ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 55697— 2013

ТЕЛЕВИДЕНИЕ ВЕЩАТЕЛЬНОЕ ЦИФРОВОЕ

Сервисная информация. Общие технические требования

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Федеральным государственным унитарным предприятием «Ордена Трудового Красного Знамени научно-исследовательский институт радио» (ФГУП НИИР)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 480 «Связь»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 8 ноября 2013 г. №1338-ст

4 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном формационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (gost.ru)

© Стандартинформ, 2014

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины и определения 1
4 Обозначения и сокращения 2
5 Структура сервисной информации 4
6. Таблицы сервисной информации 8
Приложение А (обязательное). Дескрипторы программ и элементов программ 29
Приложение Б (обязательное). Определение дескрипторов и их местоположение 30
Приложение В (обязательное). Модель декодера CRC 32
Приложение Г (обязательное). Значения поля stream_type 33
Приложение Д (обязательное). Преобразования между временем и датой 34
Библиография 38

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ТЕЛЕВИДЕНИЕ ВЕЩАТЕЛЬНОЕ ЦИФРОВОЕ
Сервисная информация. Общие технические требования

Digital video broadcasting
Service Information (SI). General technical requirements

Дата введения — 2014—09—01

1 Область применения

Настоящий стандарт распространяется на данные сервисной информации, входящие в состав транспортных потоков систем DVB первого и второго поколения, он устанавливает:

- методы передачи сервисной информации;
- структуры, используемые для передачи сервисной информации;
- синтаксис структур передачи сервисной информации.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты:

ГОСТ Р 52210–2004 Телевидение вещательное цифровое. Термины и определения

ГОСТ Р 52591–2006 Система передачи данных пользователя в цифровом телевизионном формате. Основные параметры

ГОСТ 17657–79 Передача данных. Термины и определения

ГОСТ 24375–80 Радиосвязь. Термины и определения

ГОСТ 24402–88 Телеобработка данных и вычислительные сети. Термины и определения

Примечание – При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования – на официальном сайте Федерального агентства по техническому регулированию и метрологии по стандартизации в сети Интернет или по ежегодно издаваемому информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения, Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины и определения

В настоящем стандарте применены термины по ГОСТ Р 52210, ГОСТ Р 52591, ГОСТ Р 52722, ГОСТ Р 53528, ГОСТ 17657, ГОСТ 24375, ГОСТ 24402, а также следующие термины с соответствующими определениями, заимствованными из следующих документов международных организаций [1 – 5]:

3.1 **дескриптор** (descriptor): Структура данных, служащая для описания типа передаваемых данных и их значений.

3.2 **идентификатор (identifier):** Мнемоника, определенная для описания различных типов данных, которые используются в закодированном битовом потоке.

3.3 **конфиденциальная информация (private information):** Конфиденциальная информация представляет собой любые данные пользователя, которые не кодируются в соответствии со стандартом MPEG-2 Systems (ИСО/МЭК 13818–1:2013).

3.4 **конфиденциальная секция (private section):** Все типы секций, за исключением секций, определенных в стандарте MPEG-2 Systems (ИСО/МЭК 13818–1:2013).

Издание официальное

3.5 **мнемокод** (mnemonic code): Мнемоническое обозначение в стандартах MPEG-2.
3.6 **открытая секция** (non-private section): Все типы секций программно-зависимой информации, определенные в стандарте MPEG-2 Systems (ИСО/МЭК 13818–1:2013).
3.7 **мнемоника** (mnemonic): Совокупность специальных приемов и способов, облегчающих запоминание и запись нужной информации.
3.8 **пакет** (packet): Последовательность информационных и служебных символов цифрового вещательного телевидения, передаваемая, обрабатываемая и коммутируемая как единое целое.
3.9 **программа** (programme): Совокупность одного или большего числа событий под управлением вещателя.
3.10 **программно-зависимая информация** (program specific information): Совокупность нормативных данных, которые необходимы для демультиплексирования транспортных потоков и успешного восстановления программ.
3.11 **секция** (section): Синтаксическая структура, которая должна использоваться для преобразования каждой таблицы в информацию пакетов транспортного потока.
3.12 **сервисная информация** (service information): Служебные данные, передаваемые в транспортных пакетах и описывающие систему доставки, содержание и расписание вещаемых программ и т.д.
3.13 **синтаксис** (syntax): Часть языка программирования, которая описывает структуру программ как набор символов.
3.14 **служба** (service): Последовательность программ под управлением вещателя, которая может передаваться как часть расписания программы.
3.15 **событие** (event): Относящаяся к обычной службе группировка передаваемых элементарных потоков данных с определенным временем начала и конца.
3.16 **список инструментов IPMP** (IPMP tool list): Перечень необходимых инструментов (программных модулей) IPMP, служащих для защиты последовательности данных.
3.17 **субтаблица** (sub-table): Часть таблицы, содержащая сходные данные. Структурно – это набор секций с одинаковыми значениями табличного идентификатора table_id, расширения табличного идентификатора table_id_extension и номера версии version_number.
3.18 **таблица сервисной информации** (service information table): Совокупность секций, имеющих одинаковое значение table_id.
3.19 **транспортный пакет** (transport packet): Пакет данных фиксированной длины 188 байтов, содержащий заголовок длиной 4 байта и поля адаптации и полезной нагрузки общей длиной до 184 байтов.
3.20 **транспортный поток** (transport stream): Метод мультиплексирования в один поток и пакетной передачи последовательности транспортных пакетов.
3.21 **частичный транспортный поток** (partial transport stream): Поток битов, полученный из транспортного потока MPEG-2 удалением тех пакетов транспортного потока, которые не относятся к одной определенной выбранной программе или нескольким выбранным программам.
Примечание – "Частичные" (например записанные) транспортные потоки представляют собой подмножество потоков данных, полученных из исходного транспортного потока.

4 Обозначения и сокращения

В настоящем стандарте применены следующие обозначения и сокращения:

4.1 adaptation_field поле – поле длиной 2 бита указывает, следует ли после этого заголовка пакета транспортного потока поле адаптации и/или полезная нагрузка.
4.2 ADTS (Audio Data Transport Stream) – аудиоданные транспортного потока.
4.3 AVC (Advanced Video Coding) – поток видеоданных, кодированных в соответствии со стандартом MPEG 4.10.
4.4 BAT (Bouquet Association Table) – таблица взаимосвязи программных пакетов.
4.5 bslbf (bit string, left bit first) – мнемоническое обозначение в стандартах MPEG-2, означающее "последовательность битов, левый бит первый".
4.6 CAS (Condition Access System) – система условного доступа.
4.7 CAT (Conditional Access Table) – таблица условного доступа.
4.8 CRC (Cyclic Redundancy Check) – проверка по четности циклическим избыточным кодом (циклическая проверка на основе избыточности).
4.9 DEC (decimal) – индекс десятичной системы счисления.

4.10 descriptor_tag – поле длиной 8 бит, которое служит меткой для идентификации каждого дескриптора.
4.11 DIT (Discontinuity Information Table) – таблица неоднородности информации.
4.12 DSM-CC (Digital Storage Media Command and Control) – система команд и управления для средств цифровой записи.
4.13 EIT (Event Information Table) – таблица информации о программном элементе.
4.14 EMM (entitlement management message) – сообщение, предоставляющее право доступа.
4.15 FlexMux поток – в стандарте MPEG-4 – последовательность элементарных потоков с одинаковыми требованиями по качеству обслуживания, объединенных в один поток.
4.16 HEX (hexadecimal) – индекс шестнадцатиричной системы счисления.
4.17 IPMP (Intellectual Property Management and Protection) – стандарт управления и защиты прав интеллектуальной собственности на цифровые данные формата MPEG-2.
4.18 LATM (Low Overhead Audio Transport Multiplex) – мультиплексирование транспортных потоков аудио с низким уровнем служебных данных.
4.19 MHEG (Multimedia/Hypermedia Expert Group) – группа экспертов по кодированию мультимедиа и гипермедиа.
4.20 MJD (Modified Julian Date) – модифицированная юлианская дата.
4.21 MPEG-2 (Moving Picture Experts Group) – обобщенное название группы стандартов ИСО/МЭК 13818 в области кодирования, обработки и транспортировки сигналов изображения и звука.
4.22 NIT (Network Information Table) – таблица сетевой информации.
4.23 NVOD (Near Video on Demand) – услуга "почти видео по запросу".
4.24 PAT (Program Association Table) – таблица взаимосвязи программ (таблица программ).
4.25 PCR (Program Clock Reference) – временные метки транспортного потока, при помощи которых происходит временная синхронизация в декодере.
4.26 PES (Packetized Elementary Stream) – пакетированный элементарный поток.
4.27 PID (packet identifier) – идентификатор пакета данных (поле длиной 13 бит, указывающее тип данных, находящихся в полезной нагрузке пакета).
4.28 PMT (Program Map Table) – таблица структуры программы.
4.29 pointer_field – поле транспортного пакета длиной 8 бит, значением которого должно быть количество байтов, которые следуют непосредственно после поля pointer_field до первого байта первой секции, которая присутствует в полезной нагрузке пакета транспортного потока.
Примечание – Значение 0x00 поля pointer_field указывает, что секция начинается сразу же после поля pointer_field.
4.30 PSI (Program Specific Information) – программно-зависимая информация.
4.31 rpchof (remainder polynomial coefficients, highest order first) – мнемоническое обозначение в стандартах MPEG-2, означающее "коэффициенты многочлена остатка (от деления), наибольшая степень следует первой".
4.32 RST (Running Status Table) – таблица текущего статуса.
4.33 SDT (Service Description Table) – таблица описания услуги.
4.34 SI (Service Information) – сервисная информация.
4.35 SIT (Selection Information Table) – таблица выбираемой информации
4.36 ST (Stuffing Table) – таблица байтов согласования скоростей.
4.37 TDT (Time and Date Table) – таблица времени и даты.
4.38 TOT (Time Offset Table) – таблица сдвига по времени.
4.39 TSDT (Transport Stream Description Table) – таблица описания транспортного потока.
4.40 uimsbf (signed integer, most significant bit first) – мнемоническое обозначение в стандартах MPEG-2, означающее “целое число, самый старший бит первый”.
4.41 UTC (Universal Time, Co-ordinated) – Всемирное координированное время.
4.42 ЕТСИ (ETSI) – Европейский институт стандартов связи.
4.43 ИСО (ISO) – Международная организация по стандартизации.
4.44 МЭК (IEC) – Международная электротехническая комиссия.
4.45 ПСП (PRS) – псевдослучайная последовательность.
4.46 СУД (CAS) – система условного доступа.
4.47 ТП (TS) – транспортный поток.

5 Структура сервисной информации

5.1 Общие положения

5.1.1 В составе транспортных потоков систем цифрового ТВ вещания DVB различного назначения должны передаваться данные сервисной (служебной) информации SI, описывающие систему доставки, содержание и расписание вещаемых потоков данных в мультиплексах, доступных пользователю в месте приема [1, 2].

5.1.2 В состав сервисной информации SI должна входить группа служебных данных программно-зависимой информации в виде 6 таблиц PSI, которые необходимы для демультиплексирования транспортных потоков и успешного восстановления программ в пределах одного мультиплекса [3]:

1 Таблица взаимосвязи программ PAT;
2 Таблица структуры программы PMT;
3 Таблица условного доступа CAT;
4 Таблица описания транспортного потока TSDT;
5 Таблица сетевой информации NIT;
6 Таблица управляющей информации IPMP;

5.1.3 Данные сервисной информации SI должны содержать также дополнительную служебную информацию, которая предоставляет пользователям возможность получения описания услуг (служб) и программных элементов (событий), передаваемых в различных мультиплексах разных сетей. Дополнительные данные SI должны быть организованы в 9 таблиц [1]:

1 Таблица взаимосвязи программных пакетов BAT;
2 Таблица описания услуги SDT;
3 Таблица информации о программных элементах EIT;
4 Таблица времени и даты TDT;
5 Таблица сдвига по времени TOT;
6 Таблица текущего статуса RST;
7 Таблица байтов согласования скоростей ST;
8 Таблица неоднородности информации DIT;
9 Таблица выбираемой информации SIT.

