспечения для систем электросвязи