5.1.4 В частичных транспортных потоках не должны передаваться какие-либо таблицы сервисной информации за исключением копий таблиц PAT и PMT из состава информации PSI, а также таблицы выбираемой информации SIT и таблицы неоднородности информации DIT.

5.2 Программно-зависимая информация

5.2.1 Программно-зависимая информация PSI может содержать как нормативную информацию, так и конфиденциальную информацию, которая позволяет выполнять в декодерах демультиплексирование программ. Один или несколько элементарных потоков, образующих программу, должны быть идентифицированы своим PID. Для реализации условного доступа программы, элементарные потоки или их части могут быть скремблированы (зашифрованы). Программно-зависимая информация PSI не должна подвергаться скремблированию (шифрованию) [3].

5.2.2 В транспортных потоках программно-зависимая информация PSI должна быть организована в шесть табличных структур, содержащих информацию, необходимую для автоматического конфигурирования приемника, который разделяет и декодирует различные программные потоки, входящие в состав мультиплекса. Хотя эти структуры могут считаться простыми таблицами, они должны быть разделены на одну или несколько секций и введены в пакеты транспортного потока, некоторые с заранее определенными значениями PID, а другие с теми значениями PID, которые выбирает пользователь. Название таблиц программно-зависимой информации PSI, зарезервированные за ними номера идентификаторов PID, а также назначение таблиц приведены в таблице 1 [3].

Таблица 1

Название таблицы	Зарезервированный номер PID	Назначение таблицы
Таблица взаимосвязи программ (PAT)	0x00	Связывает номер программы и PID таблицы структуры программы PMT
Таблица структуры программы (PMT)	Присвоен в PAT	Содержит значения PID для компонентов одной или нескольких программ

Окончание таблицы 1

Название таблицы	Зарезервированный номер PID	Назначение таблицы
Таблица условного доступа (CAT)	0x01	Присваивает одному или каждому из нескольких (частных) потоков EMM уникальное значение PID
Таблица описания транспортного потока (TSDT)	0x02	Связывает один или несколько дескрипторов со всем транспортным потоком
Таблица сетевой информации (NIT)	Присвоен в PAT	Является дополнительной. Содержит сведения о физических параметрах доступных сетей (частоты каналов, номера спутниковых транспондеров, характеристики модуляции и т. п.) и сведения о мультиплексах, передаваемых в этих сетях
Таблица управляющей информации IPMP	0x03	Содержит список инструментов IPMP, контейнер прав, контейнер инструментов, определенные в стандарте MPEG-2

5.3 Метод описания синтаксиса двоичного потока

5.3.1 Математические операторы, используемые в настоящем стандарте для описания логических конструкций данных, аналогичны тем, что используются в языке программирования "С". Каждый блок данных в двоичном потоке напечатан жирным шрифтом. Он описывается своим названием, длиной в битах, мнемокодом своего типа и порядком передачи.

5.3.2 Действие, выполняемое элементом декодированных данных в двоичном потоке, зависит от значения этого элемента данных и от ранее декодированных элементов данных. Декодирование элементов данных и определение состояния переменных, использованных при их декодировании, описываются в разделах, содержащих семантическое описание синтаксиса. Нижеприведенные конструкции используются для описания условий, в которых элементы данных представлены и имеют нормальный тип [3].

Описание этого синтаксиса выполнено с использованием условных обозначений кодов "С", в соответствии с которыми, если переменная или выражение имеет ненулевое значение, это означает, что условие выполняется (имеет значение "true"):

while (условие) {
data_element
. . .
}
	Если это условие выполняется, то далее в потоке данных появляется группа элементов данных **data_element**. Это повторяется до тех пор, пока это условие не перестанет выполняться.

do {
data_element
. . .
}
while (условие)
	Элемент данных всегда появляется, как минимум, один раз. Этот элемент данных повторяется до тех пор, пока не перестанет выполняться это условие.

if (условие) {
data_element
. . .
}
	Если это условие выполняется, то далее в потоке данных появляется первая группа элементов данных **data_element**.

else {
data_element
. . .
}
	Если это условие не выполняется, то далее в потоке данных появляется вторая группа элементов данных **data_element**.

for (i = 0; i < n; i++) {
data_element
. . .
}
	Группа элементов данных появляется *n* раз. Условные конструкции внутри группы элементов данных могут зависеть от значения переменной *i* управления циклом, которая для первого появления устанавливается равной нулю, и увеличивается на 1 при втором появлении и так далее.

5.4 Таблицы и секции

5.4.1 Данные сервисной SI и программно-зависимой PSI информации должны быть структурно представлены в форме таблиц, сегментированных на одну или несколько секций передаваемых в пакетах транспортного потока.

5.4.2 Вместе с таблицами PSI допускается передавать таблицы конфиденциальной информации. Конфиденциальная информация может быть структурирована тем же способом, который используется для передачи таблиц PSI, и синтаксис для преобразования этой конфиденциальной информации должен быть идентичен синтаксису, используемому для преобразования информации таблиц PSI. Для этих целей определяется конфиденциальная секция (**private_section**). Если конфиденциальная информация передается в пакетах транспортного потока с тем же значением PID, что и у пакетов транспортного потока, в которых передаются таблицы структуры программы PMT (как определено в таблице ассоциаций для программы PAT), то должны использоваться синтаксис и семантика конфиденциальной секции (**private_section**). Данные, передаваемые в байтах **private_data_bytes**, могут быть скремблированными (зашифрованными). Однако, никакие другие поля секции **private_section** скремблироваться (шифроваться) не должны.

5.4.3 Все таблицы не должны подвергаться изменениям до окончания процесса декодирования в приемнике. Таблицы не должны быть скремблированы (зашифрованы) при передаче, за исключением таблицы EIT, для которой допускается скремблирование (шифрование).

5.4.4 При использовании скремблирования (шифрования) пакетов транспортного потока допускается заполнение промежутка от конца секции до конца пакета неинформационными битами заполнения так, чтобы любые переходы между скремблированными (шифрованными) и нескремблированными (нешифрованными) данными происходили на границах пакета.

5.4.5 В системах, где предусматривается случайный доступ, необходимо несколько раз заново передавать секции SI даже в тех случаях, когда изменений конфигурации не происходит.

5.4.6 Минимальный временной интервал между появлением последнего байта текущей секции и первым байтом следующей переданной секции с теми же самыми идентификаторами PID, **table_id** и **table_id_extension** и с тем же самым или отличающимся номером секции **section_number** должен быть равен 25 мс. Это ограничение должно применяться для транспортных потоков с полной скоростью передачи данных до 100 Мбит/с.

5.5 Отображение секций в пакеты транспортного потока

5.5.1 Между секциями и пакетами транспортного потока должно быть установлено четкое соответствие. Секции могут начинаться в начале полезной нагрузки пакета транспортного потока. Начало первой секции в полезной нагрузке транспортного пакета должно быть указано в поле **pointer_field**. В транспортном пакете не должно быть более одного поля **pointer_field**. Начало следующей секции должно быть идентифицировано подсчетом длины предыдущей. Используемый синтаксис не должен допускать никаких промежутков между секциями в пределах пакета транспортного потока.

5.5.2 В пределах пакетов транспортного потока с любым единственным значением PID одна секция должна закончиться прежде, чем может быть разрешен старт следующей. Если секция должна закончиться ранее конца транспортного пакета, но начинать другую секцию неудобно, то для заполнения свободного пространства в пределах поля полезной нагрузки допускается использовать механизм балластных вставок символов.

5.5.3 Процесс заполнения свободного пространства должен производиться путем записи в каждый свободный байт транспортного пакета значения “0xFF”. Значение байта “0xFF” не должно использоваться в табличном идентификаторе **table_id**. Если байт, следующий непосредственно за последним байтом секции, имеет значение “0xFF”, то вся оставшаяся часть транспортного пакета должна быть заполнена байтами “0xFF”. Байты заполнения могут игнорироваться декодером. Заполнение байтами может быть также выполнено в пределах поля адаптации пакета **adaptation_field**.

5.6 Кодирование полей идентификаторов PID и table_id

5.6.1 Значения идентификатора пакета PID, которые должны использоваться для опознавания пакетов транспортного потока, переносящих секции сервисной информации SI, приведены в таблице 2.

5.6.2 Значения идентификатора секций **table_id**, которые должны использоваться в таблицах сервисной информации, приведены в таблице 3.

Таблица 2

Таблица	Значение идентификатора PID
PAT	0x0000
CAT	0x0001
TSDT	0x0002
Зарезервировано	0x0003 – 0x000F
NIT, ST	0x0010
SDT, BAT, ST	0x0011
EIT, ST, CIT	0x0012
RST, ST	0x0013
TDT, TOT, ST	0x0014
Синхронизация сети	0x0015
RNT	0x0016
Зарезервировано на будущее	0x0017 – 0x001B
Внутриполосная сигнализация	0x001C
Измерения	0x001D
DIT	0x001E
SIT	0x001F

Таблица 3

Значение	Описание
0x00	program_association_section
0x01	coriditional_access_section
0x02	program_map_section

Продолжение таблицы 3

Значение	Описание
0x03	TS_Description_section
0x04	ИСО_IEC_14496_scene_Description_section
0x05	ИСО_IEC_14496_object_descriptor_section
0x06	Metadata_section
0x07	IPMP_Control_Information_section
0x08 – 0x3F	Зарезервировано
0x40	network_information_section – текущая сеть
0x41	network_information_section – другая сеть
0x42	service_description_section – actual_transport_stream
0x43 – 0x45	Зарезервировано для использования в будущем
0x46	service_description _section – other_transport stream
0x47 – 0x49	Зарезервировано для использования в будущем
0x4A	bouquet_associaton_section
0x4B – 0x4D	Зарезервировано для использования в будущем
0x4E	event_information_section – actual_transport_stream, текущий/последующий
0x4F	event_information_section – other_transport_stream, текущий/последующий
0x50 – 0x5F	event_information_section – actual_transport_stream,schedule
0x60 – 0x6F	event_information_section – other_transport_stream,schedule
0x70	time_date_section
0x71	running_status_section
0x72	stuffing_section
0x73	time_offset_section
0x74	application information section
0x75	container section

Окончание таблицы 3

Значение	Описание
0x76	related content section
0x77	content identifier section
0x78	Секция MPE-FEC
0x79	resolution notification section
0x7A	Секция MPE-IFEC
0x7B – 0x7D	Зарезервировано для использования в будущем
0x7E	discontinuity_information_section
0x7F	selection_information_section
0x80 – 0xFE	user_defined
0xFF	Зарезервировано

5.7 Дескрипторы программ и элементов программ

5.7.1 Дескрипторы программ и элементов программ представляют собой структуры, которые следует использовать для расширения определений программ и элементов программ. Все дескрипторы должны иметь формат, который должен начинаться с поля метки **descriptor_tag**. После метки **descriptor_tag** должно следовать поле длины дескриптора **descriptor_length** и поле данных [3].

5.7.2 Семантические определения полей дескрипторов:

descriptor_tag: метка дескриптора – поле длиной 8 бит, которое идентифицирует каждый дескриптор.

descriptor_length: длина дескриптора – поле длиной 8 бит, определяющее число байтов дескриптора, который следует сразу после поля descriptor_length.

Значения меток дескриптора descriptor_tag и названия дескрипторов, определенные в [3] для программно-зависимой информации PSI, приведены в Приложении А.

5.7.3 Дескрипторы, определенные в [1] и применяемые в системах DVB, а также их местоположение в таблицах сервисной информации SI приведены в Приложении Б. Приведенное местоположение дескрипторов в конкретных таблицах SI не ограничивает их возможного применения в других таблицах SI [1].

6 Таблицы сервисной информации

6.1 Таблица взаимосвязи программ PAT

6.1.1 Таблица взаимосвязи программ PAT должна определять соответствие между меткой **program_number** и значением PID для пакетов транспортного потока. Метка **program_number** – это числовая метка, связанная с программой. Для каждой службы в мультиплексе таблица PAT должна указывать местонахождение соответствующей таблицы PMT (значения PID пакетов транспортного потока). Она также должна указывать местонахождение таблицы сетевой информации NIT.

6.1.2 Общая таблица может содержаться в одной или нескольких секциях с описанным далее синтаксисом. Она может быть сегментирована и может занимать несколько секций (см. таблицу 4).

Т а б л и ц а 4

Синтаксис	Число битов	Мнемокод
program_association_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
"0"	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i = 0; i < N; i++) {		
program_number	16	uimsbf
reserved	3	bslbf
if (program_number = = "0") {		
network_PID	13	uimsbf

Окончание таблицы 4

Синтаксис	Число битов	Мнемокод

}
else {
program_map_PID
}
}
CRC_32
}
	13
	
	32
		
		uimsbf
		
		rpchof

6.1.3 Каждая секция PAT должна передаваться не реже, чем каждые 100 мс [4].

6.1.4 Семантические определения полей в секции взаимосвязи программ PAT:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено равным "1".

reserved: *зарезервировано*.

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны иметь значение "00". Оставшиеся 10 битов определяют количество байтов в секции, начинающейся сразу же после поля section_length, включая CRC. Значение в этом поле не должно превышать величины 1021 (0x3FD).

transport_stream_id: *идентификатор транспортного потока* – поле длиной 16 бит, которое служит меткой, для того чтобы отличить этот транспортный поток от любого другого мультиплексированного потока в сети. Его значение определяется пользователем.

version_number: *номер версии* – поле длиной 5 бит представляет собой номер версии полной таблицы взаимосвязи программ. Номер версии должен увеличиваться на 1 (по модулю 32) всегда, когда происходят изменения в таблице взаимосвязи программ. Когда индикатор current_next_indicator установлен в "1", поле version_number должно соответствовать применяемой в текущий момент версии таблицы PAT. Когда индикатор current_next_indicator установлен в "0", поле version_number должно соответствовать следующей применяемой версии таблицы взаимосвязи программ.

current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, будучи установлено в "1", указывает, что в настоящий момент применяется переданная таблица PAT. Когда этот бит установлен в "0", он указывает, что переданная таблица уже неприменима, и что должна быть введена в действие следующая таблица.

section_number: *номер секции* – поле длиной 8 бит указывает номер этой секции. Поле section_number первой секции в таблицы взаимосвязи программ должно быть представлено в шестнадцатеричном формате. Оно должно увеличиваться на 1 с появлением каждой дополнительной секции в таблице взаимосвязи программ.

last_section_number: *номер последней секции* – поле длиной 8 бит указывает номер последней секции (т.е., секции с наибольшим номером section_number) полной таблицы взаимосвязи программ.

program_number: *номер программы* – поле длиной 16 бит. Он определяет программу, к которой применяется поле program_map_PID. Когда он установлен в 0x0000, то следующим опорным PID должен быть PID сети. Во всех других случаях значение этого поля определяется пользователем. Это поле не должно принимать никакого значения более чем один раз в пределах одной версии таблицы взаимосвязи программ. (Поле program_number может использоваться, например, для обозначения радиовещательного канала).

network_PID: *PID сети вещания* – это поле длиной 13 бит, которое используется только в объединении со значением поля program_number = 0x0000 и определяет значение PID пакетов транспортного потока, которые должны содержать таблицу сетевой информации. Присутствие идентификатора network_PID является необязательным.

program_map_PID – поле длиной 13 бит, определяющее значение PID пакетов транспортного потока, которые должны содержать секцию program_map_section, применимую программе, определенной полем program_number. Ни одно поле program_number не должно иметь более одного назначения program_map_PID.

CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции PAT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.2 Таблица структуры программы PMT

6.2.1 Таблица структуры программы PMT обеспечивает связь между номерами программ и составными элементами программы при помощи идентификаторов программ PID. Таблица структуры

программы PMT должна идентифицировать и указывать местоположение потоков, которые составляют каждую службу и местоположение полей эталонных меток времени программы конкретной службы. Таблица структуры программы PMT должна передаваться в пакетах, значения PID для которых выбираются кодером. Эта таблица содержится в одной или нескольких секциях с описанным ниже синтаксисом. Она может быть сегментирована и занимать несколько секций. Секции должны определяться полем номера программы **program_number**.

Синтаксис секции структуры программы PMT для транспортного потока приведен в таблице 5.

Таблица 5

Синтаксис	Число битов	Мнемокод
TS_program_map_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
"0"	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
program_number	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved	3	bslbf
PCR_PID	13	uimsbf
reserved	4	bslbf
program_info_length	12	uimsbf
for (i = 0; i < N; i++) {		
descriptor()		
}		
for (i = 0; i < N1; i++) {		
stream_type	8	uimsbf
reserved	3	bslbf
elementary_PID	13	uimsbf
reserved	4	bslbf
ES_info_length	12	uimsbf
for (i = 0; i < N2; i++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

6.2.2 Каждая секция PMT должна передаваться не реже, чем каждые 100 мс [4].

6.2.3 Семантические определения полей в секции структуры программы PMT:

table_id: *идентификатор таблицы* – поле длиной 8 бит, которое для секции TS_program_map_section должно быть всегда установлено в значение 0x02, как показано в таблице 3.

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено равным "1".

reserved: *зарезервировано*.

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны иметь значение "00". Оставшиеся 10 битов определяют количество байтов в секции, которая начинается сразу же после поля section_length, включая CRC. Значение в этом поле не должно превышать величины 1021 (0x3FD).

program_number: *номер программы* – поле длиной 16 бит. Оно определяет программу, к которой применим идентификатор program_map_PID. Одно определение программы должно передаваться только в одной секции TS_program_map_section. Это означает, что структура программы никогда не бывает длиннее, чем 1016 (0x3F8).

version_number: *номер версии* – поле длиной 5 бит представляет собой номер версии секции TS_program_map_section. Номер версии должен увеличиваться на 1 (по модулю 32), каждый раз, когда происходит изменение информации, передаваемой в этой секции. Номер версии ссылается на

определение отдельной программы, и, следовательно, отдельной секции. Когда индикатор current_next_indicator установлен в "1", поле version_number должно соответствовать применяемой в текущий момент версии секции TS_program_map_section. Когда индикатор current_next_indicator установлен в "0", поле version_number должно соответствовать следующей применяемой версии секции TS_program_map_section.

current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, когда установлено в "1", указывает, что в настоящий момент применяется переданная секция TS_program_map_section. Когда этот бит установлен в "0", он указывает, что переданная секция TS_program_map_section уже неприменима, и что должна быть введена в действие следующая секция TS_program_map_section.

section_number: *номер секции* – поле длиной 8 бит, значение которого должно быть представлено в шестнадцатеричном виде.

last_section_number: *номер последней секции* – значение поля длиной 8 бит должно быть представлено в шестнадцатеричном виде.

PCR_PID – поле длиной 13 бит, указывающее значение PID пакетов транспортного потока, которые должны содержать поля PCR, действующие для программы, определенной полем program_number. Если для потоков конфиденциальной информации ни одно значение PCR не связано с определением программы, то поле PCR_PID должно принимать значение 0x1FFF.

program_info_length – поле длиной 12 бит, первые два бита которого должны иметь значение "00". Оставшиеся 10 битов определяют количество байтов в дескрипторах, которые следуют сразу же после поля program_info_length.

stream_type: *тип потока* – поле длиной 8 бит, определяющее тип элемента программы, передаваемого в пакетах с PID, значение, которое определяется полем elementary_PID. Значения поля stream_type определяются в Приложении Г.

elementary_PID: *PID элементарного потока* – поле длиной 13 бит, определяющее значение PID пакетов транспортного потока, в которых передается элемент соответствующей программы.

ES_info_length – поле длиной 12 бит, первые два бита которого должны иметь значение "00". Оставшиеся 10 битов определяют количество байтов описателей элемента соответствующей программы, который следует сразу же после поля **ES_info_length**.

CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции PMT транспортного потока результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.3 Таблица условного доступа CAT

6.3.1 Таблица условного доступа CAT должна описывать связь между одной или несколькими системами условного доступа, используемыми в мультиплексе, их потоки EMM и любые конкретные параметры, связанные с ними. Информация таблицы CAT – конфиденциальная (в данном стандарте она не может быть определена) и зависящая от конкретной используемой системы условного доступа, но она должна содержать местоположение потока EMM, когда таковой существует.

6.3.2 Таблица CAT может состоять из одной или нескольких секций с синтаксисом, приведенным в таблице 6. Она может быть сегментирована и занимать несколько секций.

Таблица 6

Синтаксис	Число битов	Мнемокод
CA_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
"0"	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
reserved	18	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	32	rpchof
}		

6.3.3 Семантические определения полей в секции условного доступа:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено равным "1".

reserved: *зарезервировано*.

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны иметь значение "00". Оставшиеся 10 битов определяют количество байтов в секции, которая начинается сразу же после поля section_length, включая поле CRC. Значение в этом поле не должно превышать величины 1021 (0x3FD).

version_number: *номер версии* – поле длиной 5 бит представляет собой номер версии полной таблицы условного доступа. Номер версии должен увеличиваться на 1 (по модулю 32) каждый раз, когда происходит изменение информации, передаваемой в таблице CAT. Когда индикатор current_next_indicator установлен в "1", поле version_number должно соответствовать применяемой в текущий момент версии таблицы условного доступа. Когда индикатор current_next_indicator установлен в "0", поле version_number должно соответствовать следующей применяемой версии таблицы условного доступа.

current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, будучи установлено в "1" указывает, что в настоящий момент применяется переданная таблица условного доступа. Когда этот бит установлен в "0", он указывает, что переданная таблица – следующая таблица условного доступа.

section_number: *номер секции* – поле длиной 8 бит указывает номер этой секции. Поле section_number первой секции в таблице условного доступа должно иметь значение 0x00. Оно должно увеличиваться на 1 с введением каждой дополнительной секции в таблицу условного доступа.

last_section_number: *номер последней секции* – поле длиной 8 бит указывает номер последней секции (т.е. секции с наибольшим номером section_number) полной таблицы условного доступа.

CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции условного доступа CAT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.4 Таблица описания транспортного потока TSDT

6.4.1 Таблица описания транспортного потока TSDT является дополнительной. Если она присутствует, то описание транспортного потока должно передаваться в пакетах транспортного потока, которые имеют значение PID = 0x0002, и должно применяться ко всему транспортному потоку целиком. Таблица описания транспортного потока TSDT должна обеспечивать передачу дескрипторов для всего транспортного потока в целом. В таблице TSDT должно использоваться значение table_id = 0x03, определенное в таблице 4, и оно должно передаваться в пакетах транспортного потока, имеющих значение PID = 0x0002, как определено в таблице 2.

Синтаксис таблицы описания транспортного потока TSDT представлен в таблице 7.

Таблица 7

Синтаксис	Число битов	Мнемокод
TS_Description_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
"0"	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
reserved	18	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i = 0; i < N; i++) {		
descriptor()		
}		
CRC_32	32	rpchof
}		

6.4.2 Семантические определения полей в секции транспортного потока TSDT:

table_id – *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть

установлено равным "1".
reserved: *зарезервировано*.
section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны иметь значение "00". Оставшиеся 10 битов определяют количество байтов в секции, которая начинается сразу же после поля section_length, включая CRC. Значение в этом поле не должно превышать величины 1021 (0x3FD).
version_number: *номер версии* – поле длиной 5 бит представляет собой номер версии всей таблицы описания транспортного потока в целом. Номер версии должен увеличиваться на 1 (по модулю 32) всегда, когда меняется определение таблицы описания транспортного потока. Когда индикатор current_next_indicator установлен в "1", то поле version_number должно соответствовать применяемой в текущий момент таблице описания транспортного потока. Когда индикатор current_next_indicator установлен в "0", поле version_number должно соответствовать следующей применяемой версии таблицы описания транспортного потока.
current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, будучи установлено в "1", указывает, что в настоящий момент применяется переданная таблица описания транспортного потока. Когда этот бит установлен в "0", он указывает, что переданная таблица уже неприменима, и что должна быть введена в действие следующая таблица.
section_number: *номер секции* – поле длиной 8 бит указывает номер секции. Поле section_number первой секции в таблице описания транспортного потока должно иметь значение 0x00. Оно должно увеличиваться на 1 с введением каждой дополнительной секции в таблицу описания транспортного потока.
last_section_number: *номер последней секции* – поле длиной 8 бит указывает номер последней секции (т.е., секции с наибольшим номером section_number) полной таблицы описания транспортного потока.
CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции описания транспортного потока TSDT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.5 Таблица сетевой информации NIT

6.5.1 Таблица сетевой информации NIT должна отражать информацию, относящуюся к физической организации мультиплексов и транспортных потоков, передаваемых по данной сети, а также характеристики сети. Комбинация идентификаторов исходной сети original_network_id и транспортного потока transport_stream_id должны позволять идентифицировать каждый транспортный поток единственным образом. Так как сетевые идентификаторы network_id служат уникальными кодами опознавания сетей, то за сетями должны быть закреплены их индивидуальные значения. Сетевые идентификаторы network_id и идентификаторы исходной сети original_network_id могут принимать одинаковые или различные значения в зависимости от накладываемых на них ограничений [1].

6.5.2 Таблица сетевой информации является дополнительной, и ее содержание должно быть конфиденциальным. Если она присутствует, то она должна передаваться в пакетах транспортного потока, которые имеют одинаковое значение PID, размещаемое в поле network_PID. Значение network_PID должно определяться пользователем и, если присутствует, то должно находиться в таблице ассоциаций для программы в зарезервированном значении program_number 0x0000. Если существует таблица сетевой информации, она должна передаваться в виде одной или нескольких частных секций private_section [3].

6.5.3 Приемник-декодер может иметь функцию сохранения информации NIT в энергонезависимой памяти, что позволяет минимизировать время доступа при переключении между каналами. Существует возможность передавать таблицы NIT для других сетей как дополнение таблицы для текущей сети. Установление различий между таблицами NIT для текущей сети и для других сетей должно обеспечиваться использованием различных значений табличного идентификатора table_id.

6.5.4 Таблица NIT должна быть сегментирована в секции сетевой информации network_information_section с использованием синтаксиса таблицы 8. Любые секции, составляющие часть таблицы NIT, должны передаваться в транспортных пакетах со значением идентификатора PID, равным 0x0010. Любые секции таблицы NIT, которые описывают фактическую сеть (т.е. сеть, частью которой является транспортный поток, содержащий NIT), должны иметь значение табличного идентификатора table id, равное 0x40, и одинаковые значения поля table_id_extension (network_id). Любые секции NIT, которые относятся к другой сети, а не к текущей, должны принимать значение table_id, равное 0x41, а идентификатор network_id должен иметь значение, выделяемое другой сетью.

Таблица 8

Синтаксис	Число битов	Мнемокод
network_informaion_section () {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
network_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
network_descriptors_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
for(j=0;j<N;j++){		
descriptor()		
}		
}		
crc_32	32	rpchof
}		

6.5.5 Каждая секция NIT должна передаваться не реже, чем каждые 10 с. Минимальный временной интервал между получением последнего байта секции и первым байтом следующей передаваемой секции, имеющей те же самые table_id и table_id_extension, должен быть не менее 25 мс [4].

6.5.6 Семантические определения полей в секции сетевой информации NIT:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено в 1.

reserved_future_use: *зарезервировано для использования в будущем.*

reserved: *зарезервировано.*

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны быть установлены в “00”. Оно определяет число байтов секции, начинающихся сразу после поля section length (включая код проверки CRC). Длина поля section length не должна превышать 1021 байт, чтобы секция целиком имела максимальную длину 1024 байта.

network_id: *идентификатор сети* – поле длиной 16 бит служит меткой, идентифицирующей систему доставки, о которой сообщается в таблице NIT, и позволяющей отличить ее от любой другой системы доставки.

version_number: *номер версии* – поле длиной 5 бит является номером версии субтаблицы. Номер version_number должен быть увеличен на 1 всякий раз, когда происходит изменение в информации, переносимой в рамках субтаблицы. Когда он достигает значения 31, то циклически сбрасывается в 0. Когда индикатор current_next_indicator установлен в 1, то номер version_number должен быть таким же, как у применяющейся в данный момент субтаблицы, определенной значениями идентификаторов table_id и network_id. Когда индикатор current_next_indicator установлен в 0, номер version_number должен быть таким же, как у следующей применимой субтаблицы, определенной значениями идентификаторов table_id и network_id.

current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, будучи

установлено в 1, показывает, что субтаблица является применимой в настоящее время, а будучи установлено в 0, – на то что посланная субтаблица еще не является применимой и должна быть получена следующая, которая будет введена в действие.

section_number: *номер секции* – поле длиной 8 бит, сообщающее номер секции. Номер первой секции в sub_table должен иметь значение "0x00". Номер section_number должен быть увеличен на 1 для каждой дополнительной секции с теми же самыми идентификаторами table_id и network_id.

last_section_number: *номер последней секции* – поле длиной 8 бит, определяющее номер последней секции (т.е. секции с самым высоким значением section_number) субтаблицы, частью которой является эта секция.

network_descriptors_length: *длина сетевых дескрипторов* – поле длиной 12 бит, задающее полную длину в байтах следующих сетевых дескрипторов.

transport_stream_loop_length – *длина циклов транспортного потока* – поле длиной 12 бит, сообщающее общую длину в байтах циклов транспортного потока, которые следуют за данным полем и заканчиваются перед первым байтом кода проверки CRC-32.

transport_stream_id: *идентификатор транспортного потока* – поле длиной 16 бит, которое служит меткой, идентифицирующей и позволяющей отличить данный транспортный поток от любого другого мультиплекса в системе доставки.

original_network_id: *идентификатор исходной сети* – поле длиной 16 бит, которое служит меткой, идентифицирующей network_id первоначальной системы доставки.

transport_descriptors_length: *длина транспортных дескрипторов* – поле длиной 12 бит, которое определяет в байтах *полную длину* следующих за ним *дескрипторов транспортного потока.*

CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции сетевой информации NIT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.6 Таблица управляющей информации IPMP

6.6.1 Таблица управляющей информации IPMP должна присутствовать в транспортном потоке, если какой-либо из компонентов транспортного потока использует механизм IPMP. Передача в транспортном потоке таблицы IPMP должна сигнализироваться идентификатором PID со значением 0x03.

6.6.2 Управляющая информация IPMP должна содержать следующие структурные элементы: список инструментов IPMP, граф управления IPMP (указывает, где в системе применен тот или иной инструмент), контейнер инструментов IPMP (программные модули необходимых инструментов) и контейнер прав IPMP (правила применения).

6.6.3 Таблица управляющей информации IPMP должна присутствовать, если какой-либо из компонентов в транспортном потоке использует сжатие по стандарту MPEG-4.

6.6.4 Так как размер таблицы управляющей информации IPMP может быть большим, то частота повторения этой таблицы должна быть подобрана таким образом, чтобы при декодировании в приемнике-декодере она отвечала требованиям к транспортному буферу системы.

6.6.5 Различным классам управляющей информации IPMP, переносимой в секциях с идентификатором table_id, равным 0x07, должны соответствовать теги class_tag, приведенные в таблице 9.

Таблица 9

Значение class_tag	Имя класса	Описание
0x00	Запрещено	–
0x01	Список инструментов IPMP	Инструменты IPMP, требуемые для защиты последовательности данных
0x02	Информация об инструментах IPMP	Описание одного из инструментов IPMP
0x03	Контейнер инструментов IPMP	Перенос двоичного содержания инструментов IPMP
0x04	Контейнер прав IPMP	Перенос прав, относящихся к контенту
0x05	Параметрическое описание IPMP	Параметрическое описание
0x06 – 0xC0	Зарезервировано	–
0xC1 – 0xFE	Конфиденциальная информация пользователя	–
0xFF	Запрещено	–

6.6.6 Синтаксис таблицы управляющей информации IPMP должен соответствовать представленному в таблице 10.

Таблица 10

Синтаксис	Число битов	Мнемокод
IPMP_Control_Info_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
"0"	1	bslbf
reserved	2	bslbf
section_length	20	uimsbf
reserved	2	bslbf
ipmp_control_info_version	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
descriptor_lenght	24	uimsbf
for (i = 0; i < N; i++) {		
descriptor()		
}		
isSigned	1	bslbf
reserved	7	bslbf
If (isSigned)		
Signature	8	ByteArrey
NumCerts	8	uimsbf
for (i = 0; i < numCerts; i++) {		
CertType	8	uimsbf
Certificate	8	ByteArrey
}		
Verifying_Tool_ID	128	uimsbf
}		
CRC_32	32	rpchof
}		

6.6.7 Семантические определения полей в секции управляющей информации IPMP:

table_id – *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено равным "1".

reserved: *зарезервировано*.

section_length: *длина секции* – поле длиной 20 бит, определяющее число байтов данной секции следующих за ним, включая байты поля CRC. Значение поля не должно превышать 1048573.

ipmp_control_info_version – поле длиной 5 бит описывает *номер версии полной таблицы управляющей информации IPMP*. Номер ipmp_control_info_version должен быть увеличен на 1 всякий раз, когда происходит изменение в информации, переносимой в рамках таблицы. Когда он достигает значения 31, то циклически сбрасывается в 0. Когда индикатор current_next_indicator установлен в 1, то номер версии должен быть таким же, как у применяющейся в данный момент таблицы управляющей информации IPMP. Когда индикатор current_next_indicator установлен в 0, номер версии должен быть таким же, как у следующей применимой таблицы управляющей информации IPMP.

current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, будучи установлено в 1, показывает, что таблица управляющей информации IPMP является применимой в настоящее время, а будучи установлено в 0, – на то что посланная таблица управляющей информации IPMP еще не применима и должна быть получена следующая, которая будет введена в действие.

section_number: *номер секции* – поле длиной 8 бит, сообщающее номер секции. Номер первой секции в таблице управляющей информации IPMP должен иметь значение “0x00”. Номер section_number должен быть увеличен на 1 для каждой дополнительной секции таблицы управляющей информации IPMP. Когда он достигает значения 256, то циклически сбрасывается в 0.

last_section_number: *номер последней секции* – поле длиной 8 бит, определяющее номер последней секции (т.е. секции с самым высоким значением section_number) таблицы управляющей информации IPMP, частью которой является эта секция.

descriptor_lenght: длина дескриптора – поле длиной 16 бит, определяющее общую длину поля дескрипторов, следующих сразу за данным полем.

isSigned: *подписано* – поле длиной 1 бит, которое отражает наличие подписи в таблице управляющей информации IPMP.
Signature: *подпись* – поле длиной 8 бит, которое содержит подпись для всей таблицы управляющей информации IPMP. Поле включает дескрипторы описывающие перечень применяемых механизмов и их контейнер.
CertType: *тип сертификации* – поле длиной 8 бит, которое описывает тип применяемого механизма сертификации.
NumCerts: *число сертификатов* – поле длиной 8 бит, которое описывает количество включенных сертификатов.
Certificate: *сертификат* – поле длиной 8 бит, содержащее список сертификатов.
Verifying_Tool_ID: *идентификатор проверки сертификатов* – поле длиной 128 бит, которое содержит идентификационные номера инструментов, необходимых для проверки сертификатов. В данном поле может, например, передаваться идентификационный номер терминала.
CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции IPMT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.7 Таблица взаимосвязи программных пакетов BAT

6.7.1 Таблица взаимосвязи программных пакетов BAT должна содержать информацию, относящуюся к группам служб. Под группой служб понимается совокупность служб, элементы которых могут передаваться в различных сетях.

6.7.2 Таблица BAT должна быть сегментирована в секции групп служб bouquet_association_sections с использованием синтаксиса, приведенного в таблице 11. Любые секции, составляющие часть таблицы BAT, должны передаваться в транспортных пакетах со значением идентификатора PID, равным 0x0011. Любые секции таблицы BAT должны иметь значение табличного идентификатора table_id, равное 0x4A.

Таблица 11

Синтаксис	Число битов	Мнемокод
bouquet_association_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
bouquet_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
bouquet_descriptors_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for(i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
for(j=0;j<N;j++){		
descriptor()		

Окончание таблицы 11

Синтаксис	Число битов	Мнемокод
} } CRC_32 }	32	rpchof

6.7.3 Семантические определения полей в секции взаимосвязи программных пакетов BAT:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено в 1.

reserved_future_use: *зарезервировано для использования в будущем*.

reserved: *зарезервировано*.

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны быть установлены в "00". Оно определяет число байтов секции, начинающихся сразу после поля section length (включая код проверки CRC). Длина поля section length не должна превышать 1021 байт, чтобы секция целиком имела максимальную длину 1024 байта.

bouquet_id: *идентификатор взаимосвязи программных пакетов* – поле длиной 16 бит служит меткой, идентифицирующей группу служб.

version_number: *номер версии* – поле длиной 5 бит является номером версии субтаблицы sub_table. Номер version_number должен быть увеличен на 1 всякий раз, когда происходит изменение в информации, переносимой в рамках sub_table. Когда он достигает значения 31, то циклически сбрасывается в 0. Когда индикатор current_next_indicator установлен в 1, то номер version_number должен быть таким же, как у применяющейся в данный момент субтаблицы, определенной значениями идентификаторов table_id и bouquet_id. Когда индикатор current_next_indicator установлен в 0, номер version_number должен быть таким же, как у следующей применимой субтаблицы sub_table, определенной значениями идентификаторов table_id и bouquet_id.

current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, будучи установлено в 1, показывает, что субтаблица является применимой в настоящее время, а будучи установлено в 0, – на то что посланная субтаблица еще не является применимой и должна быть получена следующая, которая будет введена в действие.

section_number: *номер секции* – поле длиной 8 бит, сообщающее номер секции. Номер первой секции в sub_table должен иметь значение "0x00". Номер section_number должен быть увеличен на 1 для каждой дополнительной секции с теми же самыми идентификаторами table_id и bouquet_id.

last_section_number: *номер последней секции* – поле длиной 8 бит, определяющее номер последней секции (т.е. секции с самым высоким значением section_number) субтаблицы, частью которой является эта секция.

bouquet_descriptors_length: *длина дескрипторов таблицы взаимосвязи программных пакетов* – поле длиной 12 бит, задающее полную длину в байтах следующих дескрипторов.

transport_stream_loop_length: *длина циклов транспортного потока* – поле длиной 12 бит, сообщающее общую длину в байтах циклов транспортного потока, которые следуют за данным полем и заканчиваются перед первым байтом кода проверки CRC-32.

transport_stream_id: *идентификатор транспортного потока* – поле длиной 16 бит, которое служит меткой, идентифицирующей и позволяющей отличить данный транспортный поток от любого другого мультиплекса в системе доставки.

original_network_id: *идентификатор исходной сети* – поле длиной 16 бит, которое служит меткой, идентифицирующей network_id первоначальной системы доставки.

transport_descriptors_length: *длина транспортных дескрипторов* – поле длиной 12 бит, которое определяет в байтах *полную длину* следующих за ним *дескрипторов транспортного потока.*

CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции взаимосвязи программных пакетов BAT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.8 Таблица описания услуги SDT

6.8.1 Таблица описания услуги SDT должна содержать перечень названий служб, провайдеров услуг и другие параметры, связанные с каждой службой в мультиплексе стандарта MPEG-2. Данные службы могут быть частью транспортного потока, в котором передается таблица SDT, или входить в состав других транспортных потоков, что определяется посредством значения идентификатора таблицы table_id.

6.8.2 Таблица SDT должна быть сегментирована в секции описания услуги service_description_sections с использованием синтаксиса, приведенного в таблице 12. Любые секции, составляющие часть таблицы SDT, должны передаваться в транспортных пакетах со значением идентификатора PID, равным 0x0011. Любые секции таблицы SDT, описывающей данный транспортный поток (т.е. содержащий таблицу SDT), должны иметь значение табличного идентификатора table_id, равное 0x42, и одинаковые значения полей table_id_extension (transport_stream_id) и original_network_id. Любые секции таблицы SDT, описывающей отличный от данного транспортный поток, должны иметь значение табличного идентификатора table_id, равное 0x46.

Таблица 12

Синтаксис	Число битов	Мнемокод
service_description_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
original_network_id	16	uimsbf
reserved_future_use	8	bslbf
for (i=0;i<N;i++){		
service_id	16	uimsbf
reserved_future_use	6	bslbf
EIT_schedule_flag	1	bslbf
EIT_present_following_flag	1	bslbf
running_status	3	uimsbf
free_CA_mode	1	bslbf
descriptors_loop_length	12	uimsbf
for (j=0;j<N;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

6.8.3 Семантические определения полей в секции описания услуги SDT:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено в 1.

reserved_future_use: *зарезервировано для использования в будущем*.

reserved: *зарезервировано*.

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны быть установлены в "00". Оно определяет число байтов секции, начинающихся сразу после поля section length (включая код проверки CRC). Длина поля section length не должна превышать 1021 байт, чтобы секция целиком имела максимальную длину 1024 байта.

transport_stream_id: *идентификатор транспортного потока* – поле длиной 16 бит, которое служит меткой, идентифицирующей и позволяющей отличить данный транспортный поток от любого другого мультиплекса в системе доставки.

version_number: *номер версии* – поле длиной 5 бит является номером версии субтаблицы. Номер version_number должен быть увеличен на 1 всякий раз, когда происходит изменение в информации, переносимой в рамках субтаблицы. Когда он достигает значения 31, то циклически сбрасывается в 0. Когда индикатор current_next_indicator установлен в 1, то номер version_number должен быть таким

же, как у применяющейся в данный момент субтаблицы sub_table. Когда индикатор current_next_indicator установлен в 0, номер version_number должен быть таким же, как у следующей применимой субтаблицы sub_table.

current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, будучи установлено в 1, показывает, что sub_table является применимой в настоящее время, а будучи установлено в 0, – на то что посланная sub_table еще не применима и должна быть получена следующая, которая будет введена в действие.

section_number: *номер секции* – поле длиной 8 бит, сообщающее номер секции. Номер первой секции в sub_table должен иметь значение "0x00". Номер section_number должен быть увеличен на 1 для каждой дополнительной секции с теми же самыми идентификаторами table_id, transport_stream_id, и original_network_id.

last_section_number: *номер последней секции* – поле длиной 8 бит, определяющее номер последней секции (т.е. секции с самым высоким значением section_number) субтаблицы, частью которой является эта секция.

original_network_id: *идентификатор исходной сети* – поле длиной 16 бит, которое служит меткой, идентифицирующей network_id первоначальной системы доставки.

service_id: *идентификатор услуги* - поле длиной 16 бит, которое служит ярлыком, позволяющим отличить текущую услугу от других услуг, передаваемых внутри ТП. Значение идентификатора службы service_id совпадает со значением номера программы program_number в соответствующей секции структуры программы program_map_section.

EIT_schedule_flag: *флаг расписания таблицы информации о программных элементах* – поле длиной 1 бит, которое будучи установлено в «1», показывает, что информация о программных элементах EIT для данной службы содержится в данном транспортном потоке. В случае если значение флага равно «0», информация о расписании для службы не должна передаваться в транспортном потоке.

EIT_present_following_flag – поле длиной 1 бит, которое будучи установлено в «1», показывает, что информация EIT_present_following для данной службы содержится в данном транспортном потоке. В случае если значение флага равно «0», информация EIT_present_following для службы не должна передаваться в транспортном потоке.

running_status: *статус выполнения* – поле длиной 3 бита, показывающее статус службы (см. таблицу 13).

Таблица 13

Значение	Описание
0	Не определено
1	Не выполняется
2	Начнется через несколько секунд (для видеозаписи)
3	Выполнение прервано
4	Выполняется
5	Служба отключена
6–7	Зарезервировано для будущего использования

free_CA_mode – поле длиной 1 бит, которое будучи установлено в «0», показывает, что все потоки, составляющие данную службу, не шифруются. Значение флага равное «1» показывает, что доступ к одному или нескольким потокам может быть ограничен системой условного доступа CA.

transport_descriptors_length: *длина транспортных дескрипторов* – поле длиной 12 бит, которое определяет в байтах полную длину следующих за ним дескрипторов.

CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции описания услуги SDT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.9 Таблица информации о программном элементе EIT

6.9.1 Таблица информации о программном элементе EIT должна предоставлять в хронологическом порядке сведения, описывающие программные элементы (события), содержащиеся в каждой услуге. Таблицы информации о программном элементе EIT должны быть

классифицированы по четырем типам, отличающимся различными значениями идентификатора table_id:

1) текущий транспортный поток, информация о текущем и последующем программных элементах, значение table_id равно "0x4E";
2) другой транспортный поток, информация о текущем и последующем программных элементах, значение table_id равно "0x4F";
3) текущий транспортный поток, информация о расписании программных элементов, значение table_id равно от "0x50" до "0x5F";
4) другой транспортный поток, информация о расписании программных элементов, значение table_id равно от "0x60" до "0x6F".

6.9.2 Любые составляющие таблиц EIT, относящиеся к данному транспортному потоку, должны иметь одинаковые значения полей transport_stream_id и original_network_id.

6.9.3 Таблица EIT, содержащая информацию о текущем и последующем программных элементах, должна включать в себя только данные о текущем событии и следующем за ним событии, которые передаются рассматриваемой службой в составе данного или другого транспортного потока, за исключением случая службы "почти видео по запросу" (NVOD). В этом случае может передаваться информация о более чем двух событиях. Таблицы расписания событий, относящиеся к данному или другому транспортному потоку, должны содержать список событий в виде расписания, которое включает в себя события за временными границами следующего события. Таблицы EIT не являются обязательными. Данные, касающиеся программных элементов, должны предоставляться в хронологическом порядке.

6.9.4 Таблица EIT должна быть сегментирована в секции *информации о программном элементе* event_information_sections с использованием синтаксиса, приведенного в таблице 14. Любые секции, составляющие часть таблицы EIT, должны передаваться в транспортных пакетах со значением идентификатора PID, равным 0x0012.

Таблица 14

Синтаксис	Число битов	Мнемокод
event_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
service_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
segment_last_section_number	8	uimsbf
last_table_id	8	uimsbf
for(i=0;i<N;i++){		
event_id	16	uimsbf
start_time	40	bslbf
duration	24	uimsbf
running_status	3	uimsbf
free_CA_mode	1	bslbf
descriptors_loop_length	12	uimsbf
for(i=0;i<N;i++){		

Окончание таблицы 14

Синтаксис	Число битов	Мнемокод
descriptor() } } CRC_32 }	32	rpchof

6.9.5 Семантические определения полей в секции информации о программном элементе EIT:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено в 1.

reserved_future_use: *зарезервировано для использования в будущем*.

reserved: *зарезервировано*.

section_length: *длина секции* – поле длиной 12 бит, которое определяет число байтов секции, начинающихся сразу после поля section_length (включая код проверки CRC). Длина поля section_length не должна превышать 4093 байт, чтобы секция целиком имела максимальную длину 4096 байта.

service_id: *идентификатор услуги* - поле длиной 16 бит, которое служит ярлыком, позволяющим отличить текущую услугу от других услуг, передаваемых внутри ТП. Значение идентификатора службы service_id совпадает со значением номера программы program_number в соответствующей секции структуры программы program_map_section.

version_number: *номер версии* – поле длиной 5 бит является номером версии субтаблицы sub_table. Номер version_number должен быть увеличен на 1 всякий раз, когда происходит изменение в информации, переносимой в рамках sub_table. Когда он достигает значения 31, то циклически сбрасывается в 0. Когда индикатор current_next_indicator установлен в 1, то номер version_number должен быть таким же, как у применяющейся в данный момент субтаблицы sub_table. Когда индикатор current_next_indicator установлен в 0, номер version_number должен быть таким же, как у следующей применимой субтаблицы sub_table.

current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, будучи установлено в 1, показывает, что sub_table является применимой в настоящее время, а будучи установлено в 0, – на то что посланная sub_table еще не применима и должна быть получена следующая, которая будет введена в действие.

section_number: *номер секции* – поле длиной 8 бит, сообщающее номер секции. Номер первой секции в sub_table должен иметь значение “0x00”. Номер section_number должен быть увеличен на 1 для каждой дополнительной секции с теми же самыми идентификаторами table_id, service_id, transport_stream_id и original_network_id. В этом случае субтаблица может быть представлена в виде совокупности сегментов. Внутри каждого сегмента значение поля номер секции должно быть увеличено на 1 для каждой дополнительной секции, но возможен скачок в нумерации между последней секцией сегмента и первой секцией следующего сегмента.

last_section_number: *номер последней секции* – поле длиной 8 бит, определяющее номер последней секции (т.е. секции с самым высоким значением section_number) субтаблицы, частью которой является эта секция.

transport_stream_id: *идентификатор транспортного потока* – поле длиной 16 бит, которое служит меткой, идентифицирующей и позволяющей отличить транспортный поток, который описывает таблицы EIT, от любого другого мультиплекса в системе доставки.

original_network_id: *идентификатор исходной сети* – поле длиной 16 бит, которое служит меткой, идентифицирующей network_id исходной системы доставки.

segment_last_section_number: *номер последней секции сегмента* – поле длиной 8 бит, которое определяет номер последней секции данного сегмента субтаблицы. Для несегментированных субтаблиц данное поле должно иметь тоже значение, что и поле last_section_number.

last_table_id: *идентификатор последней таблицы* – поле длиной 8 бит, которое определяет последнее используемое значение идентификатора таблицы.

event_id: *идентификатор события* – поле длиной 16 бит, содержащее идентификационный номер описываемого события (однозначно определенного в описании службы).

start_time: *время начала* – поле длиной 40 бит, содержащее время начала события в формате Всемирного координированного времени (UTC), модифицированной юлианской даты (MJD). Данное поле кодируется в виде 16 наименее значимых битов MJB, за которыми следуют 24 бита,

кодирующие 6 цифр четырехбитным двоично-десятичным кодом. Если время начала не определено (например, для сервиса NVOD) все биты должны иметь значение 1 (ПРИМЕР: 93/10/13 12:45:00 кодируется как "0xC079124500").

duration: *длительность* – поле длиной 24 бита, содержащее длительность события в часах, минутах, секундах в виде 6 чисел, кодированных четырехбитным двоично-десятичным кодом (ПРИМЕР: 01:45:30 кодируется как "0x014530").

running_status: *статус выполнения* – поле длиной 3 бита, показывающее статус службы (см. таблицу 12). Для сервиса NVOD значение должно быть равно 0.

free_CA_mode – поле длиной 1 бит, которое будучи установлено в «0», показывает, что все потоки, составляющие данную службу, не шифруются. Значение флага равное «1» показывает, что доступ к одному или нескольким потокам может быть ограничен системой условного доступа CA.

transport_descriptors_length: *длина транспортных дескрипторов* – поле длиной 12 бит, которое определяет в байтах полную длину следующих за ним дескрипторов.

CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции информации о программном элементе EIT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.10 Таблица времени и даты TDT

6.10.1 Таблица времени и даты TDT должна использоваться для передачи информации точного времени, включая текущее время и дату. Данные TDT должны передаваться в отдельной таблице из-за частого обновления этой информации. Таблица времени и даты должна содержать информацию о текущем времени и дате только в формате Всемирного координированного времени.

6.10.2 Таблица TDT должна состоять из одной секции с использованием синтаксиса таблицы 15. Секция таблицы TDT должна передаваться в транспортных пакетах с идентификатором PID равным 0x0014 и идентификатором таблицы table_id равным 0x70.

Таблица 15

Синтаксис	Число битов	Мнемокод
time_date_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
UTC_time	40	bslbf
}		

6.10.3 Семантические определения полей в секции времени и даты TDT:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено в 0.

reserved_future_use: *зарезервировано для использования в будущем.*

reserved: *зарезервировано.*

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны иметь значение 00. Данное поле определяет число байтов секции, начинающихся сразу после поля section_length

UTC_time: *Всемирное координированное время* – поле длиной 40 бит, содержащее текущее время и дату в форматах UTC и MJD (см. Приложение Д). Данное поле кодируется в виде 16 наименее значимых битов MJB, за которыми следуют 24 бита, кодирующих 6 цифр четырехбитным двоично-десятичным кодом.

6.11 Таблица сдвига по времени TOT

6.11.1 Таблица сдвига по времени TOT содержит информацию о времени в формате Всемирного координированного времени и дате, а также значение сдвига (смещения) местного времени от всемирного.

6.11.2 Таблица TDT должна состоять из одной секции с использованием синтаксиса таблицы 16. Секция таблицы TDT должна передаваться в транспортных пакетах с идентификатором PID равным 0x0014 и идентификатором таблицы table_id равным 0x73.

Таблица 16

Синтаксис	Число битов	Мнемокод
time_offset_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
UTC_time	40	bslbf
reserved	4	bslbf
descriptors_loop_length	12	uimsbf
for(i=0;i<N;i++){		
descriptor()		
}		
CRC_32	32	rpchof
}		

6.11.3 Семантические определения полей в секции сдвига по времени TOT:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено в 0.

reserved_future_use: *зарезервировано для использования в будущем.*

reserved: *зарезервировано.*

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны иметь значение 00. Данное поле определяет число байтов секции, начинающихся сразу после поля section_length

UTC_time: *Всемирное координированное время* – поле длиной 40 бит, содержащее текущее время и дату в форматах UTC и MJD (см. Приложение Д). Данное поле кодируется в виде 16 наименее значимых битов MJB, за которыми следуют 24 бита, кодирующих 6 цифр четырехбитным двоично-десятичным кодом.

descriptors_loop_length: *длина дескрипторного цикла* – поле длиной 12 бит, сообщающее общую длину в байтах следующих за ним дескрипторов.

CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции информации о сдвиге по времени TOT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

6.12 Таблица текущего статуса RST

6.12.1 Таблица текущего статуса RST должна обеспечивать точное и своевременное обновление статуса одного или нескольких программных элементов, что может быть необходимо в случае, если программный элемент начинается раньше или позже из-за изменения расписания. Использование данных таблиц должно обеспечивать механизм быстрого обновления данных.

6.12.2 Таблица RST должна быть сегментирована в секции текущего статуса running_status_sections с использованием синтаксиса таблицы 17. Любые секции, составляющие часть таблицы RST, должны передаваться в транспортных пакетах со значением идентификатора PID, равным 0x0013, и идентификатором таблицы table_id, равным 0x71.

Таблица 17

Синтаксис	Число битов	Мнемокод
running_status_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf

Окончание таблицы 17

Синтаксис	Число битов	Мнемокод
section_length	12	uimsbf
for (i=0;i<N;i++){		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
service_id	16	uimsbf
event_id	16	uimsbf
reserved_future_use	5	bslbf
running_status	3	uimsbf
}		
}		

6.12.3 Семантические определения полей в секции текущего статуса RST:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено в 0.

reserved_future_use: *зарезервировано для использования в будущем*.

reserved: *зарезервировано*.

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны иметь значение 00. Данное поле определяет число байтов секции, начинающихся сразу после поля section_length. Длина поля section_length не должна превышать 1021 байт, чтобы секция целиком имела максимальную длину 1024 байта.

transport_stream_id: *идентификатор транспортного потока* – поле длиной 16 бит, которое служит меткой, идентифицирующей и позволяющей отличить данный транспортный поток от любого другого мультиплекса в системе доставки.

original_network_id: *идентификатор исходной сети* – поле длиной 16 бит, которое служит меткой, идентифицирующей network_id первоначальной системы доставки.

service_id: *идентификатор услуги* - поле длиной 16 бит, которое служит ярлыком, позволяющим отличить текущую услугу от других услуг, передаваемых внутри ТП. Значение идентификатора службы service_id совпадает со значением номера программы program_number в соответствующей секции структуры программы program_map_section.

event_id: *идентификатор события* – поле длиной 16 бит, содержащее идентификационный номер описываемого события.

running_status: *статус выполнения* – поле длиной 3 бита, описывающее статус службы.

6.13 Таблица байтов согласования скорости ST

6.13.1 Таблица байтов согласования скорости ST должна использоваться с целью отмены действия существующих секций на граничных пунктах системы доставки, например, на головной станции кабельной сети. Если одна из секций таблицы ST перезаписана, то тогда все секции этой таблицы также должны быть перезаписаны с целью сохранения целостности поля секции section_number.

6.13.2 Синтаксис таблицы неоднородности информации ST должен соответствовать представленному в таблице 18.

Таблица 18

Синтаксис	Число битов	Мнемокод
stuffing_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	Bslbf
reserved	2	bslbf
section_length	12	uimsbf
for (i=0;i<N;i++){		

Окончание таблицы 18

Синтаксис	Число битов	Мнемокод
data_byte } }	8	uimsbf

6.13.3 Семантические определения полей в секции согласования скорости ST:
table_id: *идентификатор таблицы* (см. таблицу 3).
section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое может принимать как значение 0, так и 1.
reserved_future_use: *зарезервировано для использования в будущем.*
reserved: *зарезервировано.*
section_length: *длина секции* – поле длиной 12 бит, определяющее число байтов секции, начинающихся сразу после поля section_length поле. Длина поля section_length не должна превышать 4093 байт, чтобы секция целиком имела максимальную длину 4096 байта.
data_byte: *байт данных* – поле длиной 8 бит, которое может принимать любое значение и не несет никакой смысловой нагрузки.

6.14 Таблица неоднородности информации DIT

6.14.1 Таблица неоднородности информации DIT должна использоваться только в частичных транспортных потоках. Не допускается передавать таблицу DIT в полных стандартных транспортных потоках MPEG-2, предназначенных для вещания.

6.14.2 Таблицу DIT следует вводить в переходных точках транспортного потока, когда сервисная информация SI может иметь временные разрывы или иные неоднородности.

6.14.3 Синтаксис таблицы неоднородности информации DIT должен соответствовать представленному в таблице 19.

Т а б л и ц а 19

Синтаксис	Число битов	Мнемокод
discontinuity_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transition_flag	1	uimsbf
reserved_future_use	7	bslbf
}		

6.14.4 Семантические определения полей в секции неоднородности информации DIT:
table_id: *идентификатор таблицы* (см. таблицу 3).
section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено в 0.
reserved_future_use: *зарезервировано для использования в будущем.*
reserved: *зарезервировано.*
section_length: *длина секции* – поле длиной 12 бит, установленное в 0x001.
transition_flag: *флаг перехода* - поле для флага, указывающего на тип перехода в транспортном потоке. Флаг, установленный в "1", означает, что переход связан с изменением первоначального источника. Изменения первоначального источника могут быть связаны с изменением, как самого транспортного потока, так и с положением в транспортном потоке (в случае смещения во времени). Флаг, установленный в "0" означает, что переход связан только с выбором, при сохранении изначального транспортного потока и положения в нем.

6.15 Таблица выбираемой информации SIT

6.15.1 Таблица выбираемой информации SIT должна содержать информацию, описывающую услуги и программные элементы, переносимые в частичных транспортных потоках, которые могут быть "прерывающимися", т.е. содержать временные разрывы. Таблица SIT должна включать в себя итоговую подборку всей сервисной информации вещательного транспортного потока. Не допускается передавать таблицу SIT в полных стандартных транспортных потоках MPEG-2, предназначенных для вещания.

6.15.2 Если таблица SIT присутствует в потоке данных, этот поток должен идентифицироваться как частичный транспортный поток, полученный от цифрового интерфейса. В этом случае приемник-декодер не должен производить поиск сервисной информации, связанной с вещанием, и должен опираться исключительно на информацию, переносимую в таблице SIT.

6.15.3 Синтаксис таблицы выбираемой информации SIT должен соответствовать представленному в таблице 20.

Таблица 20

Синтаксис	Число битов	Мнемокод
selection_information_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
DVB_reserved_future_use	1	bslbf
ISO_reserved	2	bslbf
section_length	12	uimsbf
DVB_reserved_future_use	16	uimsbf
ISO_reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
DVB_reserved_for_future_use	4	uimsbf
transmission_info_loop_length	12	bslbf
for(i =0;i<N;i++) {		
descriptor()		
}		
for(i=0;i<N;i++){		
service_id	16	uimsbf
DVB_reserved_future_use	1	uimsbf
running_status	3	bslbf
service_loop_length	12	bslbf
for(j=0;j<N;j++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

6.15.4 Семантические определения полей в секции выбираемой информации SIT:

table_id: *идентификатор таблицы* (см. таблицу 3).

section_syntax_indicator: *индикатор синтаксиса секции* – поле длиной 1 бит, которое должно быть установлено в 0.

DVB_reserved_future_use: *зарезервировано DVB для использования в будущем.*

ISO_reserved: *зарезервировано ИСО.*

section_length: *длина секции* – поле длиной 12 бит, первые два бита которого должны иметь значение 00. Данное поле определяет число байтов секции, начинающихся сразу после поля section_length. Длина поля section_length не должна превышать 4093 байт, чтобы секция целиком имела максимальную длину 4096 байта.

version_number: *номер версии* – поле длиной 5 бит представляет собой номер версии всей таблицы выбираемой информации. Номер версии должен увеличиваться на 1 (по модулю 32) всегда, когда меняется определение таблицы выбираемой информации. Когда индикатор current_next_indicator установлен в "1", то поле version_number должно соответствовать применяемой в текущий момент таблице SIT. Когда индикатор current_next_indicator установлен в "0", поле version_number должно соответствовать следующей применяемой версии таблицы SIT.

current_next_indicator: *индикатор текущая/следующая* – поле длиной 1 бит, которое, будучи установлено в "1", указывает, что в настоящий момент применяется переданная таблица выбираемой информации. Когда этот бит установлен в "0", он указывает, что переданная таблица уже неприменима, и что должна быть введена в действие следующая таблица.

section_number: *номер секции* – поле длиной 8 бит предоставляет информацию о номере секции таблицы. Значение секции section_number должно быть представлено в шестнадцатеричной системе.

last_section_number: *номер последней секции* – поле длиной 8 бит определяет номер последней секции. Значение секции last_section_number должно быть представлено в шестнадцатеричной системе.
transmission_info_loop_length: *длина передаваемого дескрипторного цикла* - поле длиной 12 бит, сообщающее общую длину в байтах следующего за ним дескрипторного цикла, определяющего параметры передачи частичного ТП.
service_id: *идентификатор услуги* - поле длиной 16 бит, которое служит ярлыком, позволяющим отличить текущую услугу от других услуг, передаваемых внутри ТП. Поле service_id – то же самое, что и program_number в соответствующей секции program_map_section.
running_status: *статус выполнения* – поле длиной 3 бита, показывающее статус текущей службы. Если нет текущей службы, статус потока рассматривается как “не текущий”.
service_loop_length: *длина дескрипторного цикла услуги* – поле длиной 12 бит, сообщающее общую длину в байтах следующего за ним дескрипторного цикла, содержащего сервисную информацию, относящуюся к услуге и программным элементам предаваемого частичного транспортного потока.
CRC_32 – поле длиной 32 бита, которое содержит значение кода циклической проверки на основе избыточности CRC. При отсутствии ошибок в данных после обработки полной секции выбираемой информации SIT результатом проверки являются "нули" на выходе регистров в декодере, определенном в Приложении В.

Приложение А
(обязательное)

Дескрипторы программ и элементов программ

В таблице А.1 приведены значения меток дескриптора descriptor_tag и названия дескрипторов, определенные в [3].

Таблица А.1

descriptor_tag DEC / HEX	Название дескриптора
0 / 0x00	Зарезервировано
1 / 0x01	Зарезервировано
2 / 0x02	video_stream_descriptor
3 / 0x03	audio_stream_descriptor
4 / 0x04	hierarchy_descriptor
5 / 0x05	registration_descriptor
6 / 0x06	data_stream_alignment_descriptor
7 / 0x07	target_background_grid_descriptor
8 / 0x08	video_window_descriptor
9 / 0x09	CA_descriptor
10 / 0x0A	ISO_639_language_descriptor
11 / 0x0B	system_clock_descriptor
12 / 0x0C	multiplex_buffer_utilization_descriptor
13 / 0x0B	copyright_descriptor
14 / 0x0E	maximum_bitrate_descriptor
15 / 0x0F	private_data_indicator_descriptor
16 / 0x10	smoothing_buffer_descriptor
17 / 0x11	STD_descriptor
18 / 0x12	IBP_descriptor
0x13 – 0x1A	Дополнительно определено
27 / 0x1B	MPEG-4_video_descriptor
28 / 0x1C	MPEG-4_audio_descriptor
29 / 0x1D	IOD_descriptor
30 / 0x1E	SL_descriptor
31 / 0x1F	FMC_descriptor
32 / 0x20	external_ES_ID_descriptor
33 / 0x21	MuxCode_descriptor
34 / 0x22	FmxBufferSize_descriptor
35 / 0x23	multiplexbuffer_descriptor
36 / 0x24	content_labeling_descriptor
37 / 0x25	metadata_pointer_descriptor
38 / 0x26	metadata_descriptor
39 / 0x27	metadata_STD_descriptor
40 / 0x28	AVC video descriptor
41 / 0x29	IPMP_descriptor (defined in ISO/IEC 13818–11, MPEG-2 IPMP)
42 / 0x2A	AVC timing and HRD descriptor
43 / 0x2B	MPEG-2_AAC_audio_descriptor
44 / 0x2C	FlexMuxTiming_descriptor
0x2D – 0x3F	Зарезервировано

Приложение Б
(обязательное)

Определение дескрипторов и их местоположение

В таблице Б.1 перечислены дескрипторы, применяемые в системах DVB и их местоположение в таблицах сервисной информации SI. Приведенное местоположение дескрипторов в конкретных таблицах SI не ограничивает их возможного применения в других таблицах SI [1].

Таблица Б.1

descriptor_tag DEC / HEX	Название дескриптора	Местоположение в таблицах
64 / 0x40	network_name_descriptor	NIT
65 / 0x41	service_list_descriptor	NIT, BAT
66 / 0x42	stuffing_descriptor	NIT, BAT, SDT, EIT, SIT
67 / 0x43	satellite_delvery_system_descriptor	NIT
68 / 0x44	cable_delivery_system_descriptor	NIT
69 / 0x45	VBI_data_descriptor	PMT
70 / 0x46	VBI_teletext_descriptor	PMT
71 / 0x47	bouquet_name_descriptor	BAT, SDT, SIT
72 / 0x48	service_descriptor	SDT, SIT
73 / 0x49	country_availability_descriptor	BAT, SDT< SIT
74 / 0x4A	linkage_descriptor	NIT, BAT, SDT, EIT, SIT
75 / 0x4B	NVOD_reference_descriptor	SDT, SIT
76 / 0x4C	time_shifted_service_descriptor	SDT, SIT
77 / 0x4D	short_event_descriptor	EIT, SIT
78 / 0x4E	extended_event_descriptor	EIT, SIT
79 / 0x4F	time_shifted_event_descriptor	EIT, SIT
80 / 0x50	component_descriptor	EIT, SIT
81 / 0x51	mosaic_descriptor	SDT, PMT
82 / 0x52	stream_identifier_descriptor	PMT
83 / 0x53	CA_identifier_descriptor	BAT, SDT, EIT, SIT
84 / 0x54	content_descriptor	EIT, SIT
85 / 0x55	parental_rating_descriptor	EIT, SIT
86 / 0x56	teletext_descriptor	PMT
87 / 0x57	telephone_descriptor	PMT
88 / 0x58	local_time_offset_descriptor	TOT
89 / 0x59	subtitling_descriptor	NIT
90 / 0x5A	terrestrial_delivery_system_descriptor	NIT
91 / 0x5B	multilingual_network_name_descriptor	NIT
92 / 0x5C	multilingual_bouquet_name_descriptor	BAT
93 / 0x5D	multilingual_service_name_descriptor	SDT, SIT
94 / 0x5E	multilingual_component_descriptor	EIT, SIT
95 / 0x5F	private_data_specifier_descriptor	NIT, BAT, SDT, EIT, PMT, SIT
96 / 0x60	service_mode_descriptor	PMT
97 / 0x61	short_smoothing_buffer_descriptor	EIT, SIT
98 / 0x62	frequency_list_descriptor	NIT
99 / 0x63	partial_tranport_stream_descriptor	SIT
100 / 0x64	data_broadcast_descriptor	PMT
101 / 0x65	CA_system_descriptor	PMT, EIT
102 / 0x66	data_broadcast_id_descriptor	PMT
103 / 0x67	transport_stream_descriptor	
104 / 0x68	DSNG_descriptor	
105 / 0x69	PDC_descriptor	EIT
106 / 0x6A	AC-3_descriptor	PMT
107 / 0x6B	ancilliary_data_descriptor	PMT
108 / 0x6C	cell_list_descriptor	NIT

Продолжение таблицы Б1

descriptor_tag DEC / HEX	Название дескриптора	Местоположение в таблицах
109 / 0x6D	cell_frequency_link_descriptor	NIT
110 / 0x6E	announcement_support_descriptor	NIT
111 / 0x6F	application_signalling_descriptor	PMT
112 / 0x70	adaptation_field_data_descriptor	PMT
113 / 0x71	service_identifier_descriptor	SDT
114 / 0x72	service_availability_descriptor	SDT
115 / 0x73	default_authority_descriptor	NIT, BAS, SDT
116 / 0x74	related_content_descriptor	PMT
117 / 0x75	TVA_id_descriptor	EIT
118 / 0x76	content_identifier_descriptor	EIT
119 / 0x77	time_slice_fec_identifier_descriptor	NIT
120 / 0x78	ECM_repetition_rate_descriptor	PMT
121 / 0x79	S2_satellite_delivery_system_descriptor	NIT
122 / 0x7A	enhanced_AC-3_descriptor	PMT
123 / 0x7B	DTS descriptor	PMT
124 / 0x7C	AAC descriptor	PMT
125 / 0x7D	XAIT location descriptor	NIT,BAT,SDT,EIT,TOT,PMT,SIT
126 / 0x7E	FTA_content_management_descriptor	NIT,BAT,SDT,EIT
127 / 0x7F	extension descriptor	NIT,BAT,SDT,EIT,TOT,PMT,SIT
0x80 – 0xFE	Определяется пользователем	

Приложение В
(обязательное)

Модель декодера CRC

Модель 32-битового декодера кода циклической проверки на четность CRC-32 должна соответствовать структурной схеме, приведенной на рисунке В.1 [1].

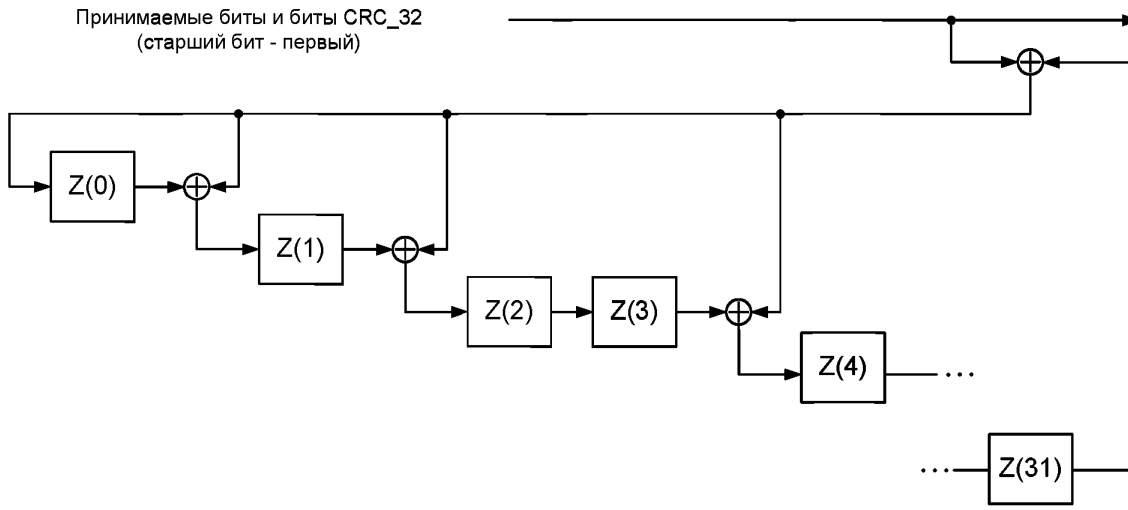


Рисунок В.1 – 32-битовая модель декодера CRC

Схема декодера кода CRC-32 должна работать следующим образом. Декодер CRC-32 обрабатывает данные по битам. Схема декодера CRC-32 состоит из 14 сумматоров по модулю 2 (обозначены знаком "+") и 32 элементов задержки z(i). Входной сигнал декодера CRC-32 суммируется с выходным сигналом элемента z(31), и результат суммирования подается на вход элемента z(0) и на один из входов каждого из остальных 13 сумматоров. Другой вход каждого оставшегося сумматора является выходом элемента z(i), тогда как выход каждого оставшегося сумматора соединяется с входом элемента z(i + 1), где i = 0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22 и 25.

Код циклической проверки CRC-32 должен вычисляться как результат деления входного блока данных на порождающий многочлен:

$$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1 \qquad (B.1)$$

Каждый байт данных на входе декодера CRC-32 вводится в схему по одному биту в каждом такте, при этом первым следует самый левый бит (старший). Например, если входным сигналом является байт стартового кода 0x01, то сначала в декодер CRC должны поступать семь "0", и после них одна "1". Перед вычислением контрольной суммы CRC-32 данных поступающей секции все элементы задержки z(i) должны быть инициализированы путем установки их в исходное значение "1". После инициализации декодера CRC-32 на его вход последовательно поступают все байты секции, включая четыре байта проверки четности CRC_32. После сдвига последнего бита последнего байта CRC_32 внутрь декодера, т.е. внутрь элемента z(0), и после сложения его с выходным сигналом элемента z(31) необходимо считать выходной сигнал со всех элементов задержки. В том случае, если ошибок нет, каждый из выходных сигналов z(i) должен быть равен нулю. В кодере CRC-32 данные поля CRC_32 кодируются таким образом, чтобы это условие гарантированно выполнялось.

Приложение Г
(обязательное)

Значения поля stream_type

Значения поля stream_type должны соответствовать приведенным в таблице Г.1 [3].

Т а б л и ц а Г.1

Значение	Описание
0x00	Зарезервировано в МСЭ-Т | ИСО/МЭК
0x01	Видеопоток в соответствии с ИСО/МЭК 11172
0x02	Видеопоток в соответствии с МСЭ-Т Н.262 | ИСО/МЭК 13818-2
0x03	Аудиопоток в соответствии с ИСО/МЭК 11172-3
0x04	Аудиопоток в соответствии с ИСО/ МЭК 13818-3
0x05	Конфиденциальные секции в соответствии с [3]
0x06	Пакеты PES, содержащие конфиденциальную информацию в соответствии с [3]
0x07	MHEG в соответствии с ИСО/МЭК 13522
0x08	DSM-CC в соответствии с [3]
0x09	В соответствии с Рекомендацией МСЭ-Т Н.222.1
0x0A	Расширение для DSM-CC тип А
0x0B	Расширение для DSM-CC тип В
0x0C	Расширение для DSM-CC тип С
0x0D	Расширение для DSM-CC тип D
0x0E	Дополнительные данные в соответствии с [3]
0x0F	Аудиопоток с синтаксисом транспортного потока ADTS
0x10	Визуальная информация в соответствии с ИСО/МЭК 14496-2
0x11	Аудио с синтаксисом транспорта LATM в соответствии с ИСО/МЭК 14496-3
0x12	Пакетированный поток SL в соответствии с ИСО/МЭК 14496-1 или поток FlexMux, передаваемый в пакетах PES
0x13	Пакетированный поток SL в соответствии с ИСО/МЭК 14496-1 или поток FlexMux, передаваемый в секциях ИСО/МЭК 14496_sections
0x14	Протокол синхронизированной загрузки в соответствии с ИСО/МЭК 13818-6
0x15	Метаданные, передаваемые в пакетах PES
0x16	Метаданные, передаваемые в секциях metadata_sections
0x17	Метаданные, передаваемые в данных карусельного типа в соответствии с ИСО/МЭК 13818-6
0x18	Метаданные, передаваемые в объекте карусельного типа в соответствии с ИСО/МЭК 13818-6
0x19	Метаданные, передаваемые в протоколе синхронизированной загрузки в соответствии с ИСО/МЭК 13818-6
0x1A	Поток IPMP
0x1B	Видеопоток AVC, определенный в Рекомендации МСЭ-Т Н.264 | ИСО/МЭК 14496-10
0x1C–0x7E	Зарезервирован в соответствии с [3]
0x7F	Поток 0IPMP m
0x80–0xFF	Конфиденциальный поток пользователя

Приложение Д
(обязательное)

Преобразования между временем и датой

Алгоритм преобразования между двумя отметками времени, условно обозначенными как "модифицированная юлианская дата + всемирное координированное время (MJD + UTC)" и ""местная" MJD + местное время"", должен быть сведен к добавлению или вычитанию местного сдвига по времени.

Смещения времени положительны для долгот к востоку от Гринвича и отрицательны для долгот к западу от Гринвича.

Возможные варианты преобразования времени показаны на рисунке Д.1 [1].

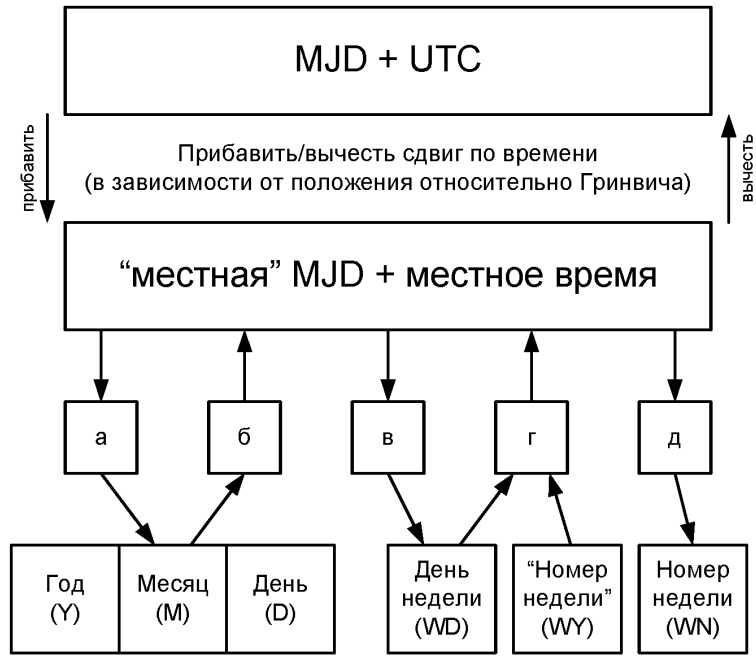


Рисунок Д.1 – Направления преобразований времени между модифицированным юлианским календарем и Всемирным координированным временем

Пять возможных направлений преобразования времени, показанных на рисунке Д.1, могут быть выражены с помощью следующих математических выражений:

а) Найти Y, M, D из MJD

$Y' = int [(MJD – 15\,078{,}2) / 365{,}25]$

$M' = int \{ [MJD – 14\,956{,}1 – int (Y' \times 365{,}25)] / 30{,}6001 \}$

$D = MJD – 14\,956 – int (Y' \times 365{,}25) – int (M' \times 30{,}6001)$

If M' = 14 or M' = 15, then K = 1;

else K = 0

$Y = Y' + K$

$M = M' – 1 – K \times 12$

б) Найти MJD из Y, M, D

If M = 1 or M = 2, then L = 1;

else L = 0

$MJD = 14\,956 + D + int [(Y – L) \times 365{,}25] + int [(M + 1 + L \times 12) \times 30{,}6001]$

в) Найти WD из MJD

$$WD = [(MJD + 2) \bmod 7] + 1$$

г) Найти MJD из WY, WN, WD

$$MJD = 15\,012 + WD + 7 \times \{ WN + int [(WY \times 1\,461 / 28) + 0{,}41] \}$$

д) Найти WY, WN из MJD

$$W = int [(MJD / 7) - 2\,144{,}64]$$

$$WY = int [(W \times 28 / 1\,461) - 0{,}0079]$$

$$WN = W - int [(WY \times 1\,461 / 28) + 0{,}41],$$

где:

D	– день месяца (от 1 до 31);
int	– целая часть (не учитывая остаток);
K, L, M', W, Y'	– промежуточные переменные;
M	– месяцы от января (M = 1) до декабря (M = 12);
MJD	– модифицированная юлианская дата (календарь);
mod 7	– остаток (от 1 до 6) после деления по модулю 7;
UTC	– Всемирное координированное время;
WD	– день недели с понедельника (WD = 1) по воскресенье (WD = 7);
WN	– номер недели в соответствии с ISO 8601;
WY	– "номер недели", начиная с 1900 года;
x	– знак умножения;
Y	– год, начиная с 1900 (пример: y=113 для 2013 года).

Библиография

[1]	Европейский стандарт связи EN 300 468 (European Standard (Telecommunications series) ETSI EN 300 468 V1.12.1 (2011–10))	Цифровое телевизионное вещание. Спецификация для сервисной информации (SI) в системах DVB. (Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems)
[2]	Международный стандарт ITU-T Recommendation J.94	Сервисная информация для цифрового вещания в кабельных телевизионных системах. (Service information for digital broadcasting in cable television systems)
[3]	Международный стандарт ITU-T Recommendation H.222.0 (2000) | ISO/IEC 13818–1:2000 *	Информационные технологии. Родовое кодирование киноизображений и сопровождающей звуковой информации. Системы. Часть 1. (Information technology – Generic coding of moving pictures and associated audio information: Systems - Part 1)
[4]	Рекомендация МСЭ-R BT.1300–3 (08/2005)	Методы уплотнения, транспортировки и идентификации сигналов изображения, звукового сопровождения и данных для наземного цифрового телевизионного радиовещания
[5]	Многоязычный словарь Европейского вещательного союза. Часть 3. EBU Multilingual Glossary of terms and phrases used in broadcasting. Part 3 (Dec. 1998). – EBU Tech. 3274	Применение цифровой техники в телерадиовещании. (Digital applications in broadcasting)

УДК 621.397:006.354 ОКС 33.170

Ключевые слова: сервисная информация, таблица сервисной информации, специальная информация программы, транспортный пакет, дескриптор, идентификатор пакета данных

Подписано в печать 01.08.2014. Формат 60x84 1/8.
Усл. печ. л.4,65. Тираж 34 экз. Зак. 1440.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ФГУП «СТАНДАРТИНФОРМ»

123995 Москва, Гранатный пер., 4.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru