

ICS 35.040
L71



中华人民共和国国家标准

GB/T 20090.1 — YYYY

信息技术 先进音视频编码 第 1 部分：系统

Information technology - Advanced coding of
audio and video - Part 1: System
(报批稿)

××××-××-××发布

××××-××-××实施

国家质量监督检验检疫总局 发布

目 次

目 次.....	I
前 言.....	VII
引 言.....	VIII
0.1 传输流	IX
0.2 节目流	X
0.3 传输流与节目流的转换	XI
0.4 分组包的基本流	XI
0.5 定时模型	XII
0.6 条件存取	XII
0.7 复用流操作	XII
0.8 单个流操作 (PES 分组包层)	XII
0.9 系统参考解码器	XIII
0.10 应用	XIII
1 范围.....	1
2 规范性引用文件.....	1
3 术语和定义.....	2
3.1 比特率 bitrate.....	2
3.2 字节对齐 byte aligned.....	2
3.3 信道 channel.....	2
3.4 编码表示 coded representation.....	2
3.5 压缩 compression.....	2
3.6 固定比特率 constant bitrate.....	2
3.7 受限系统参数流 CSPS (constrained system parameter stream) [系统]	2
3.8 循环冗余码校验 CRC(cyclic redundancy check).....	2
3.9 数据元素 data element.....	2
3.10 解码流 decoded stream.....	2
3.11 解码器 decoder.....	2
3.12 解码(过程) decoding (process)	3
3.13 解码时间戳 DTS (decoding time-stamp) [系统].....	3
3.14 数字存储介质 DSM (digital storage media)	3
3.15 数字存储介质的命令和控制 DSM-CC (digital storage medium command and control)	3
3.16 授权控制信息 ECM (entitlement control message)	3
3.17 授权管理信息 EMM (entitlement management message)	3
3.18 编辑 editing.....	3
3.19 基本流 ES (elementary stream) [系统]	3
3.20 基本流参考时钟 ESCR (elementary stream clock reference)[系统].....	3
3.21 编码器 encoder.....	3
3.22 编码(过程) encoding (process)	3
3.23 熵编码 entropy coding.....	3
3.24 事件 event.....	3
3.25 快速正向回放 fast forward playback [视频]	3

3.26 禁止 forbidden	4
3.27 (复用)流 (multiplexed) stream [系统]	4
3.28 层 layer [视频和系统]	4
3.29 分组 pack [系统]	4
3.30 包数据 packet data [系统]	4
3.31 节目成分标识符 PEID (program element identifier) [系统]	4
3.32 节目成分信息表 PEIT (program element information table) [系统]	4
3.33 补充 padding [音频]	4
3.34 有效负载数据 payload	4
3.35 分组基本流 PES (packet element stream) [系统]	4
3.36 PES 分组包 PES packet [系统]	4
3.37 PES 分组包头 PES packet header [系统]	4
3.38 PES 流 PES stream [系统]	5
3.39 展现时间戳 PTS (Presentation time-stamp) [系统]	5
3.40 展现单元 PU (presentation unit) [系统]	5
3.41 节目 program [系统]	5
3.42 节目参考时钟 PCR (program clock reference) [系统]	5
3.43 节目元素 program element [系统]	5
3.44 节目特定信息 PSI (program specific information) [系统]	5
3.45 随机访问 random access	5
3.46 相对显示时间 relative_display_time	5
3.47 保留 reserved	5
3.48 加扰 scrambling [系统]	5
3.49 源流 source stream	5
3.50 拼接 splicing [系统]	5
3.51 起始码 start codes [系统]	6
3.52 STD 输入缓冲区 STD input buffer [系统]	6
3.53 静态图像 still picture	6
3.54 系统头 system header [系统]	6
3.55 系统参考时钟 SCR (system clock reference) [系统]	6
3.56 系统目标解码器 STD (system target decoder) [系统]	6
3.57 时间戳 time-stamp [系统]	6
3.58 传输率 transport rate	6
3.59 传输流包头 transport stream packet header [系统]	6
3.60 可变比特率 variable bitrate	6
4 符号	6
4.1 算术运算符	6
4.2 逻辑运算符	7
4.3 关系运算符	7
4.4 按位运算符	7
4.5 赋值	7
4.6 助记符	8
4.7 常量	8
5 比特流语法的描述方式	8
6 传输流比特流要求	9

6.1 传输流编码结构与参数	9
6.2 传输流系统目标解码器	9
6.3 传输流语法语义规范	15
6.4 节目成分信息表	43
6.5 节目特定信息	46
7 节目流比特流的要求	58
7.1 节目流编码结构与参数	58
7.2 节目流系统目标解码器	58
7.3 节目流语法语义规定	61
7.4 节目流映射	67
7.5 节目流目录	69
8 节目和节目元素描述子	73
8.1 节目和节目元素描述子中各字段的语义定义	73
8.2 视频流描述子	74
8.3 视频流描述子中各字段的语义定义	75
8.4 音频流描述子	77
8.5 音频流描述子中各字段的语义定义	77
8.6 层次描述子	77
8.7 层次描述子中各字段的语义定义	79
8.8 注册描述子	79
8.9 注册描述子中各字段的语义定义	80
8.10 数据流对齐描述子	80
8.11 数据流对齐描述子中各字段的语义定义	80
8.12 目标背景栅格描述子	81
8.13 目标背景栅格描述子中各字段的语义	81
8.14 视频窗描述子	82
8.15 视频窗描述子中各字段的语义定义	82
8.16 条件访问描述子	83
8.17 条件访问描述子中字段的语义定义	83
8.18 ISO 639 语言描述子	83
8.19 ISO 639 语言描述子中各字段的语义定义	85
8.20 系统时钟描述子	85
8.21 系统时钟描述子中字段的语义定义	87
8.22 复用缓冲区应用描述子	87
8.23 复用缓冲区应用描述子中各字段的语义定义	87
8.24 版权描述子	88
8.25 版权描述子中各字段的语义定义	88
8.26 最大比特率描述子	88
8.27 最大比特率描述子中字段的语义定义	88
8.28 私有数据指示符描述子	89
8.29 私有数据指示符描述子中各字段的语义定义	89
8.30 平滑缓冲区描述子	89
8.31 平滑缓冲区描述子中各字段的语义定义	91
8.32 STD 描述子	91
8.33 STD 描述子中各字段的语义定义	91

8.34 IBP 描述子	91
8.35 IBP 描述子中各字段的语义定义	92
9 对多路复用流语义的约束	92
9.1 系统参考时钟的编码频率	92
9.2 节目参考时钟的编码频率	92
9.3 基本流参考时钟的编码频率	92
9.4 展现时间戳的编码频率	93
9.5 时间戳的条件编码	93
9.6 PES 分组包头中 P-STD_buffer_size 的编码频率	93
9.7 节目流中系统头编码	93
9.8 受限系统参数节目流	93
9.9 传输流中采样率的锁定	94
10 与 GB/T 17975 的兼容性	94
附 录 A (规范性附录) CRC 解码器模型	95
附 录 B (资料性附录) 数字存储介质命令与控制(DSM-CC)	96
B.1 引言	96
B.1.1 目标	96
B.1.2 未来的应用	96
B.1.3 优点	96
B.1.4 基本功能	96
B.2 通用元素	97
B.2.1 范围	97
B.2.2 DSM-CC 应用概述	97
B.2.3 DSM-CC 命令和响应的传输	98
B.3 技术原理	99
B.3.1 术语和定义	99
B.3.2 DSM-CC 语法规则	99
B.3.3 DSM-CC 语法规则中各字段的语义	101
B.3.4 控制层	101
B.3.5 控制层中各字段的语义	104
B.3.6 响应层	105
B.3.7 响应层中各字段的语义	105
B.3.8 时间编码	106
B.3.9 时间编码中各字段的语义	107
附 录 C (资料性附录) 节目特定信息	108
C.1 传输流中节目特定信息的说明	108
C.2 引言	108
C.3 功能机制	108
C.4 段到传输流包的映射	109
C.5 重复率和随机访问	109
C.6 节目是什么	109
C.7 program_number 的分配	110
C.8 典型系统中 PSI 的使用	110
C.9 PSI 结构的关系	110
C.9.1 节目关联表	111

C.9.2 节目映射表	112
C.9.3 条件访问表	112
C.9.4 网络信息表	112
C.9.5 私有段 private_section()	112
C.10 带宽利用和信号采集时间	114
附 录 D (资料性附录) 本部分的系统时间信息时钟恢复模型及应用指南	117
引言	117
D.1 时钟恢复模型	117
D.2 音频和视频展现同步	118
D.3 解码器中的系统时钟恢复	119
D.4 transport_rate 抖动	121
D.5 网络抖动情形下的时钟恢复	122
D.6 用于色度副载波生成的系统时钟	122
D.7 分量视频与音频的重构	123
D.8 帧滑动	123
D.9 网络抖动的平滑处理	124
附 录 E (资料性附录) 数据传输应用	125
E.1 一般性考虑	125
E.2 建议	125
附 录 F (资料性附录) 本部分语法的图形表示	126
引言	126
F.1 传输流语法	126
F.2 PES 分组包	127
F.3 节目相关段	127
F.4 CA 段	127
F.5 传输流节目映射段	128
F.6 私有段	128
F.7 节目流	129
F.8 节目流映射	129
附 录 G (资料性附录) 通用信息	130
G.1 同步字节冲突	130
G.2 跳过的图像状态和解码处理	130
G.3 PEID 值的选择	130
G.4 PES start_code 冲突	130
附 录 H (资料性附录) 私有数据	131
附 录 I (资料性附录) 系统一致性和实时接口	132
附 录 J (资料性附录) 引入抖动的网络与 AVS 解码器的接口	133
J.1 引言	133
J.2 网络一致模型	133
J.3 平滑抖动的网络规范	134
J.4 解码器实现举例	135
J.4.1 跟有 AVS 解码器的网络适配器	135
J.4.2 集成解码器	135
附 录 K (资料性附录) 拼接传输流	136
K.1 引言	136

GB/T 20090.1 — YYYY	
K.2 拼接点的不同类型。	136
K.2.1 普通拼接点	136
K.2.2 无缝拼接点	136
K.3 拼接中解码器的行为	137
K.3.1 非无缝拼接	137
K.3.2 无缝拼接	137
附 录 L (资料性附录) 系统解复用过程	139
附 录 M (资料性附录) 系统复用过程	142

前 言

GB/T 20090 在《信息技术 先进音视频编码》的总标题下，拟包括以下九个部分：

- 第 1 部分：系统；
- 第 2 部分：视频；
- 第 3 部分：音频；
- 第 4 部分：一致性测试；
- 第 5 部分：参考软件；
- 第 6 部分：数字媒体版权管理；
- 第 7 部分：移动视频；
- 第 8 部分：在 IP 网络上传输 AVS；
- 第 9 部分：AVS 文件格式。

本部分为 GB/T 20090 的第 1 部分。

本部分的附录 A 为规范性附录，附录 B~附录 K 为资料性附录。

本部分由中华人民共和国信息产业部提出。

本部分由全国信息技术标准化技术委员会归口。

本部分由中国科学院计算技术研究所、清华大学、北京算通科技发展有限公司、华中科技大学、北京邮电大学起草。

本部分起草人员为：高 文、黄铁军、杨士强、高鹏飞、丁亚强、杨志杰、林福辉、朱光喜、林永兵、贺玉文、瑜 莉、王雷、冯 博、张桢睿、杨昭辉、杨 洁、汪恒晶。

引 言

GB/T 20090 的本部分论述了如何将一路或多路音频、视频流或其它基本数据流合成单路或多路复用流，以适应于存储和传送。系统编码遵循本部分指定的语法和语义规则，并提供了使解码器缓冲区能在一个宽范围的补偿和接收条件下进行同步解码的信息。

系统编码可有两种形式：传输流和节目流，每一种针对不同的应用集合加以优化，本部分中定义的传输流和节目流提供了编码语法，该语法对于同步解码及展现音频、视频信息是必要的也是充分的，同时保证了解码器中数据缓冲区不发生上溢和下溢。在该语法中，利用有关编码和视频数据解码和演示的时间戳以及有关数据自身传输的时间戳对信息进行编码。传输流和节目流都是面向分组包的多路复用流。

单一音频和视频基本流的多路复用过程参见图 1。视频和音频数据按 GB/T 20090.2 和 GB/T 20090.3 编码。压缩数据被打包以形成 PES 分组包。在形成 PES 分组包的过程中可能会加入独立使用传输流或节目流的 PES 分组包所需的信息。当 PES 分组包进一步与系统层信息结合形成传输流或节目流时，这一信息是不必要的也是不能加入的。本部分覆盖了竖直虚线右边所示的处理过程。

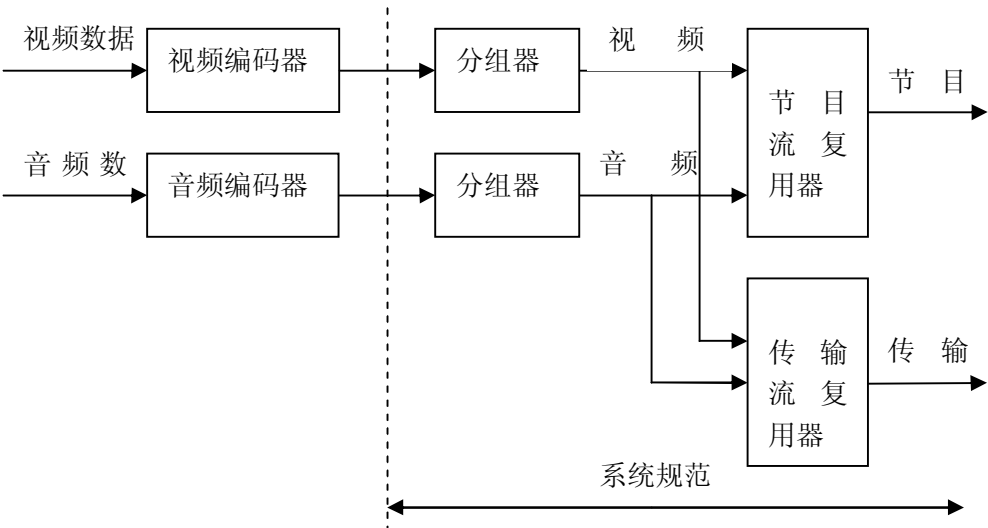


图 1 本部分范围简图

节目流与 GB/T 17191 系统层类似或相近。它是由具有共同时间基准的一个或多个 PES 分组包合并而成的单一流。

有些应用中要求包含单个节目的基本流是未多路复用的分离流。对这些应用，基本流也可作为分离的节目流编码。每一基本流含一个节目流且具有共同时间基准。在这种情况下，不同流中 SCR 字段的编码值必须一致。

和单一节目流一样，所有的基本流都可被同步解码。

节目流被设计为用于相对无差错环境中，且适用于牵涉到诸如交互式多媒体应用等系统信息软件处理的应用。节目流分组包长度可变，而且可能很长。

传输流将具有一个或多个不同时钟基准的一个或多个节目组成一个单一流。由属于同一个节目的多个基本流组成的多个 PES 分组包共享一个时钟基准。传输流为用于可能出现差错的环境设计的，例如在有损或有噪媒体中的存储或传输。传输流分组包长度为 188 字节。

节目流和传输流是为不同应用设计的，它们的定义并不严格遵守分层模型。彼此之间的转换是可能合

理的，但并不互为子集或超集。特别是，从一个传输流中抽取一个节目的内容并创建一个有效节目是可能的。该工作利用 PES 分组包的公共互换格式完成，但并非节目流需要的所有字段都包含在传输流中，有一些必须导出。而在分层模型中，传输流可能横跨多个层，且被设计为在宽带应用中高效和易于实现。

系统规范中陈述的语法和语义规则的范围是不同的：语法规则仅用于系统层编码，并不延伸到音频、视频规范的压缩层编码，而语义规则适用于复用流。

本系统规范并未规定编码器或解码器的体系结构或实现方法，也未对多路复用器或解复用器作相应的规定。然而，比特流的性质对编码器、解码器、多路复用器和解复用器提出了功能和性能上的要求，例如，编码器必须满足最小的时钟容差要求等。尽管有这样或那样的要求，编码器、解码器、多路复用器和解复用器的设计与实现仍然有相当大的自由度。

0.1 传输流

传输流作为一种流，是针对在那些可能会出现显著错误(往往表现为位差错或丢失分组包)的环境中进行节目传送和存储而定义的。这些节目包含按照 GB/T 20090.2 和 GB/T 20090.3 编码的数据以及其它数据。

传输流的速率可以是恒定的也可以是可变的。在任何情况下，所包含的基本流也是速率恒定或可变的。在每一种情况下，流的语法或语义限制是相同的，传输流速率由传输率(transport_rate)字段的值定义，这些 transport_rate 字段通常分离在每个流中。

构造和传送包含多个具有独立时基的节目的传输流，以使得总体比特流是可变的，这会存在一些困难。见 6.2.2。

传输流可以以任何方式构造，只要能生成一个有效的流。一个包含一个或多个节目的传输流可以从基本编码数据流、节目流或其它可能包含一个或多个节目的传输流构造得出。

传输流是按照在最小开销的情况下能对传输流执行某些操作的原则而设计的。这些操作包括：

从传输流的一个节目中获得编码数据，解码并展现，如图 2 所示。

从传输流的一个节目中抽取传输流分组包并生成一个仅包含该节目的不同的传输流作为输出，如图 3 所示。

从一个或多个传输流中抽取一个或多个节目的传输流分组包并生成一个不同的传输流。

从传输流中抽取一个节目内容并生成包含该节目的一个节目流，如图 4 所示。

把一个节目流转化为传输流，并在有损环境中传输。然后再行重建一个有效的、在某些情况下完全相同的节目流。

图 2 和图 3 描述了以一个传输流作为输入的解复用及解码系统原型。图 2 说明了第一种情况，即一个传输流被直接解复用和解码。传输流构造分为两层：系统层和压缩层。

传输流解码的输入流在压缩层外有一个系统层包围着。音频解码器及视频解码器的输入流只含一个压缩层。

接收传输流的解码器原型的操作既适用于整个传输流(“复用流操作”)，也适用于单个基本流(“特定流操作”)。传输流系统层被分为两个子层，一个用于复用流操作(传输流分组包层)，另一个用于特定流操作(PES 分组包层)。

图 2 也给出了一个包括视频和音频的传输流解码器原型以说明解码器的功能，其结构并不是唯一的。有些系统解码器功能，例如解码器时间控制，可能被相等地分配到基本流解码器或通道特定解码器中，但该图有助于讨论。类似的，通道特定解码器测出的错误也可以用多种途径通知独立的音频和视频解码器。这些通信途径并未显示在图中，该解码器原型的设计并不意味着对传输流解码器的设计作出任何标准化的要求。实际上，非音频/视频数据也是允许的，但并未在图 2 中画出。

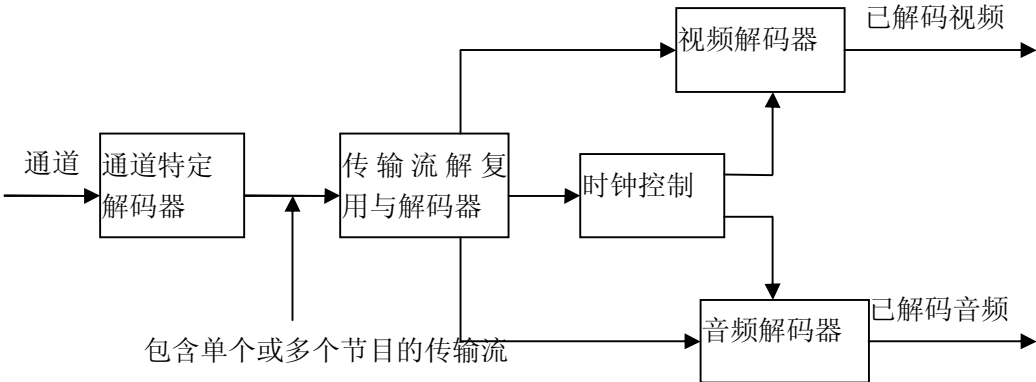


图 2 传输流解复用和解码原型示例

图 3 说明了另一种情况，即一个包含多个节目的传输流被转变为一个只含单个节目的传输流，这种情况下的再复用操作可能需要纠正 `transport_rate` 值以补偿比特流中 `transport_rate` 位置的变化。

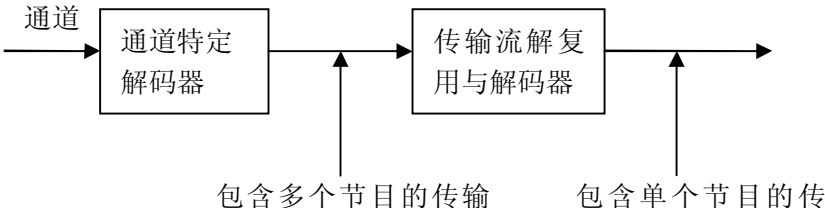


图 3 传输多路复用原型示例

图 4 说明了一个多节目传输流先被解复用再转变为节目流的情况。

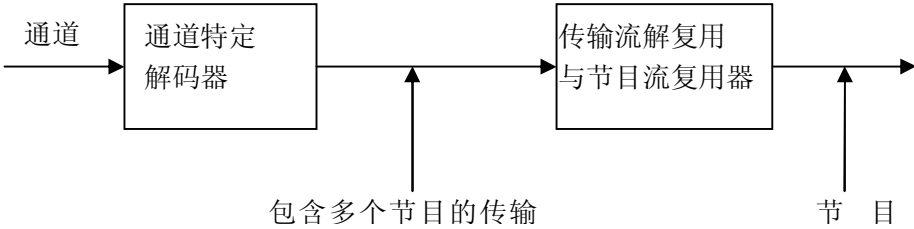


图 4 传输流到节目流转换原型示例

图 3 和图 4 指出，不同类型和构造的传输流之间的转换是可能的和合理的。在传输流和节目流的语法中都定义了一些特定字段以方便上述转换过程，但并不要求解复用器或解码器的具体实现要包含以上所有功能。

0.2 节目流

节目流作为一种流，是针对在那些出错率很低且系统编码的处理过程作为主要考虑因素的环境中进行一个节目的传送和存储而定义的，该节目包含编码数据和其它数据。

节目流的速率可以是恒定也可以是可变的。在任何情况下，所包含的基本流的速率也是恒定或可变的。在每一种情况下，流的语法或语义限制是相同的。节目流速率是由系统时钟参考(SCR)字段与 `mux-rate` 字段的值 and 位置所定义的。

图 5 描述了一个音频/视频节目流解码系统原型。其结构并不是唯一的——包括解码器时间控制在内的系统解码器功能可能被相等地分配到基本流解码器或通道特定解码器中，但该图有助于讨论。该解码器原型的设计并不意味着对节目流解码器的设计作出任何标准化的要求。实际上，非音频/视频数据也是允许的，但在图中并未画出。

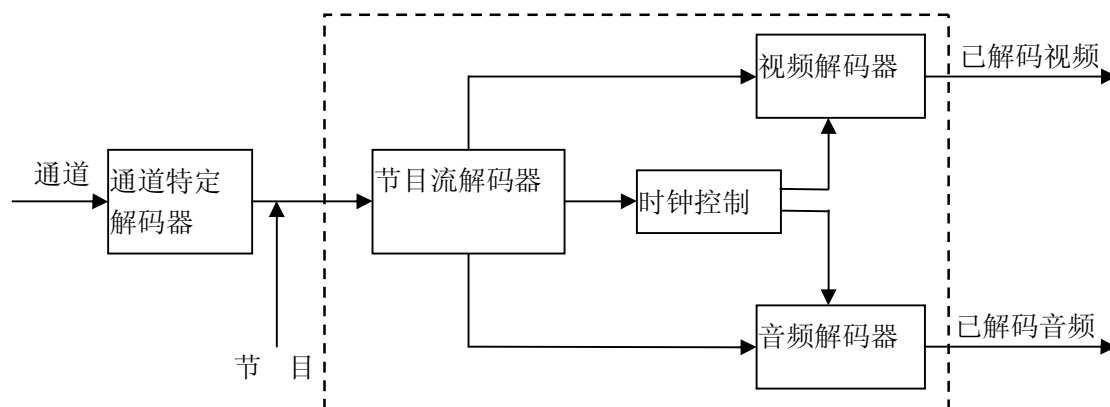


图 5 节目流解码器原型

图 5 所示的节目流解码器原型是由系统、视频和音频解码器三部分组成的，它们分别符合本标准的第 1 部分、第 2 部分和第 3 部分。在该解码器中，单路或多路音频/视频流的复用编码表示假定以某种特定通道格式在特定通道中存储或传输。特定通道格式不由本部分决定，特定通道解码也不是本解码器原型的一部分。

原型解码器接受节目流作为输入并依靠节目流解码器从流中提取信息。节目流解码器分离复用流，由此产生的基本流作为音频和视频解码器的输入。音频和视频解码器的输出是已解码的音频和视频信号。节目流解码器、音频和视频解码器以及通道特定解码器之间的定时信息流应包含在设计中，但并未在图中画出。利用定时信息，音频和视频解码器相互之间以及与通道之间可以实现同步。

节目流构造分为两层：系统层和压缩层。节目流解码器的输入流在压缩层外有一个系统层包围着，音频解码器及视频解码器的输入流只含一个压缩层。

解码器原型的操作既适用于整个节目流(“复用流操作”)，也适用于单个基本流(“特定流操作”)。节目流系统层被分为两个子层，一个用于复用流操作(包层)，另一个用于特定流操作(PES 分组包层)

0.3 传输流与节目流的转换

利用 PES 分组包，传输流与节目流之间的转换是可能的也是合理的，这是由包含在本部分的 6.1 和 7.1 中的传输流和节目流规范得出的。在某些限制下，PES 分组包可能直接从一个复用比特流的有效数据映射到另一个复用比特流的有效数据。如果在所有 PES 分组包中都有 `program_packet_sequence_counter`，就可能标识出 PES 分组包的正确次序以帮助实现这一功能。

在这两种流的表和标题中，均可得到转换所必需的其它特定信息，例如基本流之间的关系。这些数据，如果有的话，在任何流中转换前后都应是正确的。

0.4 分组包的基本流

正如 6.3.6 中的语法定义所指出的那样，传输流和节目流是从 PES 分组包中逻辑地建立的。PES 分组包被用于传输流与节目流之间的转换。在有些场合进行这种转换时，无需变动 PES 分组包。PES 分组包的尺寸可能比传输流分组包的尺寸大得多。

具有一个流 ID 的一个基本流的一系列连续 PES 分组包可用于构造 PES 流。当 PES 流分组包用于形成 PES 流时，应当在 6.3.8 中所定义的限制下，带有基本流时钟参考(ESCR)字段和基本流速率(ES_Rate)字段。PES 流数据应是来自基本流且保持原次序的连续字节。PES 流中不包含一些包含在节目流和传输流中必须的系统信息。例如，包含在包头、系统头、节目流映像、节目流目录、节目映射表中的信息以及在传输流分组包语法中的元素。

PES 流是一个在本部分的实现中可能有用的逻辑结构，但它并不被定义为一个用于相互交换和交互操作的流。应用程序在需要仅含一个基本流的流时可使用仅含一个基本流的节目流或传输流，这些流包含了所有必须的系统信息。每一个都包含一个基本流的多个多节目流或传输流可以在公共时间基准下

构造起来，以传送一个带视频和音频的完整节目。

0.5 定时模型

系统、视频和音频都有一个定时模型，在该模型中，从编码器的信号输入到解码器的信号输出之间的端到端延迟是恒定的，这一延迟是编码、编码器缓冲区、多路复用、传送或存储、解复用、解码器缓冲区以及展现延迟的总和。作为该定时模型的一部分，所有视频图像和音频采样仅展现一次(除非经过特殊编码)，且 I 图像间隔和音频采样速率在编码器和解码器中一致。系统流编码包括了定时信息以用于实现端到端延迟恒定的系统。实现不严格遵守该模型的解码器也是可能的。但此时的解码器必须负责以一种可接受方式完成以上要求。定时包含在本部分的标准规范中，所有有效的比特流，无论它们是如何被创建的，都必须遵循这一规范。

所有定时是根据称为系统时钟的公共时钟定义的。在节目流中，该时钟与视频或音频采样时钟之间可以有确定的比值，也可以有一个与比值略有偏差的工作频率，但仍提供精确的端到端定时和时钟补偿。在传输流中，系统时钟被限制为在任何时刻均与音频和视频采样时钟保持确定比值，这限制是为了简化解码器中的采样速率恢复。

0.6 条件存取

系统数据流的定义支持用于对编码在节目流和传输流中节目条件存取的加密和加扰。这里并未指定条件存取机制。由于设计了流定义，因此实际的条件存取系统的实现是合理的，并且有一些特定语法元素对此系统提供特定支持。

0.7 复用流操作

复用流操作包括通道数据读出的协调、时钟的调整以及缓冲区管理。这些任务是紧密相连的。若通道数据传送速率是可控的，则可调节数据传送以使解码缓冲区不发生上溢或下溢。但是若数据速率不可控，则基本流解码器就必须使它们的定时服从于从通道中接收的数据以避免上溢或下溢。

节目流由包组成，包头有助于以上任务的完成。包头指定了从通道中送来的每一字节进入节目流解码器的预定时间，这个预定到达时刻表作为时钟校正和缓冲区管理的参考。虽然解码器不一定要严格遵守该时刻表，但必须对有关偏差作出补偿。

类似地，传输流由传输流分组包构成。分组包头中包含有信息以指定从通道中送来的每一字节进入传输流解码器的预定时间。该时刻表提供了与上述节目流中完全相同的功能。

另一个复用流操作是解码器能确定解码传输流或节目流时所需的资源。每个节目流的第一个包均包含一些参数来协助解码器完成此功能。例如，流的最大数据速率以及同步视频通道的最大数目。传输流也包含类似的全局适用的信息。

每个传输流和节目流都包含一些信息，以标识组成一个节目的各基本流的相关特征以及基本流之间的相互关系。这些信息可能包括音频通道中的语言，以及在实现多层视频编码时各视频流之间的关系。

0.8 单个流操作（PES 分组包层）

基本的特定流操作为：

- a) 解复用；
- b) 多个基本流的同步回放。

0.8.1 解复用

编码时，节目流由复用基本流组成，而传输流则由复用基本流、节目流或其它传输流的内容多路复用而成。基本流除音频、视频数据流外还可能包括私有流、备用流及填充流。流被临时性地分割为分组包，分组包被串行化。一个 PES 分组包，包含仅来自一个基本流的编码字节。

节目流中的分组包长度可以是固定的或可变的，但必须遵守 7.1 和 7.2 中规定的约束。传输流分组包

长度是 188 字节，PES 分组包的长度可以是固定的或可变的，在大多数的应用中相对较长。

解码时需要对复用的节目流或传输流解复用以重建基本流，这可以借助节目流分组包头中的 `stream_id` 和传输流分组包头中的分组包标识码来完成。

0.8.2 同步

多个基本流之间的同步通过节目流或传输流中的相对显示时间(`relative_display_time`)来完成。时间戳通常以 90kHz 为单位。N 个基本流解码的同步是通过使流的解码被调整至一个公共主控时钟基准，而不是通过使流的解码彼此适应。主控时钟基准可以是 N 个解码器时钟中的一个，也可以是数据源时钟或某个外部时钟。

传输流可能包含多个节目，其中的每一节目都可能有自己的时钟基准。一个传输流中不同节目的时钟基准可能不同。

`relative_display_time` 用于单个基本流解码，它存在于传输流的 TS 分组包层中。编码器在捕获时记录时间戳，将该时间戳减去当前系统时间得到相对显示时间，并连同有关编码数据被传输到解码器，而解码器再利用它们来安排展现时间时，就能够实现端到端的同步。

单通道解码系统的同步，通过使用传输流中的 `transport_rate` 来实现。`transport_rate` 是编码比特流自身传输率的采样值。它们来自于同一个时间基准，该时间基准在同一个节目中也用作音频和视频的 `relative_display_time` 值。因为每一节目可能有自己的时间基准，所以一个包含多个节目的传输流中的每个节目各自有独立的 `transport_rate` 字段。在某些场合下，节目共享 `transport_rate` 字段也是可能的。确定一个节目与哪个 `transport_rate` 相关联的方法可以见 6.5 节目特定信息(PSI)。一个节目有且仅有一个相关的 `transport_rate` 时间基准。在易发生错误的环境中，为了在包丢失情况下仍能正确的恢复时钟，传输包中可以包含包计数值 `packet_count`。

0.8.3 与压缩层的关系

在某种意义上说 PES 分组包层是独立于压缩层之外的，但并不绝对。考虑到 PES 分组包的有效负载不需像本标准的第 2、3 部分所规定的那样以压缩层开始码字开头，PES 分组包层是独立的。例如，视频的起始码字可出现在 PES 分组包有效负载的任何位置，同时可被 PES 分组包首部所分开。然而，在 PES 分组包首部编码的时间标签是用来决定压缩层结构的显示时间（即显示单位）。此外，当基本流数据符合本标准的第 2、3 部分时，`PES_packet_data_bytes` 字段应遵照本部分进行字节对齐。

0.9 系统参考解码器

本部分使用了“系统目标解码器”(STD)，对传输流（见 6.2）称作“传输流系统目标解码器”(T-STD)，而对节目流（见 7.2）称作“节目流系统目标解码器”(P-STD)，以此提供时序和缓冲之间关系的形式化表示。因为 STD 的参数是依据 GB/T 20090.2 的各个字段（比如，缓冲区大小）而得到的，所以每一个基本流都有自己的 STD 参数。编码器应产生符合适当的 STD 约束的比特流，物理解码器可以假定一个流能在其 STD 上正确播放，但必须对它与 STD 在设计上的不同之处作出补偿。

0.10 应用

本部分定义的数据有很宽的应用范围，应用开发者可以简单地选择最合适的数据流。

现代的数据通信网络可以支持本系列标准的视频和音频。这要求一个实时传送协议。节目流可能适用于这样的网络传送。

节目流也适合于 CD-ROM 上的多媒体应用，对节目流的软件处理是便利的。

传输流更适用于易发生错误的环境，比如在远程网络或通过无线广播系统传送压缩的比特流。

许多应用要求在各种存储媒体(DSM)上存储和读出本标准比特流。为了便于对这种媒体的控制，本部分的附录 B 及 ISO/IEC 13818 的第 6 部分给出了一种数字存储媒体命令和控制 (DSM CC) 协议。

信息技术 先进音视频编码 第1部分：系统

1 范围

GB/T 20090 的本部分给出系统层编码规范。它主要被设计用于支持把 GB/T 20090 的第2部分和第3部分定义的视频和音频编码方式组合起来。系统层支持以下五个基本功能：

- a) 解码时多个压缩流的同步；
- b) 多条压缩流交织为一个单一流；
- c) 为启动解码而对缓冲区进行初始化；
- d) 连续的缓冲区管理；
- e) 时间标识。

一个 GB/T 20090.1 多路复用比特流可以是传输流或节目流。两种流均由 PES 分组包或包含其它必要信息的分组构成。两种流类型均支持来自具有一个共同时间基准节目的视频和音频压缩流的复用。传输流还支持来自具有独立时间基准的多个节目的视频和音频压缩流的复用。对于几乎不发生差错的环境而言，节目流通常更为合适，并且支持节目信息的软件处理。传输流更适合于可能出错的环境。

一个 GB/T 20090.1 多路复用比特流，不论是传输流还是节目流，其结构分两层：最外层是系统层，最内层是压缩层。系统层提供了使用系统中一个或多个压缩数据流所必需的功能。GB/T 20090 的音频和视频部分定义了音频和视频数据的压缩编码层。其它类型数据编码的定义不包括在本标准中，但如果它们符合第9章中定义的限制，则将被系统层支持。

2 规范性引用文件

下列文件中的条款通过 GB/T 20090 的本部分的引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本部分部分，然而，鼓励根据本部分部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

GB/T 15273.1—1994 信息技术 八位单字节编码图形字符集 第1部分：拉丁字母 (idt ISO 8859-1: 1987)

GB/T 17191.1—1997 信息技术 具有 1.5Mbit/s 数据传输率的数字存储介质运动图象及其伴音的编码 第1部分：系统(idt ISO/IEC 11172-1: 1993)

GB/T 17191.2—1997 信息技术 具有 1.5Mbit/s 数据传输率的数字存储介质运动图象及其伴音的编码 第2部分：视频(idt ISO/IEC 11172-2: 1993)

GB/T 17191.3—1997 信息技术 具有 1.5Mbit/s 数据传输率的数字存储介质运动图象及其伴音的编码 第3部分：音频(idt ISO/IEC 11172-3: 1993)

GB/T 17975.1—2000 信息技术 运动图象及其伴音信息的通用编码 第1部分：系统(idt ISO/IEC 13818-1: 1996)

GB/T 17975.2—2000 信息技术 运动图象及其伴音信息的通用编码 第2部分：视频(idt ISO/IEC 13818-2: 1997)

GB/T 17975.3—2002 信息技术 运动图像及其伴音信息的通用编码 第3部分：音频 (idt ISO/IEC 13818-3: 1998)

GB/T 20090.2—YYYY 信息技术 先进音视频编码 第2部分：视频

GB/T 20090.3—YYYY 信息技术 先进音视频编码 第3部分：音频

ISO 639-2: 1998 语种名称代码 第2部分：Alpha-3 代码

ISO/IEC 13818-6: 1998 信息技术 运动图像及其伴音信息的通用编码 第6部分：DSM-CC 扩展

ISO/IEC 13522-1: 1997 信息技术 多媒体和超媒体信息的编码 第1部分：MHEG 对象表示 基本表示法 (ASN.1)

3 术语和定义

下列术语和定义适用于本标准。若只适用于某一部分，则用方括号注明。

访问单元 access unit [系统]

一个展现单元的编码表示。对音频而言，一个访问单元就是一个音频帧的编码表示。

对视频而言，在视频压缩的情况下，一个访问单元包括一帧图像中所有的编码数据及跟随其后的任何填充，直至(但不包括)下一个访问单元的开始。如果图像不是由 group_start_code 或 sequence_header_code 起始，则访问单元由图像起始码开始。如果图像由 group_start_code 或 sequence_header_code 起始，则图像由上述起始码的第一个中的第一个字节开始。如果图像是比特流中 sequence_end_code 之前的最后一帧图像，则该编码图像中的最后一个字节与 sequence_end_code(包括 sequence_end_code)之间的所有字节均属于该访问单元。

3.1 比特率 bitrate

压缩的比特流从信道传输到解码器输入端的速率。

3.2 字节对齐 byte aligned

如果某一位在编码比特流中的位置从流的第一位算起是 8 的倍数，则该位是字节对齐的。

3.3 信道 channel

存储或传输 GB/T 20090 比特流的数字介质。

3.4 编码表示 coded representation

数据元素的编码表示格式。

3.5 压缩 compression

减少用于表示某个数据项的比特数目。

3.6 固定比特率 constant bitrate

压缩比特流从开始到结束比特率保持不变的方式。

3.7 受限系统参数流 CSPS (constrained system parameter stream) [系统]

遵循 GB/T 20090.1 中 9.8 所定义的约束条件的一个节目流。

3.8 循环冗余码校验 CRC(cyclic redundancy check)

用于检验数据的正确性。

3.9 数据元素 data element

编码之前和解码之后所表示的一个数据项。

3.10 解码流 decoded stream

压缩比特流的解码重构。

3.11 解码器 decoder

解码过程的具体实现实体。

3.12 解码(过程) decoding (process)

在 GB/T 20090.1 中定义的读入一个输入的编码比特流并产生解码的图像或音频信号样本的过程。

3.13 解码时间戳 DTS (decoding time-stamp) [系统]

PES 分组包头中的一个字段，用来指出一个访问单元在系统目标解码器中被解码的时刻。

3.14 数字存储介质 DSM (digital storage media)

数字存储或传输的设备或系统。

3.15 数字存储介质的命令和控制 DSM-CC (digital storage medium command and control)

3.16 授权控制信息 ECM (entitlement control message)

一些私有的条件访问信息，以指定控制语句和其它可能的，通常是流所特有的加扰及(或)控制参数。

3.17 授权管理信息 EMM (entitlement management message)

一些私有的条件访问信息，以指定权限等级或特定解码器的服务。它们可以被提供给一个或一组解码器。

3.18 编辑 editing

对一个或多个压缩比特流进行操作以生成一个新的压缩比特流的过程。编辑后的比特流必须与编辑前满足相同的要求。

3.19 基本流 ES (elementary stream) [系统]

泛指 PES 分组包中编码视频流、编码音频流或其它某一个编码比特流。一个基本流以有且仅有一个 stream_id 的 PES 分组包序列来传送。

3.20 基本流参考时钟 ESCR (elementary stream clock reference)[系统]

PES 流中的时间戳，PES 流解码器从中获取时间信息。

3.21 编码器 encoder

编码过程的具体实现实体。

3.22 编码(过程) encoding (process)

读入输入图像或音频样本流，并产生符合 GB/T 20090.1 的编码比特流的过程。该过程并未在 GB/T 20090.1 中规定。

3.23 熵编码 entropy coding

为减少冗余而对信号的数字表示进行的可变长无失真的一种编码方法。

3.24 事件 event

一个事件定义为有共同的时间基准、相关的起始时间和相关的结束时间的基本流的集合。

3.25 快速正向回放 fast forward playback [视频]

用快于实际速度，并按显示顺序显示图像序列或序列一部分的过程。

3.26 禁止 forbidden

GB/T 20090.1 中定义编码比特流时，绝不能使用的私有数值。

3.27 (复用)流 (multiplexed) stream [系统]

由零个或多个基本流按照符合 GB/T 20090.1 的方式组成的位流。

3.28 层 layer [视频和系统]

GB/T 20090 的第 1、第 2 部分中定义的系统和视频规范中数据层次结构中的一个层次。

3.29 分组 pack [系统]

分组由一个 pack 包头及随后的零个或多个 PES 分组包所构成，它是 GB/T 20090.1 的 7.3.3 中所描述的系统编码语法中的一个层次。

3.30 包数据 packet data [系统]

一个包内所含的连续数据字节，它来自某个基本流。

3.31 节目成分标识符 PEID (program element identifier) [系统]

它是 GB/T 20090.1 的表 23 中定义的用来标识一个或多个节目传输流中的节目成分的整数。

3.32 节目成分信息表 PEIT (program element information table) [系统]

PEIT 包括用来解复用传输流所必需的标准化数据。

3.33 补充 padding [音频]

一种调节音频帧平均时间长度的方法，对应于 PCM 采样的持续时间，有条件地在音频帧中加进时隙。

3.34 有效负载数据 payload

在一个包中跟在包头字节之后的那些字节。例如，一些传输流包的有效负载数据包括一个 PES 包头、PES 分组包数据字节或指针域以及 PSI 段或私有数据，但一个 PES 分组包的有效负载数据仅包含 PES 分组包数据字节。传输流包头和适应字段不是有效负载数据。

3.35 分组基本流 PES (packet element stream) [系统]

已分组的基本流。

3.36 PES 分组包 PES packet [系统]

传输基本流数据的数据结构。一个 PES 分组包包含一个 PES 分组包头，其后跟有一些来自基本数据流的连续字节。它是本部分 6.3.6 所描述的系统编码语法的一层。

3.37 PES 分组包头 PES packet header [系统]

一个 PES 分组包中的前导字段。当一个流不是补充流时，它一直到但并不包括 PES 分组包数据字节字段。在补充流的情况下，PES 分组包头被类似地定义为 PES 分组包的前导字段，直到但并不包括补充字节字段。

3.38 PES 流 PES stream [系统]

PES 流包含一些 PES 分组包。这些分组包的有效负载中包含来自于单个基本流的数据，且分组具有相同的流标识。特定的语义约束可以适用。

3.39 展现时间戳 PTS (Presentation time-stamp) [系统]

在 PES 分组包头中可能包含的一个字段，用来指出一个展现单元在系统目标解码器中被展现的时刻。

3.40 展现单元 PU (presentation unit) [系统]

已解码的一个音频访问单元或一帧解码图像。

3.41 节目 program [系统]

一个节目是节目元素的集合。节目元素可能是基本流。它不需要有任何定义的时间基准。那些有时间基准的则为共同时间基准，以用于同步展现。

3.42 节目参考时钟 PCR (program clock reference) [系统]

PES 流中的时间戳。PES 流解码器从中获取时间信息。

3.43 节目元素 program element [系统]

可能包含在一个节目中的基本流或其它数据流的一个流。

3.44 节目特定信息 PSI (program specific information) [系统]

PSI 包括用来解复用传输流和成功地再生节目所必需的标准化数据。

注：在本部分的 6.5 中对其进行了描述。专门定义的 PSI 数据的一个例子是非强制性的网络信息表。

3.45 随机访问 random access

从任意点开始读入编码比特流并解码的过程。

3.46 相对显示时间 relative_display_time

在 TS 中可能包含的一个字段，用来指出一个展现单元在系统目标解码器中被显示的相对时刻。

3.47 保留 reserved

用来在定义编码比特流时表示某数值可以用于将来的扩充。除非在本部分中有特别定义，否则所有的备用值应被设为'1'。

3.48 加扰 scrambling [系统]

对视频、音频或编码数据流进行改动以防止未经授权地接收明文信息。这种改动是在条件访问系统控制下的一种特定的过程。

3.49 源流 source stream

在压缩编码之前的一个非复用的样本流。

3.50 拼接 splicing [系统]

系统层对两个不同的基本流进行的连接操作。所产生的系统流完全符合 GB/T 20090.1。拼接可能会引起时间基准、连续性计数器、PSI 和解码的不连续。

3.51 起始码 start codes [系统]

嵌入在编码比特流中的 32 位码。它有若干用途，包括编码语法的某些层次的标识。起始码包含了一个 24 位的前缀(0x000001)和一个 8 位的流标识。

注：参见本部分表 16。

3.52 STD 输入缓冲区 STD input buffer [系统]

系统目标解码器输入端的一个先进先出缓冲区，用来存储解码之前的基本流的压缩数据。

3.53 静态图像 still picture

指编码的静态图像，仅包含一个帧内编码的图像的视频序列。该图像有一个相关的 PTS。若该图像有后续图像，则其后续图像的展现时间将比该静态图像晚至少两个图像周期。

3.54 系统头 system header [系统]

在本部分 7.3.5 中定义的一个数据结构。它携带了概述 GB/T 20090.1 节目流系统特性的信息。

3.55 系统参考时钟 SCR (system clock reference) [系统]

节目流中的时间戳。解码器从它获取时间信息。

3.56 系统目标解码器 STD (system target decoder) [系统]

用来描述 GB/T 20090.1 多路复用流语义的一个假想的解码过程参考模型。

3.57 时间戳 time-stamp [系统]

指示事件发生时刻。

例如一个字节的到达或一个展现单元的展现。

3.58 传输率 transport rate

传输流中的传输率。解码器从它获取传输率信息。

3.59 传输流包头 transport stream packet header [系统]

传输流包的前导字段，直到(且包括)连续计数器字段。

3.60 可变比特率 variable bitrate

传输流或节目流的一种属性，到达解码器输入端的字节速率随时间而变化。

4 符号

用于描述本部分的数学运算符类似于 C 程序语言中使用的运算符，但本部分特别定义了具有截断和舍入功能的整除运算。按位运算符的定义则假定采用整数的二进制补码表示。标号和记数循环通常由零开始。

4.1 算术运算符

+	加法运算
-	减法运算（二元运算符）或取反（一元前缀运算符）
++	递增， $x++$ 相当于 $x = x + 1$ 。当用于数组下标时，在自加运算前先求变量值
--	递减， $x--$ 相当于 $x = x - 1$ 。当用于数组下标时，在自减运算前先求变量值

×	乘法运算
^	幂运算
/	结果向零截断的整除。例如，7/4 和 -7/-4 取整为 1，-7/4 和 7/-4 为 -1
//	结果舍入为最近整数的整除。除非特别指明，否则 0.5 向上舍入。例如，3//2 舍入为 2，-3//2 舍入为 -2
DIV	是结果趋向 $-\infty$ 的带截断的整除
%	取模运算符，仅适用于正整数
Sign()	$\text{Sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$
NINT()	最近的整数运算符。返回与实数最接近的整数，对 0.5 进行向上舍入
sin	正弦
cos	余弦
exp	指数
$\sqrt{\quad}$	平方根
\log_{10}	以 10 为底的对数
\log_e	以 e 为底的对数

4.2 逻辑运算符

	逻辑加，OR
&&	逻辑乘，AND
!	逻辑非，NOT

4.3 关系运算符

>	大于
>=	大于或等于
<	小于
<=	小于或等于
==	等于
!=	不等于
max [,...,]	参数表中的最大值
min [,...,]	参数表中的最小值

4.4 按位运算符

&	与运算
	或运算
~	取反运算
$a \gg b$	将 a 以 2 的补码整数表示的形式向右移 b 位。仅当 b 取正数时定义此运算
$a \ll b$	将 a 以 2 的补码整数表示的形式向左移 b 位。仅当 b 取正数时定义此运算
>>	带符号扩展的右移
<<	补充零的左移

4.5 赋值

=	赋值运算符
---	-------

4.6 助记符

下列助记符用于描述编码比特流中不同的数据类型。

bslbf	比特串，即二进位串，左位在先。其中“左”是指本部分中书写比特串的顺序。 比特串书写成单引号括住的 1 和 0 的串，如'1000 0001'。比特串中的空格有助于阅读，但无实际意义。
ch	信道。
gr	音频层 II 中 3×32 个子频带采样的颗粒，音频层 III 中 18×32 个子频带采样的颗粒。
main_data	比特流的主数据部分，包含比例因子、Huffman 编码数据以及其它辅助信息。
main_data_beg	某一帧 main_data 在比特流中的开始位置，它等于前面一帧 main_data 的结束位置加上一个二进位，可从前一帧的 main_data_end 值计算出来。
part2_length	用作比例因子的 main_data 的二进制位数。
rpchof	余数多项式的系数，最高项排列在先。
sb	子频带。
scfsi	比例因子选择信息。
switch_point_l	使用窗口切换点所依据的比例因子频带数(长块比例因子频带)。
switch_point_s	使用窗口切换点所依据的比例因子频带数(短块比例因子频带)。
tcmsbf	二的补码整数，msb(符号)位优先。
uimsbf	无符号整数，高位在先。
vlcbf	可变长代码，左边的二进制位在先，这里“左边”是指书写可变长代码的顺序。
window	在 block_type = 2 时，实际时间狭道的数目($0 \leq window \leq 2$)。

注：多字节组成的字其字节顺序是最高字节在先。

4.7 常量

<i>p</i>	3.14159265359
<i>e</i>	2.71828182845

5 比特流语法的描述方式

解码器获得的比特流的描述见 6.1 和 7.1。比特流中的每个数据项以粗体表示。由它的名称、二进制长度、类型助记符和传输顺序描述。

在比特流中由一个已解码的数据元素所引起的动作，取决于该数据元素本身的值和先前已解码的数据元素的值。在包含对语法的语义描述的章节中描述了这些数据元素的解码操作以及在解码过程中所使用的状态变量的定义。以下结构用于表达数据元素何时出现以及何时为正常类型的条件。

注意，本语法使用 C 语言的约定，变量或表达式为非零值时等价于条件为真。

while(condition){	若条件为真，则一组数据元素便紧接着在数据流中出现，直到条件为假。
data_element ...	
}	
do {	数据元素至少出现一次。如此重复直到条件为假。
data_element ...}	
while (condition)	
if (condition) {	若条件为真，则第一组数据元素紧接着出现在数据流中。
data_element ...}	
else {	若条件为假，则第二组数据元素紧接着出现在数据流中。
data_element ...	
}	

for (i=0;i<n;i++){	数据元素组出现 n 次。数据元素组中的条件结构取决于循环控制变量 i 的值。 i 第一次出现时设置为 0，以后每出现一次就增加 1。
data_element ...	
}	

注意，数据元素组可以包含有嵌套条件结构。为简洁起见，若仅有一个数据元素时，{}可以省略。

data_element[]	数据数组，数据元素数量取决于上下文。
data_element[n]	数据数组中的第 $n+1$ 个元素。
data_element[m][n]	二维数据数组中的第 $m+1$ ， $n+1$ 个元素。
data_element[l][m][n]	三维数据数组中的第 $l+1$ ， $m+1$ ， $n+1$ 个元素。
data_element[m..n]	数据元素中从位 m 到位 n 的闭区间中的所有位。

尽管使用了过程式的术语来描述语法，但并不能认为图 6 或图 7 中实现的解码过程一定会令人满意。特别地，它们定义了一个正确的无差错的输入比特流。为了能够正确地开始解码，实际的解码器必须具有寻找起始码和同步字节(传输流)的方法，并且在解码时能够识别出错误，进行删除或插入等处理。识别这些事情的方法及所采取的措施都未标准化。

6 传输流比特流要求

6.1 传输流编码结构与参数

GB/T 20090.1 传输流编码层允许一个或多个节目组合在一个流中。来自每个基本流的数据与允许节目中基本流同步展现的信息一起多路复用。

一个传输流包括一或多个节目。视频与音频基本流由访问单元组成。

基本流数据由 PES 分组包携带。一个 PES 分组包包括一个包头，后跟包数据。PES 分组包被插入到传输流包中，每一 PES 分组包头的首字节位于一个传输流包的第一个可用的有效负载位置。

PES 分组包头以一个 32 位起始码开始，该起始码也标识该分组数据所属的流或流类型。PES 分组包头可以包含解码和展现时间戳(DTS 和 PTS)。PES 分组包头也包含其它可选字段。PES 分组包数据字段包含来自一个基本流的可变数目的连续字节。

传输流包以一个 4 字节前缀开始，定义见表 2。节目特定信息(PSI)表指定包含在传输流包中的数据内容。具有相同 PEID 值的传输流包仅携带来自同一个基本流的数据。

PSI 表携带于传输流中，共有以下 4 个：

- 节目关联表；
- 节目映射表；
- 条件访问表；
- 网络信息表。

这些表包含了对于解复用和展现节目所必要的也是足够的信息，表 29 所示的节目映射表规定了一个节目中基本流和 PEID 的对应关系以及一些相关信息。该表还指出了携带每一节目 `transport_rate` 的传输流包的 PEID。如果使用加扰，则将会出现条件访问表。网络信息表是可选的，它的内容未在本部分中规定。

传输流包可以为空。空的分组用于填充传输流，它们可能在再复用处理中被插入或删除。因此并不能假定空的分组会作为有效负载数据而被传送到解码器。

本部分并未规定可能被用作条件访问系统的一部分的编码数据。但为节目服务供应商提供了一些机制以用于在解码过程中传输和标识这些数据，以及如何正确地引用本标准所规定的的数据。这种支持是通过传输流包结构和条件访问表(见表 28)来提供的。

6.2 传输流系统目标解码器

6.3 中规定的传输流语义及第 9 章规定的语义限制需要精确地定义字节的到达、解码事件以及它们发生的时间。本部分使用一个称为传输流系统目标解码器(T-STD)的假想解码器来给出这些所需定义。有

关 T-STD 的进一步解释可参见附录 D，有关 T-STD 的一致性和实时接口参见附录 I，有关 T-STD 的抖动的网络问题参见附录 J。

T-STD 是一个用于精确定义这些术语及在创建和校验传输流时模型化解码过程的一个概念化模型。T-STD 仅为在此目的而定义。在 T-STD 中有三种解码器：视频、音频和系统。图 6 给出了一个例子。T-STD 的结构和所描述的时钟都不排除各种具有不同结构或定时机制的解码器对传输流进行不间断同步回放。

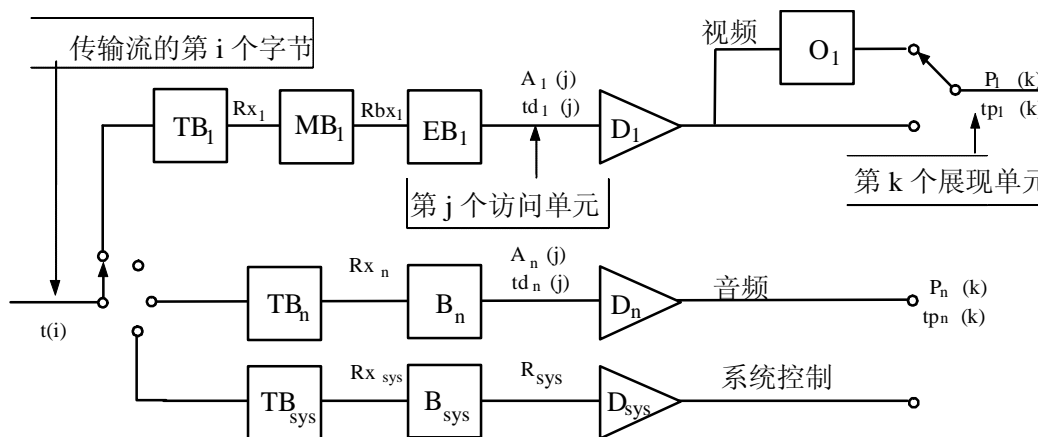


图 6 传输流系统目标解码器框图

以下记号用于描述传输流目标解码器，其中一部分已在前面图 6 中说明。

I, i', i''	传输流中字节的索引。第一个字节索引为 0。
J	基本流中访问单元的索引。
k, k', k''	基本流中展现单元的索引。
N	基本流的索引。
P	传输流中传输流包的索引。
T(i)	以秒为单位指出传输流中第 i 个字节进入系统目标解码器的时间，t(0)是一个任意常数。
transport_rate(i)	编码在 transport_rate 字段中的传输率，以比特/秒为单位来度量。其中 i 是传输率字段的最后一个字节的索引。
An(j)	基本流 n 中第 j 个访问单元，它按照解码顺序加以索引。
tdn(j)	基本流 n 中第 j 个访问单元在系统目标解码器中的解码时间，以秒 (s) 为单位。
Pn(k)	基本流 n 中第 k 个展现单元，它是解码 An(k) 的结果，并按展现顺序加以索引。
tpn(k)	基本流 n 中第 k 个访问单元在系统目标解码器中的展现时间，以秒 (s) 为单位。
T	以秒为单位的时间。
Fn(t)	系统目标解码器对基本流 n 在 t 时刻的输入缓冲区占用度，以字节为单位。
Bn	基本流 n 的主缓冲区，仅用于音频基本流。
BSn	缓冲区 Bn 的大小，以字节为单位。
Bsys	系统目标解码器的主缓冲区，用于存放正在解码过程中的节目的系统信息。
BSys	缓冲区 Bsys 的大小，以字节为单位。
MBn	基本流 n 的复用缓冲区，仅用于视频基本流。
MBSn	缓冲区 MBn 的大小，以字节为单位。
EBn	基本流 n 的基本流缓冲区，仅用于视频基本流。
EBSn	缓冲区 EBn 的大小，以字节为单位。

TB_{sys}	正在解码过程中的节目的系统信息的传输缓冲区。
TBS_{sys}	缓冲区 TB_{sys} 的大小，以字节为单位。
TB_n	基本流 n 的传输缓冲区。
TBS_n	缓冲区 TB_n 的大小，以字节为单位。
D_{sys}	节目流 n 中系统信息的解码器。
D_n	基本流 n 的解码器。
O_n	视频基本流 n 的重排序缓冲区。
R_{sys}	从 B_{sys} 中移走数据的速率。
Rx_n	从 TB_n 中移走数据的速率。
Rbx_n	在使用泄漏方式时，从 MB_n 中移走 PES 分组包有效负载数据的速率。
$Rbx_n(j)$	在使用 bbv_delay 方式时，从 MB_n 中移走 PES 分组包有效负载数据的速率。
Rx_{sys}	从 TB_{sys} 中移走数据的速率。
R_{es}	编码在序列头中的视频基本流速率。

6.2.1 系统时钟频率

T-STD 中所引用的传输率信息携带于本部分定义的一些数据字段中，见 6.3.4。在 `transport_rate` 字段中该信息作为节目的传输率值被编码。`transport_rate` 字段携带于传输流包的适应字段 `adaptation_field` 中。该传输流包具有一个和传输流节目映射段中定义的 `clock_recovery_PEID` 相等的 PEID 值。

实际解码器可能从这些值及它们各自的到达时间重建该时钟。以下给出了作用于节目的系统时钟频率的最小限制。

系统时钟频率值以赫兹（Hz）为单位，且满足如下约束：

$$27\,000\,000 - 810 \leq \text{system_clock_frequency} \leq 27\,000\,000 + 810$$

$$\text{system_clock_frequency 随时间的变化率} \leq 75 \times 10^{-3} \text{ Hz/s}$$

注：编码数据源应遵循更严格的误差限制以便适应用户的录制和回放装置的操作。

一个节目的系统时钟频率可能比要求的更精确。这一改进的精确性可经 8.20 定义的系统时钟描述子传递给解码器。

本部分定义的比特率按系统时钟频率测定。例如：27 000 000 比特/秒的比特率，意味着每 8 个时钟周期传输 1 字节数据。

在本部分中，术语“系统时钟频率”多次用于表示符合这些要求的时钟频率。为了便于表示和理解，在出现 `relative_display_time` 的公式中，时间精确到 $(300/\text{系统时钟频率})$ 秒的整数倍，`transport_rate` 精确到比特/秒。

6.2.2 传输流系统目标解码器的输入

传输流系统目标解码器的输入是一个传输流。一个传输流可能包含带有独立时间基准的多个节目，但 T-STD 一次只解码一个节目。在 T-STD 模型中，所有时间信息均表示该节目的时间基准。来自传输流的数据以分段恒定速率进入 T-STD。第 i 个字节进入 T-STD 的时间 $t(i)$ 通过对随后的输入流中的传输率字段(`transport_rate`)进行解码以及对该节目中到从上一个 `transport_rate` 之后的第一个字节开始到第 i 字节之间的传输流字节进行计数来定义。其中，`transport_rate` 编码在待解码节目的 `avs` 专用传输数据字段中。

所有其余字节的输入到达时间 $t(i)$ ，如下式(4)所示，是由 $\text{transport_rate}(i')$ 计算而来的。

$$\begin{aligned} t(1) &= t_0 \\ t(i) &= (t(i'') + (i - i'') / \text{transport_rate}(i')) \end{aligned} \quad (4)$$

式中：

$t(1)$ — 包含当前被解码节目的第一个 `transport_rate` 字段之后的第一位的字节到达解码器的时间。

i ——传输流中满足 $i'' < i < i'$ 的任何字节的索引。

i'' ——包含可用于当前被解码节目的 i 字节之前最近的 `transport_rate` 字段的最后一位的字节索引。

i' ——包含可用于当前被解码节目的 i 字节紧随着的 `transport_rate` 字段的最后一位的字节索引。

当传输分组适应字段中的 `discontinuity_indicator` 指示出时间基准不连续时, 式(4)中对输入 T-STD 的字节的到达时间的定义在旧时间基准的最后一个 `transport_rate` 和新时间基准的第一个 `transport_rate` 之间并不适用。在这种情况下, 字节到达时间的确定应对式(4)进行修正, 所用的传输率是旧时间基准的最后一个 `transport_rate`。

对 `transport_rate` 值规定了一个容限, 定义为收到 `transport_rate` 的所允许的最大相对偏差。这一相对偏差可能是由于 `transport_rate` 值的不精确或再复用时对 `transport_rate` 的修正引起的。它不包括因网络抖动或其它原因造成的到达时间误差。`transport_rate` 相对偏差容限为 $\pm 5E-6$ 。

具有多节目和可变速率的传输流

传输流可能含有带独立时间基准的多个节目。每个独立节目具有各自的 `transport_rate` 集, 分别由各自的 `clock_recovery_PEID` 值来标识, 因此不同的 `clock_recovery` 集不能同时使用。对于进入 T-STD 的节目而言, 传输流速率是分段恒定的, 所以如果一个传输流速率是可变的, 它只能在当前所考虑的节目的 `transport_rate` 出现时变化。所以不同 `transport_rate` 集对应的传输流中速率改变点也是不同的。传输流进入 T-STD 的速率将因不同节目进入 T-STD 而不同, 因此, 在一个传输流包含多个具有不同时间基准的节目且传输流速率是可变的情况下, 不可能为整个传输流构造出一个一致的 T-STD 传输方案, 但构造一个带有多个可变速率节目的恒定比特率的传输流却是简单的。

6.2.3 缓冲

由 PEID 标识的含有来自基本流 n 的数据的完整的传输流包被传送到流 n 的传输缓冲区 TB_n 中。这包括重复的传输流包和无有效负载的分组。第 i 个字节从 STD 输入到 TB_n 是瞬时的。因此, 第 i 个字节在 $t(i)$ 时刻进入大小为 TSB_n 的流 n 的缓冲区。

所有进入缓冲区 TB_n 的字节都按以下规定的速率 R_{x_n} 被移出。作为 PES 分组包的一部分或其内容的字节被传送到音频基本流和系统数据的主缓冲区以及视频基本流的多路复用缓冲区 MB_n , 其余字节则不会传送到缓冲区中, 但可能会用于控制系统。重复的传输流包不会被传送到 B_n 、 MB_n 或 B_{sys} 。

缓冲区 TB_n 按以下规则被清空:

——当 TB_n 中无数据, $R_{x_n} = 0$ 时,

——或对于视频, $R_{x_n} = 1.2 \times R_{max}[\text{profile}, \text{level}]$

其中, $R_{max}[\text{profile}, \text{level}]$ 是根据档次 `profile` 和等级 `level` 来规定的。参见 GB/T 17975.2-2000 的表 8-13。这些表规定了每一基本视频流在一定档次和等级下的速率上限。

对于 GB/T 17191.2-1997 而言, 受限参数视频流 R_x 等于 $1.2 \times R_{max}$ 。其中, R_{max} 指 GB/T 17191.2-1997 中受限参数比特流的最大比特率。

——对于音频: $R_{x_n} = 2 \times 10^6 \text{ bit/s}$ 。

——对于系统数据: $R_{x_n} = 1 \times 10^6 \text{ bit/s}$ 。

R_{x_n} 是相对于系统时钟频率来度量的。

包含系统信息, 用于解码所选节目的完整传输流包以传输流速率进入系统传输缓冲区 TB_{sys} 。这包括 PEID 值为 0、1 或 2 (如果存在) 的传输流包和经过节目关联表(PAT)验证具有所选节目的 `program_map_PEID` 值的所有传输流包。由 NIT PEID 指定的网络信息表(NIT)数据, 不被传送到 TB_{sys} 。字节以速率 $R_{x_{sys}}$ 从 TB_{sys} 移出并送往 B_{sys} , 所有字节均瞬时传送。

重复的传输流包不被送到 B_{sys} 。

不进入 TS_n 或 TB_{sys} 的传输分组包被丢弃。

传输缓冲区大小固定为 512 字节。

基本流缓冲区大小 $EBS_1 \sim EBS_n$ 是为视频定义的, 它们与携带于序列头中的 `bbv_buffer_size` 字段相等。参见 GB/T 17191.2-1997 中的受限参数总结和 GB/T 17975.2-2000 中的表 8-14。

用于视频的复用缓冲区大小 $MBS_1 \sim MBS_n$ 定义如下：

对于 AVS 基准档次

$$MBS_n = BS_{mux} + BS_{oh} + bbv_{max}[profile, level] - bbv_buffer_size$$

其中，PES 分组包开销的缓冲区 BS_{oh} 定义为：

$$BS_{oh} = (1/750)s \times R_{max}[profile, level]$$

附加复用缓冲区 BS_{mux} 定义为：

$$BS_{mux} = 0.005s \times R_{max}[profile, level]$$

而 $bbv_{max}[profile, level]$ 在 GB/T 20090.2-2006 的表 B.4, 表 B.5 中定义, $R_{max}[profile, level]$ 在 GB/T 20090.2 的表 B.4, 表 B.5 中定义, bbv_buffer_size 携带在 GB/T 20090.2-2006 的 7.2.2.1 描述的序列头中。

对于 GB/T 17975.2-2000 比特流, 目标解码器的缓冲模型见 GB/T 17975.1-1997。

大小为 $BS_{mux} = 5ms \times R_{max}[profile, level]$ 的 MBS_n 的一部分被分配为缓冲区以允许复用操作, 剩下的部分用作 BS_{oh} , 还可能供初始化复用操作使用。

注 1: PES 开销的缓冲区空间由 7.2.4 定义的 PES-STD 在 PES 流中直接作出限制。利用 PES 流来构造传输流是可能的, 但不是必须的。

BS_n 缓冲区

主缓冲区大小 $BS_1 \sim BS_n$ 定义如下。

音频

$$BS_n = BS_{mux} + BS_{dec} + BS_{oh} = 3584 \text{ 字节}$$

解码缓冲区 BS_{dec} 的访问单元大小和 PES 分组包开销缓冲区 BS_{oh} 受下式限制：

$$BS_{dec} + BS_{oh} = 2848 \text{ 字节}$$

3584 字节缓冲区的一部分(736 字节)被分配作为缓冲区以允许多路复用操作。余下的 2848 字节由访问单元缓冲区 BS_{dec} 和 BS_{oh} 及附加多路复用操作共享。

系统

用于系统数据的主缓冲区 B_{sys} 大小为 1536 字节。

视频

对 GB/T 20090.2-2006 视频基本流, 数据由泄漏方式从 MB_n 传送到 EB_n 。

泄漏方式

对于基准档次：

$$R_{bx_n} = R_{max}[profile, level]$$

如果 MB_n 中有 PES 分组包有效负载数据且缓冲区 EB_n 不满, PES 分组包有效负载以速率 R_{bx} 从 MB_n 传送到 EB_n 。若 EB_n 满, 数据不从 MB_n 中移出。当一个数据字节从 MB_n 传送到 EB_n , 所有紧接在该字节之前的且位于 MB_n 中的 PES 包头 PES 分组包头字节被立即移出并丢弃。当 MB_n 中无 PES 有效负载数据时, 数据不从 MB_n 中移出。所有进入 MB_n 的数据从中出来。所有 PES 分组包有效负载字节在离开 MB_n 的瞬时进入 EB_n 。

访问单元移动

对于每一基本流缓冲区 EB_n 和主缓冲区 B_n , 在缓冲区中时间最长的访问单元 $A_n(j)$ 的所有数据在时刻 $td_n(j)$ 被立即移走。当显示单元有足够的缓冲并且访问单元 $A_n(j)$ 的所有数据都已进入 EB_n 或 B_n , 解码器开始解码, 此时刻即为最早解码时间 $td_n(j)$ 。对于音频信号, 所有存储在访问单元之前的或嵌入在访问单元数据中的 PES 分组包头和访问单元同时移出。访问单元移出后立即解码为展现单元。

系统数据

对于系统数据, 数据以速率 R_{sys} 从主缓冲区 B_{sys} 中移出, 只要缓冲区 B_{sys} 中至少有一个字节的数据。

$$R_{sys} = \max(80000 \text{ bit/s}, transport_rate(i) + 500) \quad (7)$$

注 3: 在高传输率情况下, 增加 R_{sys} 的目的在于允许增加节目特定信息的数据速率。

低延迟

在视频序列扩展中的 low_delay 置'1'时(参见 GB/T 17975.2-2000、GB/T 20090.2-2006), EB_n 缓冲区可能

下溢。此时，当在时刻 $td_n(j)$ 检查 T-STD 基本流缓冲区 EB_n 时，访问单元的整个数据可能不在 EB_n 中。当这一情况出现时，缓冲区将在每隔两个字段周期重复检查，直到数据进入缓冲区。此时整个访问单元应瞬时从 EB_n 缓冲区移走，缓冲区 EB_n 将不会发生上溢。

当 `low_delay_mode` 标志置'1'时， EB_n 的下溢允许无限制地连续出现，T-STD 解码器将在 $td_n(j)$ 从 EB_n 缓冲区移出访问单元。注意解码器可能不能按 `display_time` 的指示重建正确的展现时间，直到 EB_n 缓冲区的下溢终止且在比特流中找到 `relative_display_time`。

特技模式

当含有 B 类型视频访问单元起始分组的 PES 分组包头中的 `DSM_Trick_mode` 标志为 1 且 `trick_mode_control` 字段为'001'(慢动作)或'010'(冻结帧)或'100'(慢倒)，B 图像访问单元不从视频数据缓冲区 EB_n 中移出直到该图像的任意字段的最后一次解码和展示。字段和图像的重复展现在 6.3.8 中的慢动作，慢倒和 `field_id_cntrl` 中定义。访问单元在规时间信息间及时地从 EB_n 移出，这取决于 `rep_cntrl` 的值。

当包含图像起始码的第一个字节的分组的 PES 分组包头中的 `DSM_trick_mode` 标志置'1'时，`trick_mode` 状态在 PES 分组包中图像起始码从缓冲区 EB_n 中移去时为真，且保持到 `DSM_trick_mode` 标志为'0'的 PES 分组包头被 T-STD 接收。特技模式状态为真时，缓冲区 EB_n 可能下溢，而来自一般流的所有其余限制均保留。

6.2.4 解码

$B_1 \sim B_n$ 以及 $EB_1 \sim EB_n$ 中的缓冲区存储的各条基本流被解码器 $D_1 \sim D_n$ 瞬时解码，且可能在 T-STD 输出端展现之前在重排序缓冲区 $O_1 \sim O_n$ 中延迟。重排序缓冲区仅当一些访问单元不符合播放顺序时，用于视频基本流。这些访问单元在显示前须重排序。特别地，若 $P_n(k)$ 是一帧或多帧 B 图像前的一帧 I 图像或 P 图像，在展现前它必须在重排序缓冲区 O_n 中延迟。在当前图像被存储之前，先前存储在 O_n 中所有的图像被展现。 $P_n(k)$ 延迟到下一 I 图像或 P 图像被解码。当它被存储在重排序缓冲区中时，下一个 B 图像被解码和显示。

展现单元 $P_n(k)$ 在 $tp_n(k)$ 时被展现。应该注意从视频基本流的开始使用足够的重排序延迟以满足整个流的需求。例如，开始仅包含 I 和 P 图像但后来包含 B 图像的流在流开始时应包括重排序延迟。

GB/T 17975.2-2000 或 GB/T 20090.2-2006 详细解释了视频图像重排序。

6.2.5 展现

解码系统的功能是从压缩数据重建展现单元并在正确的展现时间同步播放。尽管实际音频和视频播放设备，有着一定的不同的延迟及可能由后处理或输出方式造成的延迟，系统目标解码器将这些延迟假定为 0。

在图 6 的 T-STD 中一个视频展现单元(一个图像)在展现时间 $tp_n(k)$ 瞬时播放。

在 T-STD 中，音频展现单元在它的展现时间 $tp_n(k)$ 开始输出，此时解码器瞬时播出第一个样本。展现单元中的后续样本以音频采样率顺序播出。

6.2.6 缓冲区管理

传输流的构造应满足本条的条件。本条使用为 STD 定义的记号。

TB_n 和 TB_{sys} 不应发生上溢，每秒至少一次为空。 B_n 不应上溢或下溢， B_{sys} 不应上溢。

除非视频序列扩展中的低延迟标志被置为'1'(见 GB/T 20090.2-2006 或 GB/T 17975.2-2000 中的 6.2.2.3) 或 `trick_mode` 状态为真， EB_n 不应下溢。

当用来指定传送的泄漏方式生效时， MB_n 不应上溢，且每秒至少一次为空。 EB_n 不应上溢。

通过 STD 缓冲区的任意数据的延迟应不大于 1s，除非是静态图像视频数据。明确地说，即对所有 j 及访问单元 $A_n(j)$ 中字节 i ，有 $td_n(j) - t(i) \leq 1$ 。

对静止图像视频数据，所有 j 和访问单元 $A_n(j)$ 中的所有字节 i ，延迟限制为 $td_n(j) - t(i) \leq 60s$ 。

上溢和下溢的定义
假定 $F_n(t)$ 是 T-STD 缓冲区 B_n 在某个瞬间的占用程度。
在 $t=t(0)$ 前的瞬时, $F_n(t)=0$ 。
若对所有的 t 和 n ,

$$F_n(t) \leq BS_n,$$

则不发生上溢。

若对所有的 t 和 n ,

$$0 \leq F_n(t),$$

则不发生下溢。

6.3 传输流语法语义规范

以下语法描述了一个字节流。传输流包长度为 188 字节。传输流语法图形表示参见附录 F。

6.3.1 传输流

传输流见表 1。

表 1 传输流

语 法	位数	助记符
AVS_transport_stream() { do { transport_packet() } while (nextbits() == sync_byte)		

6.3.2 传输流包层

传输流包层见表 2。

表 2 本部分的传输分组

语 法	位数	助记符
transport_packet() { sync_byte transport_error_indicator payload_unit_start_indicator transport_priority PEIT_indicator transport_scrambling_control adaptation_field_control continuity_counter	 8 1 1 1 13 2 2 4	 bslbf bslbf bslbf bslbf uimsbf bslbf bslbf uimsbf

<pre> if (adaptation_field_control == '10' adaptation_field_control == '11') { adaptation_fields() } if (adaptation_field_control == '01' adaptation_field_control == '11'){ for (i = 0; i < N; i++) { data_byte </pre>	8	bslbf
---	---	-------

表 2 （续）

语 法	位数	助记符
<pre> } } } </pre>		

6.3.3 传输流包层中各字段的语义定义

关于同步字和PEID值的选择参见附录G。

同步字节字段 `sync_byte`

一个固定的值为 ‘0100 0111’ (0x47)的 8 位字段。

传输错误指示符字段 `transport_error_indicator`

一个 1 位标志。置'1'时表示相关传输流包中至少有一个不可纠正的位差错。该位可能被传输层外部的实体置'1'。置'1'后，除非错误被纠正，值不会恢复为'0'。

有效负载数据单元起始指示符字段 `payload_unit_start_indicator`

一个 1 位标志，对于携带 PES 分组包(见 6.3.6)或 PSI 数据(见 6.5)的传输流包有标准的含义。在传输流包的有效负载数据包含 PES 分组包数据时，该标志位有以下意义：'1'表示传输流包的有效负载数据以 PES 分组包的首字节开始，'0'表示该传输流包不以 PES 分组包开始。若值为'1'，则有且仅有 1 个 PES 分组包在传输流包开始，它也适用于流类型为 0x06 的私有流(见表 30)。

在传输流包的有效负载数据包含 PSI 数据时，该标志位有以下意义：若传输流包携带有 PSI 段的第一个字节，则该值应为'1'，以表示传输流包中有效负载数据的首字节带有 `point_field`。若传输流包不包含 PSI 段的第一个字节，则该值应为'0'，以表示传输流包中有效负载数据的首字节无 `point_field`。见 6.5.1 和 6.5.2 。这也适用于流类型为 0x05 的私有流(见表 30)。

对空的分组而言，该标志位置'0'。

对于只携带私有数据的传输流包，该位的意义本部分未作规定。

传输优先级字段 `transport_priority`

一位指示符。置'1'时表示相关分组比 PEID 相同但该位不为'1'的其它分组有更高的优先级。传输机制能用它来区分基本流中数据的优先级。该字段可能被仅编码在一个 PEID 中或不考虑 PEID，这取决于应用程序。该字段可能被信道专用编码器或解码器改变。

PEIT 标志字段 `PEIT_indicator` – 13 位字段, PEIT 表的标识符。当且仅当传输包中包含 PEIT 的时

候,该值为 0x000F.

传输加扰控制字段 `transport_scrambling_control`

2 位字段, 指出传输流包有效负载数据的加扰方式。传输流包头和适应字段不应该被加扰。对空的分组而言, 该字段值应设定为'00'。见表 3。

表 3 加扰控制值

值	描 述
00	非加扰
01	用户定义
10	用户定义
11	用户定义

适应字段控制字段 `adaptation_field_control`

2 位字段, 指出传输流包头后面是否有适应字段及(或)有效负载数据。见表 4。

表 4 适应字段控制值

值	描 述
00	ISO/IEC保留, 以供将来使用
01	没有适应字段, 仅有有效负载
10	仅有适应字段, 没有有效负载
11	跟有效负载的适应字段

GB/T 20090.1 解码器会丢弃 `adaptation_field_control` 字段值为'00'的传输流包。对空的分组而言, 该字段值应为'01'。

连续性计数器字段 `continuity_counter`

4 位字段, 随着每个具有相同 `PEID` 值的传输流包而递增。在达到最大值后, 回卷为 0。当分组的 `adaptation_field_control` 字段等于 '00' 或 '10' 时, 该字段不应递增。

在传输流中, 重复分组可以作为两个且仅两个具有相同 `PEID` 的连续传输流包来传送。重复分组应该与初始分组具有相同的 `continuity_counter` 值, 且 `adaptation_field_control` 字段应等于 '01' 或 '11'。在重复分组中, 初始分组中的每个字节都将重复, 但节目时间参考字段例外。若有节目参考时间字段, 则应编码为一个有效值。

`continuity_counter` 值在每个特定的传输流包中是连续的。在具有相同 `PEID` 的相邻两个传输流包之间相差一个正值。在 `adaptation_field_control` 字段值为'00'或'10'时, 或如上所述的重复分组中, 不发生递增。当 `discontinuity_indicator` 被设为'1'(见 6.3.4)时, `continuity_counter` 可能会不连续。对于空的分组, `continuity_counter` 的值未定义。

数据字节字段 `data_byte`

PES 分组包结构中的连续字节, 在 `PEID` 值为 0x1FFF 时可取任何值, 长度见 6.3.4 。它应当是来自于 PES 分组包(见 6.3.4), PSI 段(见 6.5)的连续字节的数据, 或 PSI 段之后的分组填充字节, 或不在 `PEID` 所指定的结构中的私有数据。对于 `PEID` 值为 0x1FFF 的空的分组, 该字段可以是任何值。`data_byte` 的数目 N, 由 184 减去适应字段中的字节数所规定。如下 6.3.4 所述。

6.3.4 适应字段

适应字段见表 5。

表 5 传输流适应字段

语 法	位数	助记符
Adaptation_fields() {		
adaptation_field_length	8	uimsbf
if (adaptation_field_length>0) {		
discontinuity_indicator	1	bslbf
random_access_indicator	1	bslbf
elementary_stream_priority_indicator	1	bslbf
PCR_flag	1	bslbf
OPCR_flag	1	bslbf
splicing_pointer_flag	1	bslbf
avs_specific_transport_data_flag	1	bslbf
adaptation_field_extension_flag	1	bslbf
if(PCR_flag == '1') {		
program_clock_reference_base	33	uimsbf
Reserved	6	bslbf
program_clock_reference_extension	9	uimsbf
}		
if(OPCR_flag == '1'){		
original_program_clock_reference_base	33	uimsbf
Reserved	6	bslbf
original_program_clock_reference_extension	9	uimsbf
}		
If(splicing_point_flag == '1'){		
splice_countdown	8	tcimsbf
}		
If(avs_specific_transport_data_flag == '1')		
{		
avs_specific_transport_data_length	8	uimsbf
Reserved	2	bslbf
packet_count_flag	1	bslbf
transport_rate_flag	1	bslbf
relative_display_time_flag	1	bslbf
linked_flag	1	Bslbf

语 法	位数	助记符
anchor_flag	1	Bslbf
if(packet_count_flag)		
packet_count	32	Uimbsbf
if(transport_rate_flag)		
transport_rate	32	Uimbsbf
if(relative_display_time_flag)		
relative_display_time	32	Uimbsbf
if(anchor_flag) {		
PEIT_seq_num	8	Uimbsbf
TP_seq_num	8	Uimbsbf
}		
if (linked_flag) {		
for(i=0;i< num_of_anchors;i++){		
AP_PEIT_seq_num	8	Uimbsbf
AP_TP_seq_num	8	uimbsbf
}		
}		
for(i=0; i<N; i++){		
private_data_byte	8	Uimbsbf
}		
}		
If(adaptation_field_extension == '1'){		
adaptation_field_extension_length	8	Uimbsbf
ltw_flag	1	Bslbf
piecewise_rate_flag	1	Bslbf
seamless_splice_flag	1	Bslbf
Reserved	5	Bslbf
If(ltw_flag == '1'){		
ltw_valid_flag	1	Bslbf
ltw_offset	15	Uimbsbf
}		
If(piecewise_rate_flag == '1'){		
Reserved	2	Bslbf
piecewise_rate	22	Uimbsbf

语 法	位数	助记符
}		
If(seamless_splice_flag == '1') {		
splice_type	4	Bslbf
DTS_next_AU[32..30]	3	Bslbf
marker_bit	1	Bslbf
DTS_next_AU[29..15]	15	Bslbf
marker_bit	1	Bslbf
DTS_nxt_AU[14..0]	15	Bslbf
marker_bit	1	Bslbf
}		
for(i=0;i<N;i++) {		
Reserved	8	Bslbf
}		
}		
for(i=0;i<N;i++) {		
stuffing_byte	8	Bslbf
}		
}		
}		

6.3.5 适应字段语义

适应字段中拼接传输流参见附录K。

适应字段长度字段 `adaptation_field_length`

8 位字段，指示紧随其后 `adaptation_field` 的字节长度。值为 0 用于在传输流包中插入单个填充字节。当适应字段控制的值为‘11’时，该字段值应在 0 到 182 之间。当适应字段控制的值为‘10’时，该字段值应为 183。对于携带 PES 分组包的传输流包，在没有足够的 PES 分组包数据来完全填满传输流包有效负载字节时，需要填充。通过定义一个比其内部包含的数据元素的长度之和还要长的适应字段来完成填充，这样适应字段后所剩的有效负载字节恰好能容纳有用的 PES 分组包数据。适应字段中多余的空间被填入填充字节。

这是对携带 PES 分组包的传输流包所允许的唯一的填充方法。对携带 PSI 的传输流包，6.5 中描述了另一种填充方法。

不连续性指示符字段 `discontinuity_indicator`

1 位字段，置‘1’时表示当前传输流包的不连续状态为真。当该字段为‘0’或不出现时，不连续状态为假。该字段用于两种类型的不连续性：系统时基不连续和 `continuity_counter` 不连续。

系统时基不连续性是通过在具有 `clock_recovery_PEID` 的传输流包中使用 `discontinuity_indicator` 来指出的(见 6.5.9)。当不连续状态为真时，具有相同 PEID 的传输流包中的下一个 `transport_rate` 代表相关节目的新传输率值。系统时钟不连续点被定义为包含一个有新系统时基 `transport_rate` 的分组中的第一个字节到达 T-STD 输入端的瞬间时刻。在系统时基发生不连续的分组中，`discontinuity_indicator` 应设定为‘1’。该分组之前的具有相同 `clock_recovery_PEID` 的传输流包中的 `discontinuity_indicator` 也可被设定为‘1’。在这种情况下，一旦一个 `discontinuity_indicator` 被设定为‘1’，则在后继的所有传输流包中应一直被设定为‘1’，直到且包括含有第一个具有新系统时基的 `transport_rate` 的传输流包。在系统时基不连续发生后及下一个系统时基不连续发生前，应至少收到两个具有系统时基的 `transport_rate`。此外，除非特技模式状态为真，仅来源于两个系统时基的数据在任何时候应出现在一个节目的 T-STD 缓冲区集合中。

在系统时基出现不连续之前，包含指出新时基的 `relative_display_time` 的传输流包中的第一个字节不应该到达 T-STD 的输入端。不连续发生后，包含指出前一个系统时基的 `relative_display_time` 的传输流包中的第一个字节不应该到达 T-STD 的输入端。

`continuity_counter` 的不连续是通过使用任何传输流包中的 `discontinuity_indicator` 来指出的。当 PEID 不是 `clock_recovery_PEID` 的任何传输流包中的不连续状态为真时，该分组中的 `continuity_counter` 与具有相同 PEID 的前一个传输流包相比，可能不连续。当 PEID 是 `clock_recovery_PEID` 的传输流包中的不连续状态为真时，只有在系统时基出现不连续的分组中，`continuity_counter` 才可能会不连续。

`continuity_counter` 的不连续点出现在传输流包中的不连续状态为真且该分组中的 `continuity_counter` 与具有相同 PEID 的前一个节目流分组相比不连续时。从不连续状态的开始到结束，`continuity_counter` 最多只能出现一次不连续。此外，对所有不是 `clock_recovery_PEID` 的 PEID，当具有某个 PEID 的分组中的 `discontinuity_indicator` 为‘1’时，后续的具有相同 PEID 的传输流包中的 `discontinuity_indicator` 可以为‘1’，但在具有该 PEID 的第三个后续传输流包中，不应该设定为‘1’。

在本小节中，基本流访问点定义如下：

视频：视频序列头的第一个字节。

音频：音频帧的第一个字节。

在包含基本流数据的传输分组中发生 `continuity_counter` 不连续后，具有相同 PEID 的传输流包中基本流数据的第一个字节应该是基本流访问点的第一个字节，对视频而言，还可以是后接访问点的 `sequence_end_code`。包含基本流数据，PEID 不为 `clock_recovery_PEID`，`continuity_counter` 发生不连续且出现 PTS 或 DTS 的每个传输流包应在相关节目的系统时基出现不连续后到达 T-STD 的输入端。对于不连续状态为真的情况而言，若出现两个具有相同 PEID，相同的 `continuity_counter` 值及

adaptation_field_control 值为'01'或'11'的传输流包，则第二个分组可能被丢弃。传输流不应该以此种方式构造，因为丢弃传输流包会导致 PES 分组包有效负载数据或 PSI 数据的丢失。

在包含 PSI 信息的传输流包中出现值为'1'的 discontinuity_indicator 之后，PSI 段的版本号中可能出现一个不连续。在出现这种不连续时，应发送相关节目的 TS_program_map_section 的一个版本，其中 section_length == 13 且 current_next_indicator == 1。这样，就不对 program_descriptor 和基本流数据进行描述。此时，应在后面为每个受影响的节目加上 TS_program_map_section 的一个版本，其中版本号递增 1 且 current_next_indicator == 1，以包含一个完整的节目定义。这指出了 PSI 数据中的版本变化。

随机访问指示符字段 random_access_indicator

1 位字段，指出当前传输流包以及具有相同 PEID 的后续传输流包中包含一些信息以帮助在该点进行随机访问。具体而言，当该位为'1'时，如果 PES 流类型(见表 30)为 0x01, 0x02 或 0x42，则具有当前 PEID 的起始于传输流包中有效负载的下一个 PES 分组包应包含视频序列头的第一个字节；如果 PES 流类型为 0x3, 0x4 或 0x43，则应包含音频帧的第一个字节。此外，对视频而言，展现时间戳应出现在序列头之后包含第一帧图像的 PES 分组包中。对音频而言，应出现在包含音频帧首字节的 PES 分组包中。在 clock_recovery_PEID 中，含有 transport_rate 字段的传输流包的 random_access_indicator 可能仅设定为'1'。

基本流优先级指示符字段 elementary_stream_priority_indicator

1 位字段，指出在有相同 PEID 的包中，传输流包的有效负载中携带的基本流数据的优先级。值为'1'表示有效负载比其它传输流包中的优先级高。对视频而言，只有当有效负载包含来自于帧内编码片的一个或多个字节时，该字段才可能为'1'。值为'0'表示有效负载与其它该字段值不为'1'的分组有相同的优先级。

PCR 标志字段 PCR_flag

1 位标志。置'1'时表示适应字段包含分两部分编码的 PCR；置'0'时表示适应字段不包含任何 PCR 字段。

OPCR 标志字段 OPCR_flag

1 位标志。置'1'时表示适应字段包含分两部分编码的 OPCR；置'0'时表示适应字段不包含任何 OPCR 字段。

拼接点标志字段 splicing_point_flag

1 位标志。置'1'时表示 splice_countdown 字段应出现在相关适应字段中，以说明拼接点的出现；置'0'时表示 splice_countdown 字段不出现在适应字段中。

avs 专用传输数据标志字段 avs_specific_transport_data_flag

1 位标志。置'1'时表示适应字段中含有 avs 专用传输数据以及一个或多个 private_data 字节；置'0'时表示适应字段不含有任何 avs 专用传输数据或 private_data 字节。

适应字段扩展标志字段 adaptation_field_extension_flag

1 位标志。置'1'时表示存在一个适应字段扩展。置'0'时表示适应字段不含有适应字段扩展。

节目参考时钟字段 program_clock_reference_base, program_clock_reference_extension

分两个部分编码的 42 位字段。第一部分 program_clock_reference_base，是一个由 PCR_base(i)给出其值的 33 位字段，见式 (2)。第二部分 program_clock_reference_extension，是一个由 PCR_ext(i)给出其值的 9 位字段，见式 (3)。PCR 表示包含 program_clock_reference_base 最后一位的字节到达系统目标解码器输入端的期望时间。

原始节目参考时钟字段 original_program_clock_reference_base, original_program_clock_reference_extension

分两个部分编码的 42 位字段。这两个部分，基础与扩展，与 PCR 字段中对应的两个部分的编码方法完全一致。OPCR 的出现由 OPCR_flag 来表示。只有在出现 PCR 字段的传输流包中的 OPCR 字段才会被编码。在单个节目和多个节目的传输流中都允许有 OPCR。

OPCR 有助于从另一个传输流来重构单个节目传输流。在重构初始的单个节目传输流中，OPCR 可以被拷贝到 PCR 字段。只有在初始的单个节目传输流完全被重构时，所产生的 PCR 值才是有效的。这

至少包括出现在初始传输流中的任何 PSI 和私有数据分组且可能需要其它专门的排列。这也意味着，在初始单个节目传输流中，OPCR 必须是它的相关 PCR 的一个完全一致的拷贝。

OPCR 表述如下：

$$\text{OPCR}(i) = \text{OPCR_base}(i) \times 300 + \text{PCR_ext}(i) \quad (8)$$

其中：

$$\text{OPCR_base}(i) = ((\text{system_clock_frequency} \times t(i)) \text{ DIV } 300) \% 2^{33} \quad (9)$$

$$\text{OPCR_ext}(i) = ((\text{system_clock_frequency} \times t(i)) \text{ DIV } 1) \% 300 \quad (10)$$

OPCR 字段被解码器忽略。任何多路复用器都不应该修改 OPCR 字段。

拼接倒计数字段 `splice_countdown`

8 位字段。表示一个可以是正数或负数的值。正值指出了跟在相关传输流包后直到一个拼接点处且具有相同 PEID 的传输流包的剩余数目。重复传输流包和仅包含适应字段的传输流包排除在外。拼接点位于相关 `splice_countdown` 字段为 0 的传输流包的最后一个字节之后。在 `splice_countdown` 为 0 的传输流包中，传输流包有效负载数据的最后一个数据字节应该是编码音频帧或编码图像的最后一个字节。对视频而言，相应的访问单元可能会也可能不会被 `sequence_end_code` 所终止。后面所跟的具有相同 PEID 的传输流包可能包含来源于具有相同类型的另一个不同的基本流的数据。

具有相同 PEID 的下一个传输流包(重复分组和无有效负载的分组不包括在内)中的有效负载数据应以 PES 分组包的第一个字节为开始。对音频而言，PES 分组包有效负载应以一个访问点为开始。对视频而言，PES 分组包有效负载应以一个访问点或后面跟有访问点的 `sequence_end_code` 为开始。因此，前一个编码音频帧或编码图像与分组边界对齐，或被填充后与分组边界对齐。在拼接点之后，倒数计数字段可能也会出现。当 `splice_countdown` 是一个负数值 (-n) 时，它表示相关传输流包是跟在拼接点之后的第 n 个分组(重复分组和无有效负载的分组不包括在内)。

对本小节而言，访问点定义如下：

视频： `video_sequence_header` 的首字节。

音频： 音频帧的首字节。

avs 专用传输数据长度字段 `avs_specific_transport_data_length`

8 位字段，指出其后的 avs 专用传输数据字节数。该数目不应使私有数据超出适应字段。

包计数标志字段 `packet_count_flag`

1 位标志。置'1'时表示 avs 专用传输数据字段中包含 `packet_count` 字段，置'0'时表示 avs 专用传输数据字段中不包含 `packet_count` 字段。

传输率标志字段 `transport_rate_flag`

1 位标志。置'1'时表示 avs 专用传输数据字段中包含 `transport_rate` 字段，置'0'时表示 avs 专用传输数据字段中不包含 `transport_rate` 字段。

相对显示时间标志字段 `relative_display_time_flag`

1 位标志。置'1'时表示 avs 专用传输数据字段中包含 `relative_display_time` 字段，置'0'时 avs 专用传输数据字段中不包含 `relative_display_time` 字段。

链接标志字段 `linked_flag`

1 位标志。表示当前传输包是一个被链接的数据包(LP)。

定位标志字段 `anchor_flag`

1 位标志。表示当前传输包是一个定位数据包(AP)。

包计数字段 `packet_count`

32 位无符号整数。`packet_count` 表示在当前传输流中当前包的计数值。该值随着传输包的增加递增，当达到最大值时回卷为 0。

传输率字段 `transport_rate`

32 位无符号整数。`transport_rate` 表示包含上一次 `transport_rate` 之后第一位的字节到达系统目标解码器输入端直到包含当前 `transport_rate` 最后一位的字节到达系统目标解码器输入端之间的期望平均传输

率。

相对显示时间字段 `relative_display_time`

32 位无符号整数。`relative_display_time` 表示基本流 n 的第 k 个展现单元在系统目标解码器中的相对展现时间, 即从包含 `relative_display_time` 最后一位的字节到达系统目标解码器输入端到 $P_n(k)$ 展现之间的时间。`relative_display_time` 的值以系统时钟频率的 $1/300$ (即 90 kHz) 为单位。`relative_display_time` 由展现时间和当前系统时间根据式(12)计算而来。

$$\text{relative_display_time}(k) = (\text{system_clock_frequency} \times (\text{tp}'_n(k) - t(r))) \div 300 \times 2^{32} \quad (12)$$

其中, r 表示包含 `relative_display_time` 最后一位的字节, $\text{tp}_n(k)$ 是展现单元 $P_n(k)$ 的展现时间。

对音频而言, 若 TS 私有数据段中有 `relative_display_time`, 则它是指 TS 分组包中开始的第一个访问单元。若 TS 分组包中有音频访问单元的首字节, 则有一个音频访问单元开始于该 TS 分组包中。

对视频而言, 若 TS 私有数据段中有 `relative_display_time`, 则它是指包含于 TS 分组包中开始的第一个图像起始码的访问单元。若 TS 分组包中有图像起始码的首字节, 则有一个图像起始码开始于该 TS 分组包中。

若音频中有滤波, 则系统模型假定滤波不会导致延迟。因此, 编码时 `relative_display_time` 所涉及的采样与解码时 `relative_display_time` 所涉及的采样是相同的。

在解码端, 显示时间 `display_timen(k)` 可由 `relative_display_time(k)` 和系统时间计算得到

$$\text{display_time}_n(k) = \text{relative_display_time}(k) + \text{STC_base}(r) \quad (13)$$

其中, r 表示包含 `relative_display_time(k)` 最后一位的字节, `display_timen(k)` 是在解码端展现单元 $P_n(k)$ 的展现时间, `STC_base(r)` 表示第 r 字节到达解码器时 `STC_base` 的值, `display_timen(k)` 和 `STC_base(r)` 都以系统时钟频率的 $1/300$ (即 90 kHz) 为单位。

PEIT 序列号 `PEIT_seq_num`

8 位字段。表示当前传输包所属的 PEIT 表的序列号, 选取范围为 0-255。

传输包序列号 `TP_seq_num`

8 位字段。表示当前传输包的序列号, 选取范围为 0-255。

定位 PEIT 序列号 `AP_PEIT_seq_num`

8 位字段。表示当前传输包所对应的 AP 的所属的 PEIT 表的序列号, 选取范围为 0-255。

定位传输包序列号 `AP_TP_seq_num`

8 位字段。表示当前传输包所对应的 AP 的传输包序列号, 选取范围为 0-255。

私有数据字节字段 `private_data_byte`

8 位的私有数据字段。在国际标准中未对其作出规定。

适应字段扩展长度字段 `adaptation_field_extension_length`

8 位字段, 指出紧跟其后的扩展的适应字段数据的字节数。如果有保留字节的话, 也包括在内。

合法时间窗口标志字段 `ltw_flag`

1 位字段, 置'1'时表示存在 `ltw_offset` 字段。

分段速率标志字段 `piecewise_rate_flag`

1 位字段, 置'1'时表示存在 `piecewise_rate` 字段。

无缝拼接标志字段 `seamless_splice_flag`

1 位标志, 置'1'时表示存在 `splice_type` 和 `DTS_next_AU` 字段。值为'0'时表示这两个字段都不存在。在 `splicing_point_flag` 不为'1'的传输流包中, 该字段不应该设置为'1'。一旦在 `splice_countdown` 为正值的传输流包中将该字段设定为'1', 在具有相同 PEID 且将 `splicing_point_flag` 设定为'1'的后续传输流包中也应将它设定为'1', 直到某分组中的 `splice_countdown` 为 0(包括该分组)。在设置该标志时, 若该 PEID 中携带的基本流为音频流, 则 `splice_type` 字段应设置为'0000'。若该 PEID 中携带的基本流为 GB/T 17975.2-2000 或 GB/T 20090.2-2006 视频流, 它应当满足 `splice_type` 值所指出的约束。

合法时间窗口有效标志字段 `ltw_valid_flag`

1 位标志。置'1'时表示 `ltw_offset` 值有效。值为'0'表示 `ltw_offset` 字段未定义。

合法时间窗口偏移字段 `ltw_offset`

15 位字段。只有在 `ltw_valid_flag` 的值为 1 时，其值才有定义。在有定义时，合法时间窗口偏移以 $(300/f_s)$ 秒为单位。其中， f_s 为该 PEID 所属节目的系统时钟频率，且满足：

$$\text{offset} = t_i(i) - t(i)$$

$$\text{ltw_offset} = \text{offset} // 1$$

其中， i 是该传输流包的首字节的索引，偏移量是编码在该字段中的值， $t(i)$ 是 T-STD 中的字节 i 的到达时间， $t_i(i)$ 是与传输流相关的称为“合法时间窗口”的时间间隔的上界。

合法时间窗口具有一种特性，即若传输流在时间 $t_i(i)$ ，即它的合法时间窗口的末尾，被传送到在时间 $t(i)$ 开始的 T-STD，且同一个节目的所有其它传输流包在它们的合法时间窗口的末尾被传送，那么：

对于视频：在该传输流包的首字节进入 T-STD 中该 PEID 的 MB_n 缓冲区时， MB_n 缓冲区将包含少于 184 字节的基本流数据，且 T-STD 中不会发生缓冲区冲突。

对于音频：在该传输流包的首字节进入 T-STD 中该 PEID 的 B_n 缓冲区时， B_n 缓冲区将包含少于 $BS_{dec} + 1$ 字节的基本流数据，且 T-STD 中不会发生缓冲区冲突。

根据缓冲区 B_n 的大小以及 MB_n 与 EB_n 之间的时间传送速率等因素，有可能确定另一个时间 $t_0(i)$ ，使得在间隔 $[t_0(i), t_i(i)]$ 中的任何时刻传送该分组时，不会出现 T-STD 缓冲区冲突。该时间间隔称为合法时间窗口。在本部分中未定义 $t_0(i)$ 的值。

本字段中的信息用于诸如再复用器之类的设备。这些设备可能需要该信息来重构缓冲区 MB_n 的状态。

分段速率字段 `piecewise_rate`

22 位字段，其含义只有在 `ltw_flag` 和 `ltw_valid_flag` 均被设定为'1'时才定义。当有定义时，它是一个指定假想位速率 R 的正整数，以用于定义跟在该分组后面且有相同 PEID 的传输流包中的合法时间窗口（但不包括 `legal_time_window_offset` 字段）的结束次数。

假定该传输流包的首字节和具有相同 PEID 的后续 N 个传输流包的索引分别为 $A_i, A_{i+1}, \dots, A_{i+N}$ ，且后 N 个传输流包在 `legal_time_window_offset` 字段中没有编码值，那么值 $t_i(A_{i+j})$ 应决定于

$$t_i(A_{i+j}) = t_i(A_i) + j \times 188 \times 8 \text{ (位/字节)} / R$$

其中， j 从 1 到 N 。

在该分组和要包括 `legal_time_window_offset` 字段的下一个分组之间的所有分组应该被看作 `legal_time_window_offset` 字段具有编码值：

$$\text{offset} = t_i(A_i) - t(A_i)$$

其中，值 $t_i(\cdot)$ 按上面的公式计算， $t(j)$ 是 T-STD 中的字节 j 的到达时间。

当该字段出现在没有 `legal_time_window_offset` 字段的传输流包中时，它的意义没有定义。

拼接类型字段 `splice_type`

4 位字段。该字段第一次出现以后，在它出现的后续的具有相同 PEID 的传输流包中应该有相同的值，直到 `splice_countdown` 计数为 0 的那个分组为止（包括该分组）。若携带在该 PEID 中的基本流是音频流，该字段值应为 '0000'。若携带在该 PEID 中的基本流是视频流，该字段指出了基本流为了拼接的目的而要遵循的条件。在表 6 到表 11 中，这些条件被定义为 `profile`，`level` 和 `splice_type` 的功能。

在这些表中，'splice_decoding_delay' 和 'max_splice_rate' 的值表示视频基本流应满足下列条件：

a) `splice_countdown` 为 0 的传输流包中所结束的编码图像的最后一个字节将在 bbv 模型的缓冲区中停留一段时间，时间长度为 $(\text{splice_decoding_delay } t_{n+1} - t_n)$ ，其中，在本小节中

- 1) n 是 `splice_countdown` 为 0 的传输流包中所结束的编码图像的索引，即上面提到的编码图像。
- 2) t_n 在 GB/T 20090.2-2006 中定义。
- 3) $(t_{n+1} - t_n)$ 在 GB/T 20090.2-2006 中定义。

注： t_n 是编码图像从 bbv 缓冲区中移走的时间， $(t_{n+1} - t_n)$ 是图像展现时间。

若 bbv 模型的输入在拼接点处切换到一个速率恒定为 'max_splice_rate' 的流并延续一段长度为 'splice_decoding_delay' 的时间，则 bbv 缓冲区不会溢出。

表 6 拼接参数表 1

简单档次主要等级, 主要档次主要等级, SNR 档次主要等级(两层),
空间档次高-1440 等级(基本层), 高档次主要等级(中间层+基本层)

splice_type	条 件
0000	splice_decoding_delay=120ms;max_splice_rate=15.0×10 ⁶ bit/s
0001	splice_decoding_delay=150ms;max_splice_rate=12.0×10 ⁶ bit/s
0010	splice_decoding_delay=225ms;max_splice_rate=8.0×10 ⁶ bit/s
0011	splice_decoding_delay=250ms;max_splice_rate=7.2×10 ⁶ bit/s
0100~1011	保 留
1100~1111	用户定义

表 7 拼接参数表 2

主要档次低等级, SNR 档次低等级(两层),
高档次主要等级(基本层)视频

splice_type	条 件
0000	splice_decoding_delay=115ms;max_splice_rate=4.0×10 ⁶ bit/s
0001	splice_decoding_delay=155ms;max_splice_rate=3.0×10 ⁶ bit/s
0010	splice_decoding_delay=230ms;max_splice_rate= 2.0×10 ⁶ bit/s
0011	splice_decoding_delay=250ms;max_splice_rate= 1.8×10 ⁶ bit/s
0100~1011	保 留
1100~1111	用户定义

表 8 拼接参数表 3

主要档次高-1440 等级, 空间档次高-1440 等级(所有层),
高档次高-1440 等级(中间层+基本层)视频

splice_type	条 件
0000	splice_decoding_delay=120ms;max_splice_rate=60.0×10 ⁶ bit/s
0001	splice_decoding_delay=160ms;max_splice_rate=45.0×10 ⁶ bit/s
0010	splice_decoding_delay=240ms;max_splice_rate= 30.0×10 ⁶ bit/s
0011	splice_decoding_delay=250ms;max_splice_rate= 28.5×10 ⁶ bit/s
0100~1011	保 留
1100~1111	用户定义

表 9 拼接参数表 4

主要档次高等级, 高档次高-1440 等级(所有层),
高档次高等级(中间层+基本层)视频

splice_type	条 件
-------------	-----

0000	splice_decoding_delay=120ms;max_splice_rate=80.0×10 ⁶ bit/s
0001	splice_decoding_delay=160ms;max_splice_rate=60.0×10 ⁶ bit/s
0010	splice_decoding_delay=240ms;max_splice_rate= 40.0×10 ⁶ bit/s
0011	splice_decoding_delay=250ms;max_splice_rate= 38.0×10 ⁶ bit/s
0100~1011	保 留
1100~1111	用户定义

表 10 拼接参数表 5

SNR 档次低等级(基本层)视频

splice_type	条 件
0000	splice_decoding_delay=115ms;max_splice_rate=3.0×10 ⁶ bit/s
0001	splice_decoding_delay=175ms;max_splice_rate=2.0×10 ⁶ bit/s
0010	splice_decoding_delay=250ms;max_splice_rate= 1.4×10 ⁶ bit/s
0011~1011	保 留
1100~1111	用户定义

表 11 拼接参数表 6

SNR 档次主要等级(基本层)视频

splice_type	条 件
0000	splice_decoding_delay=115ms;max_splice_rate=10.0×10 ⁶ bit/s
0001	splice_decoding_delay=145ms;max_splice_rate=8.0×10 ⁶ bit/s
0010	splice_decoding_delay=235ms;max_splice_rate= 5.0×10 ⁶ bit/s
0011	splice_decoding_delay=250ms;max_splice_rate= 4.7×10 ⁶ bit/s
0100~1011	保 留
1100~1111	用户定义

表 12 拼接参数表 7

空间档次高-1440 等级(中间层+基本层)视频

splice_type	条 件
0000	splice_decoding_delay=120ms;max_splice_rate=40.0×10 ⁶ bit/s
0001	splice_decoding_delay=160ms;max_splice_rate=30.0×10 ⁶ bit/s

表 13 （续）

splice_type	条 件
0010	splice_decoding_delay=240ms;max_splice_rate= 20.0×10 ⁶ bit/s
0011	splice_decoding_delay=250ms;max_splice_rate=19.0×10 ⁶ bit/s
0100~1011	保 留
1100~1111	用户定义

表 13 拼接参数表 8

高档次主要等级(所有层)，高档次高-1440 等级(基本层)视频

splice_type	条 件
0000	splice_decoding_delay=120ms;max_splice_rate=20.0×10 ⁶ bit/s
0001	splice_decoding_delay=160ms;max_splice_rate=15.0×10 ⁶ bit/s
0010	splice_decoding_delay=240ms;max_splice_rate= 10.0×10 ⁶ bit/s
0011	splice_decoding_delay=250ms;max_splice_rate= 9.5×10 ⁶ bit/s
0100~1011	保 留
1100~1111	用户定义

表 14 拼接参数表 9

高档次高等级(基本层)视频

splice_type	条 件
0000	splice_decoding_delay=120ms;max_splice_rate=25.0×10 ⁶ bit/s
0001	splice_decoding_delay=165ms;max_splice_rate=18.0×10 ⁶ bit/s
0010	splice_decoding_delay=250ms;max_splice_rate= 12.0×10 ⁶ bit/s
0011~1011	保 留
1100~1111	用户定义

表 15 拼接参数表 10

高档次高等级(所有层)视频

splice_type	条 件
0000	splice_decoding_delay=120ms;max_splice_rate=100.0×10 ⁶ bit/s
0001	splice_decoding_delay=160ms;max_splice_rate=75.0×10 ⁶ bit/s
0010	splice_decoding_delay=240ms;max_splice_rate= 50.0×10 ⁶ bit/s
0011	splice_decoding_delay=250ms;max_splice_rate= 48.0×10 ⁶ bit/s

表 16 (续)

splice_type	条 件
0100~1011	保 留
1100~1111	用户定义

解码时间戳下一个访问单元字段 DTS_next_AU

33 位字段，分三部分编码。对通过拼接点的连续和周期性解码而言，它指出了跟在该拼接点后面的第一个访问单元的解码时间。解码时间以 `splice_countdown` 为 0 的那个传输流包中有效的时基来表示。该字段第一次出现以后，在它所出现的所有后续传输流包中均应该有相同的值，直到 `splice_countdown` 为 0 的传输流包为止(包括该传输流包)。

填充字节字段 stuffing_byte

8 位字段，值固定为 ‘1111 1111’。它可以由编码器插入，而被解码器丢弃。

6.3.6 PES 分组包

PES 分组包见表 16。关于数据传输应用参见附录 E。

表 16 PES 分组包

语 法	位数	助记符
PES_packet(){		
packet_start_code_prefix	24	bslbf
stream_id	8	uimsbf
PES_packet_length	16	uimsbf
if(stream_id != program_stream_map		
&& stream_id !=padding_stream		
&& stream_id !=private_stream_2		
&& stream_id !=ECM		
&& stream_id !=EMM		
&& stream_id !=program_stream_directory		
&& stream_id !=DSMCC_stream		
&& stream_id !=ITU-T Rec.H.222.1 type E stream){		
'10'	2	bslbf
PES_scrambling_control	2	bslbf
PES_priority	1	bslbf
data_alignment_indicator	1	bslbf
Copyright	1	bslbf

original_or_copy	1	bslbf
PTS_DTS_flags	2	bslbf
ESCR_flag	1	bslbf
ES_rate_flag	1	bslbf
DSM_trick_mode_flag	1	bslbf
additional_copy_info_flag	1	bslbf
PES_CRC_flag	1	bslbf
PES_extension_flag	1	bslbf
PES_header_data_length	8	uimsbf
if(PTS_DTS_flags == '10'){		
'0010'	4	bslbf
PTS[32..30]	3	bslbf
marker_bit	1	bslbf
PTS[29..15]	15	bslbf
marker_bit	1	bslbf
PTS[14..0]	15	bslbf
marker_bit	1	bslbf
}		
If(PTS_DTS_flags == '11'){		
'0011'	4	bslbf
PTS[32..30]	3	bslbf
marker_bit	1	bslbf
PTS[29..15]	15	bslbf
marker_bit	1	bslbf
PTS[14..0]	15	bslbf
marker_bit	1	bslbf
'0001'	4	bslbf
DTS[32..30]	3	bslbf
marker_bit	1	bslbf
DTS[29..15]	15	bslbf
marker_bit	1	bslbf
DTS[14..0]	15	bslbf
marker_bit	1	bslbf
}		
If(ESCR_flag == '1'){		

Reserved	2	bslbf
-----------------	----------	--------------

表 16 (续)

语 法	位数	助记符
ESCR_base[32..30]	3	bslbf
marker_bit	1	bslbf
ESCR_base[29..15]	15	bslbf
marker_bit	1	bslbf
ESCR_base[14..0]	15	bslbf
marker_bit	1	bslbf
ESCR_extension	9	uimsbf
marker_bit	1	bslbf
}		
if(ES_rate_flag == '1'){		
marker_bit	1	bslbf
ES_rate	22	uimsbf
marker_bit	1	bslbf
}		
If (DSM_trick_mode_flag == '1'){		
trick_mode_control	3	uimsbf
If (trick_mode_control == fast_forward) {		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
}		
else if (trick_mode_control == slow_motion) {		
rep_cntrl	5	uimsbf
}		
else if (trick_mode_control == freeze_frame) {		
field_id	2	uimsbf
Reserved	3	bslbf
}		
else if (trick_mode_control == fast_reverse) {		
field_id	2	bslbf
Intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
}		

else if (trick_mode_control == slow_reverse) {		
--	--	--

表 16 (续)

语 法	位数	助记符
Rep_cntrl } Else Reserved } if (additional_copy_info_flag == '1'){ marker_bit additional_copy_info } if (PES_CRC_flag=='1'){ previous_PES_packet_CRC } if (PES_extension_flag == '1') { PES_private_data_flag pack_header_field_flag program_packet_sequence_counter_flag P-STD_buffer_flag reserved PES_extension_flag_2 If(PES_private_data_flag == '1'){ PES_private_data } If (pack_header_field_flag == '1'){ pack_field_length pack_header() } If (program_packer_sequence_counter_flag == '1'){ marker_bit program_packet_sequence_counter marker-bit MPEG1_MPEG2_indentifier original_stuff_length }	5 5 1 7 16 1 1 1 1 3 1 128 8 1 7 1 1 6	uimsbf bslbf bslbf bslbf bslbf bslbf bslbf bslbf bslbf bslbf bslbf uimsbf bslbf uimsbf bslbf bslbf uimsbf

表 16 （续）

语 法	位数	助记符
<pre> If (P-STD_buffer_flag == '1'){ '01' P-STD_buffer_scale P-STD_buffer_size } If (PES_extension_flag_2 == '1'){ marker_bit PES_extension_field_length for(i=0;i<PES_extension_field_length;i++){ Reserved } } for (i=0;i<N1;i++) { stuffing_byte } for (i=0;i<N2;i++){ PES_packet_data_byte } } else if (stream_id == program_stream_map stream_id == private_stream_2 stream_id == ECM stream_id == EMM stream_id == program_stream_directory stream_id == DSMCC_stream stream_id == ITU-T Rec. H.222.1 type E stream){ for (i=0;i<PES_packet_length;i++){ PES_packet_data_byte } } else if (steam_id == padding_stream){ for (i=0;i<PES_packet_length;i++){ </pre>	<pre> 2 1 13 1 7 8 8 8 </pre>	<pre> bslbf bslbf uimsbf bslbf uimsbf bslbf bslbf bslbf </pre>

表 16 （续）

语 法	位数	助记符
padding_byte } } }	8	bslbf

6.3.7 PES 分组包中各字段的语义定义

分组起始码前缀字段 packet_start_code_prefix

24 位代码，它和后面的 stream_id 构成了标识分组开始的分组起始码。它是一个值为 ‘0000 0000 0000 0000 0000 0001’ (0x000001) 的位串。

流标识字段 stream_id

在节目流中，它规定了基本流的号码和类型。定义见表 17。在传输流中，它可以被设定为正确描述表 17 中定义的基本流类型的任何有效值。在传输流中，基本流类型在 6.5 的节目特定信息中作了规定。

表 17 Stream_id 赋值

stream_id	注的编号	流 编 码
1011 1100	1	program_stream_map
1011 1101	2	private_stream_1
1011 1110		padding_stream
1011 1111	3	private_stream-2
110x xxxx		GB/T 17975.3-2002、GB/T 20090.3或GB/T 17191.3-1997音频流编号x xxxx
1110 xxxx		GB/T 17975.2-2000、GB/T 20090.2-2006或GB/T 17191.2-1997视频流编号xxxx
1111 0000	3	ECM_stream
1111 0001	3	EMM_stream
1111 0010	5	GB/T 17975.1、本部分附录B或 ISO/IEC13818-6_DSMCC_stream
1111 0011	2	ISO/IEC_13522_stream
1111 0100	6	ITU-T Rec. H.222.1类型A
1111 0101	6	ITU-T Rec. H.222.1类型B
1111 0110	6	ITU-T Rec. H.222.1类型C
1111 0111	6	ITU-T Rec. H.222.1类型D
1111 1000	6	ITU-T Rec. H.222.1类型E
1111 1001	7	ancillary_stream
1111 1010…1111 1110		保留数据流
1111 1111	4	program_stream_directory

表 17 （续）

符号x表示值'0'或'1'均被允许且可产生相同的流类型。流号码由x的取值决定。
注1：类型为program_stream_map的PES分组包有唯一的语法，在7.4.1 中作了规定。
注2： 类型为private_stream_1和ISO/IEC_13352_stream的PES分组包与GB/T 17975.2-2000、GB/T 20090.2-2006视频流及GB/T 17975.3-2002、GB/T 20090.3音频流服从相同的PES分组包语法。
注3：类型为private_stream_2,ECM_stream和EMM_stream的PES分组包与private_stream_1相似，除了在PES_packet_length字段后未规定语法。
注4：类型为program_stream_directory的PES分组包有唯一的语法，在7.5 中作了规定。
注5：类型为DSM_CC_stream的PES分组包有唯一的语法，在 ISO/IEC13818-6中作了规定。
注6： stream_id与表 30中的stream_type 0x09相关联。
注7： stream_id仅用于PES分组包。PES分组包在传输流中携带了来源于节目流或GB/T 17191.1-1997系统流的数据(见6.3.7)。

PES 分组包长度字段 PES_packet_length

16 位字段，指出了 PES 分组包中跟在该字段后的字节数目。值为 0 表示 PES 分组包长度要么没有规定要么没有限制。这种情况只允许出现在 PES 分组包的有效负载是基本视频流的数据，且 PES 分组包作为有效负载包含于传输流包中。

在 PES_packet_length 为 0 时，为避免 PES 分组理解的二义性， PES 分组的最后 24 位不能为 0x00 00 01， 否则可以通过增加或减少一个字节实现。

PES 加扰控制字段 PES_scrambling_control

2 位字段，表示 PES 分组包有效负载的加扰方式。当加扰发生在 PES 层， PES 分组包头，如果有可选字段的话也包括在内，不应被加扰。见表 18。

表 18 PES 加扰控制值

值	描 述
00	非加扰
01	用户定义
10	用户定义
11	用户定义

PES 优先级字段 PES_priority

1 位字段，指示 PES 分组包中有效负载的优先级。'1'表示 PES 分组包中有效负载的优先级高于该字段为'0'的 PES 分组包有效负载。多路复用器能使用该字段来区分安排基本流中数据的优先级。传输机制不应改动该字段。

数据对齐指示符字段 data_alignment_indicator

1 位标志：

对于非 AVS 视频流如果有 8.10 中的 data_stream_alignment_descriptor 描述子且 data_alignment_indicator 置'1'时，表示 PES 分组包头后紧跟着的 data_stream_alignment_descriptor 所指出的视频起始码或音频同步字。若其值为'1'且无该描述子，则需要在表 49 和表 50 中 alignment_type '01'所表示的对齐。当值为'0'时，没有定义是否有任何这样的对齐。

对于 AVS 视频流如果有 8.10 中的 data_stream_alignment_descriptor 描述子且 data_alignment_indicator 置'1' 时，表示 PES 分组包头后紧跟着 data_stream_alignment_descriptor 所指出的视频起始码或音频同步字。若其值为'1'且无该描述子，则需要在表 49 中 alignment_type '01'所表示的对齐。当值为'0'时，没

有定义是否有任何这样的对齐。

版权字段 `copyright`

1 位字段。置‘1’时表示相关 PES 分组包有效负载的材料受到版权保护。当值为‘0’时，没有定义该材料是否受到版权保护。8.24 中描述的版权描述子与包含 PES 分组包的基本流相关。若描述子作用于包含 PES 分组包的材料，则版权标志被置为‘1’。

原始或拷贝字段 `original_or_copy`

1 位字段。置‘1’时表示相关 PES 分组包有效负载的内容是原始的；值为‘0’表示相关 PES 分组包有效负载的内容是一份拷贝。

PTS DTS 标志字段 `PTS_DTS_flags`

2 位字段。当值为‘10’时，PTS 字段应出现在 PES 分组包头中；当值为‘11’时，PTS 字段和 DTS 字段都应出现在 PES 分组包头中；当值为‘00’时，PTS 字段和 DTS 字段都不出现在 PES 分组包头中。值‘01’是不允许的。

ESCR 标志字段 `ESCR_flag`

1 位标志。置‘1’时表示 ESCR 基础和扩展字段出现在 PES 分组包头中；值为‘0’表示没有 ESCR 字段。

ES 速率标志字段 `ES_rate_flag`

1 位标志。置‘1’时表示 ES_rate 字段出现在 PES 分组包头中；值为‘0’表示没有 ES_rate 字段。

DSM 特技方式标志字段 `DSM_trick_mode_flag`

1 位标志。置‘1’时表示有 8 位特技方式字段；值为‘0’表示没有该字段。

附加版权信息标志字段 `additional_copy_info_flag`

1 位标志。置‘1’时表示有附加拷贝信息字段；值为‘0’表示没有该字段。

PES CRC 标志字段 `PES_CRC_flag`

1 位标志。置‘1’时表示 CRC 字段出现在 PES 分组包头中；值为‘0’表示没有该字段。

PES 扩展标志字段 `PES_extension_flag`

1 位标志。置‘1’时表示 PES 分组包头中有扩展字段；值为‘0’表示没有该字段。

PES 头数据长度字段 `PES_header_data_length`

8 位字段。指出包含在 PES 分组包头中的可选字段和任何填充字节所占用的总字节数。该字段之前的字节指出了有无可选字段。

标记位字段 `marker_bit`

值为‘1’的 1 位字段。

展现时间戳字段 `PTS`

展现时间与解码时间的关系如下：PTS 是一个编码在三个分离字段中的 33 位数字。它指出了基本流 n 的第 k 个展现单元在系统目标解码器中的展现时间 $tp_n(k)$ 。PTS 的值以系统时钟频率的 $1/300$ (即 90 kHz) 为单位。展现时间由 PTS 根据式(11)计算而来。在对展现时间戳进行编码时频率的约束见 9.4。

$$PTS(k) = ((system_clock_frequency \times tp_n(k)) \text{ DIV } 300) \% 2^{33} \quad (11)$$

其中， $tp_n(k)$ 是展现单元 $P_n(k)$ 的展现时间。

对音频而言，若 PES 分组包头中有 PTS，则它是指 PES 分组包中开始的第一个访问单元。若 PES 分组包中有音频访问单元的首字节，则有一个音频访问单元开始于该 PES 分组包中。

对视频而言，若 PES 分组包头中有 PTS，则它是指包含于 PES 分组包中开始的第一个图像起始码的访问单元。若 PES 分组包中有图像起始码的首字节，则有一个图像起始码开始于该 PES 分组包中。

对音频展现单元(PU)，`low_delay` 序列中的视频 PU 以及 B 图像，展现时间 $tp_n(k)$ 应等于 $td_n(k)$ 。

对于非 `low_delay` 中的 I 图像和 P 图像，在访问单元(AU) k 和 k' 之间无解码不连续时，展现时间 $tp_n(k)$ 应等于下一个传输的 I 图像或 P 图像的解码时间 $td_n(k)$ (见 9.5)。若有解码不连续或流终止，则 $tp_n(k)$ 和 $td_n(k)$ 之间的差别应与初始流一直延续，没有不连续也没有终止时完全相同。

注 1：`low_delay` 序列是 `low_delay` 标志被设置的视频序列(见 GB/T 20090.2-2006)。

若音频中有滤波,则系统模型假定滤波不会导致延迟。因此,编码时 PTS 所涉及的采样与解码时 PTS 所涉及的采样是相同的。

ESCR 字段 ESCR_base, ESCR_extension

42 位字段,分两部分编码。第一部分是一个长度为 33 位的字段,其值 ESCR_base(i)由式(14)给出;第二部分是一个长度为 9 位的字段,其值 ESCR_ext(i)由式(15)给出。ESCR 字段指出了基本流中包含 ESCR_base 最后一个比特的字节到达 PES-STD 输出端的期望时间(见 7.2.4)。

特别地

$$\text{ESCR}(i) = \text{ESCR_base}(i) \times 300 + \text{ESCR_ext}(i) \quad (13)$$

式中:

$$\text{ESCR_base}(i) = ((\text{system_clock_frequency} \times t(i)) \text{ DIV } 300) \% 2^{33} \quad (14)$$

$$\text{ESCR_ext}(i) = ((\text{system_clock_frequency} \times t(i)) \text{ DIV } 1) \% 300 \quad (15)$$

ESCR 和 ES_rate 字段(参见下面紧接的语义)包含与 PES 流序列相关的时间信息。这些字段应满足 9.3 中定义的约束。

基本流速率字段 ES_rate

22 位无符号整数。对于 PES 流而言,它指出了系统目标解码器接收 PES 分组包的速率。该字段在它所属的 PES 分组包以及同一个 PES 流的后续 PES 分组包中一直有效,直到遇到一个新的 ES_rate 字段。该字段的值以 50 字节/秒为单位,且不能为 0。该字段用于定义 PES 流的字节到达 P-STD 输入端的时间(见 9.4 中的定义)。在各个 PES 分组包中,编码在该字段中的值可能不同。

特技方式控制字段 trick_mode_control

3 位字段。它表示作用于相关视频流的特技方式。对其它类型的基本流,该字段及其后 5 位的含义没有定义。trick_mode 状态的定义见 6.2.3 的特技模式部分。特技方式控制值见表 20。

当 trick_mode 状态为假时,对 GB/T 17975.2-2000 或 GB/T 20090.2-2006 视频而言,解码过程输出渐进序列中每幅图像的帧数 N 由 repeat_first_field 和 top_field_first 字段来规定。对 GB/T 17191.2-1997 视频而言,由序列头决定。

对于隔行序列,当 trick_mode 状态为假时,对 GB/T 20090.2-2006 视频而言,帧数 N 由 repeat_first_field 和 progressive_frame 字段来规定。

当 trick_mode 状态为真时,图像的播放帧数依赖于值 N。

表 19 特技方式控制值

值	描 述
000	快进

表 19 (续)

值	描 述
001	慢动作
010	冻结帧
011	快倒
100	慢倒
101~111	保留

当该字段值发生变化或特技模式操作停止时，可能会出现下列情况的任意组合：

时基不连续；

解码不连续；

连续性计数器不连续。

在特技模式的情况下，解码和展现的非标准速度可能会导致视频基本流数据中定义的某些字段值不正确。同样，条带结构的语义约束也可能无效。这些例外所涉及的视频语法元素为：

bit_rate;

bbv_delay;

repeat_first_field;

v_axis_positive;

field_sequence;

subcarrier;

burst_amplitude;

subcarrier_phase。

在特技模式中，解码器不应该依赖于编码在这些字段中的值。

标准并不要求解码器能解码 `trick_mode_control` 字段。但是，能解码该字段的解码器应能满足以下要求：

快进 fast forward

`trick_mode_control` 字段中的值 ‘000’。当该值出现时，它表示一个快进视频流并定义了 PES 分组包头中后续 5 位的含义。`intra_slice_refresh` 位可以被设定为‘1’以指出可能有丢失的宏块。解码器可以用前一个解码图像中相同位置的宏块来代替。表 20 中定义的 `field_id` 字段，表示应该显示哪个或哪些场。`frequency_truncation` 字段指出了可能包括的一个系数受限集合。该字段值的含义如表 21 所示。

慢动作 slow motion

`trick_mode_control` 字段中的值 ‘001’。当该值出现时，它表示一个慢动作视频流，并定义了 PES 分组包头中后续 5 个比特的含义。对渐进序列而言，该图像应被显示 $N \times \text{rep_cntrl}$ 时间，其中 N 定义如上。对 GB/T 17191.2-2000 视频和 GB/T 20090.2-2006 视频渐进序列而言，该图像应被显示 $N \times \text{rep_cntrl}$ 时间，其中 N 定义如上。

对 GB/T 20090.2-2006 隔行序列而言，该图像应被显示 $N \times \text{rep_cntrl}$ 时间。若该图像是一个帧图像，则待显示的第一个场在 `top_field_first` 为 1 时应该是顶场，在 `top_field_first` 为 0 时，应该是底场(参见 GB/T 20090.2-2006)。该场被显示 $N \times \text{rep_cntrl} / 2$ 时间。该图像的其它场被显示 $N - N \times \text{rep_cntrl} / 2$ 时间。

冻结帧 freeze frame

`trick_mode_control` 字段中的值 ‘010’。当该值出现时，它表示冻结帧视频流，并定义了 PES 分组包头中后续 5 位的含义。表 20 中定义的 `field_id` 字段，表示应该显示哪个(些)场。`field_id` 字段指出了包含该字段的 PES 分组包中开始的第一个视频访问单元，除非该 PES 分组包包含 0 个有效负载字节。在后一种情况下，`field_id` 字段指出了最近的前一个视频访问单元。

快倒 fast reverse

trick_mode_control 字段中的值 ‘011’。当该值出现时，它表示一个快倒视频流并定义了 PES 分组包头中后续 5 位的含义。intra_slice_refresh 位可以被设定为‘1’以指出可能有丢失的宏块。解码器可以用前一个解码图像中相同位置的宏块来代替。表 20 中定义的 field_id 字段，表示应该显示哪个或哪些场。frequency_truncation 字段指出了可能包括的一个系数受限集合。该字段值的含义如表 21“系数选择值”所示。

慢倒 slow reverse

trick_mode_control 字段中的值 ‘100’。当该值出现时，它表示一个慢倒视频流并定义了 PES 分组包头中后续 5 位的含义。对 GB/T 17191.2-1997 视频和 GB/T 20090.2-2006 视频渐进序列而言，该图像应被显示 $N \times \text{rep_cntrl}$ 时间，其中 N 定义如上。

对 GB/T 20090.2-2006 隔行序列而言，该图像应被显示 $N \times \text{rep_cntrl}$ 时间。若该图像是一个帧图像，则待显示的第一个场在 top_field_first 为 1 时应该是底场，在 top_field_first 为‘0’时，应该是顶场(参见 GB/T 20090.2-2006)。该场被显示 $N \times \text{rep_cntrl} / 2$ 时间。该图像的其它场被显示 $N \times \text{rep_cntrl} / 2$ 时间。

场标识字段 field_id

2 位字段，表示应该显示哪个(些)场。根据表 20 对其进行编码。

表 20 field_id 字段控制值

值	描 述
00	仅按顶场底场顺序播放
01	仅按底场顶场顺序播放
10	播放帧
11	保留

帧内编码条带更新字段 intra_slice_refresh

1 位标志。置‘1’时表示 PES 分组包的视频数据编码条带中可能有丢失的宏块；置‘0’时，表示上述情况可能不出现。更多的信息可参见 GB/T 20090.2。解码器可以用前一个解码图像中同一个位置的宏块来代替丢失的宏块。

频率截断字段 frequency_truncation

2 位字段。指出在对 PES 分组包中数据进行编码时可能用到受限系数集合。其值定义于表 21。

表 21 系数选择值

值	描述
00	仅DC系数非0
01	仅前三个系数非0
10	仅前六个系数非0
11	所有系数均可能非0

显示次数控制字段 rep_cntrl

5 位字段，指出隔行图像中每一场的显示次数或渐进图像显示次数。对隔行图像而言，顶场或底场是否应首先显示是视频序列头中 trick_mode_control 字段和 top_field_first 字段的功能。该字段值不能为‘0’。

附加版权信息字段 additional_copy_info

7 位字段，包含与版权信息有关的私有数据。

前 PES 分组包 CRC 字段 previous_PES_packet_CRC

16 位字段。在对前一个 PES 分组包(不包括该 PES 分组包头)进行处理后,该字段包含一个在解码器的 16 个寄存器中生成 0 输出的 CRC 值。该 CRC 值与附录 A 中所定义的相类似,但具有以下多项式:

$$x^{16}+x^{12}+x^5+1$$

注 2: 该 CRC 值是为了用于网络维护,例如将有间隙性错误的源隔离开来,而不是为了供基本流解码器使用。它仅用于计算数据字节,因为在传输过程中 PES 分组包头数据可能被修改。

PES 私有数据标志字段 PES_private_data_flag

1 位标志。置'1'时表示 PES 分组包头中包含私有数据;置'0'时表示 PES 分组包头中无私有数据。

分组头字段标志字段 pack_header_field_flag

1 位标志。置'1'时表示当前 PES 分组包头中有 GB/T 17191.2-1997 pack 分组头或节目流 pack 分组头。若该字段在包含于节目流中的 PES 分组包中,其值应为'0'。在传输流中,当值为'0'时表示 PES 头中无 pack 分组头。

节目分组包序列计数标志字段 program_packet_sequence_counter_flag

1 位标志。值为'1'时表示 PES 分组包有 program_packet_sequence_counter, MPEG1_MPEG2_identifier 和 original_stuff_length 字段。值为'0'时表示 PES 分组包头中无这些字段。

P-STD 缓冲区标志字段 P-STD_buffer_flag

1 位标志。置'1'时表示 PES 分组包头中有 P-STD_buffer_scale 和 P-STD_buffer_size 字段。值为'0'时表示 PES 头中无这些字段。

PES 扩展标志字段 PES_extension_flag_2

1 位标志,置'1'时表示有 PES_extension_field_length 及相关字段。

PES 私有数据字段 PES_private_data

16 位字段。包含私有数据。这些数据与其前后的字段组合在一起时,不能与 packet_start_code_prefix (0x000001)冲突。

分组字段长度字段 pack_field_length

8 位字段。表示 pack_header_field()以字节为单位时的长度。

节目分组包序列计数字段 program_packet_sequence_counter

7 位字段。它是一个可选的计数器,随着来自于节目流或 GB/T 17191.1-1997 流的每一个后续 PES 分组包或传输流中具有单个节目定义的 PES 分组包而递增,以提供与连续性计数器(见 6.3.2)相似的功能。它能用于检索节目流或原始 GB/T 17191.1-1997 流中的初始 PES 分组包序列。该计数器在达到最大值后回卷为 0。PES 分组包不能出现重复。因此,复用节目中任何两个连续的 PES 分组包不应具有相同的 program_packet_sequence_counter 值。

MPEG1 MPEG2 标识符字段 MPEG1_MPEG2_identifier

1 位标志。置'1'时表示 PES 分组包携带的信息来自于 GB/T 17191.1-1997 流;置'0'时表示 PES 分组包携带的信息来自于节目流。

初始填充长度字段 original_stuff_length

6 位字段。指定用于初始 GB/T 20090.1 分组头或初始 GB/T 17191.1-1997 分组头中的填充字节数。

P-STD 缓冲区比例字段 P-STD_buffer_scale

1 位字段。仅当该 PES 分组包包含于节目流中时才有意义。它指出了用来解释后续 P-STD_buffer_size 字段的比例因子。若前面的 stream_id 表示一个音频流,该字段值应为'0';若前面的 stream_id 表示一个视频流,该字段值应为'1'。对于所有的其它流类型,其值可以为'0'或'1'。

P-STD 缓冲区大小字段 P-STD_buffer_size

13 位无符号整数。仅当该 PES 分组包包含于节目流中时才有意义。它定义了 P-STD 输入缓冲区的大小 BS_n 。若 P-STD_buffer_scale 的值为'0',那么 P-STD_buffer_size 以 128 字节为单位来度量缓冲区的大小。若 P-STD_buffer_scale 的值为'1',那么 P-STD_buffer_size 以 1024 字节为单位来度量缓冲区的大小。因此:

if (P-STD_buffer_scale == 0)

$BS_n = P\text{-}STD_buffer_size \times 128$ (16)

else

$BS_n = P\text{-}STD_buffer_size \times 1024$ (17)

当该字段被 GB/T 17975.1-2000 或 GB/T 20090.1 系统目标解码器收到后，其编码值立即生效。

PES 扩展字段长度字段 PES_extension_field_length

7 位字段。指出了跟在该字段之后在 PES 扩展字段中直到且包括任何保留字节的数据的字节长度。

填充字节字段 stuffing_byte

8 位字段，其值恒定为 ‘1111 1111’。可以由编码器插入以满足信道的需求等。解码器丢弃该字段。一个 PES 分组包头中最多只能出现 32 个填充字节。

PES 分组包数据字节字段 PES_packet_data_byte

该字段应该是来自于由分组的 stream_id 或 PEID 所指定的基本流的连续数据字节。当基本流数据符合 GB/T 17975.2-2000、GB/T 20090.2-2006 或 GB/T 17975.3-2002、GB/T 20090.3 时，该字段应该是与本部分的字节相对齐的字节。基本流的字节顺序应得到保持。该字段的字节数 N 由 PES_packet_length 字段规定。N 应等于 PES_packet_length 减去在 PES_packet_length 字段的最后一个字节与第一个 PES_packet_data_byte 间的字节数。

补充字节字段 padding_byte

8 位字段，其值恒定为 ‘1111 1111’。该字段被解码器丢弃。

6.3.8 在传输流中携带节目流和 GB/T 17191.1-1997 系统流

传输流含有可选字段，从而以一种允许在解码器中简单地重建各个流的方式支持携带节目流及 GB/T 17191.1-1997 系统流。

当把一个节目流放到传输流中时，传输流包中携有 stream_id 值为 private_stream_1，GB/T 20090.2-2006 或 GB/T 17191.2-1997 视频和 GB/T 20090.3 或 GB/T 17191.3-1997 音频的节目流 PES 分组包。

对于这些 PES 分组包，当在传输流解码器中重构节目流时，PES 分组包数据被复制到正在被重建的节目流中。

对于 stream_id 值为 program_stream_map、padding_stream、private_stream_2、ECM、EMM、DSM_CC_stream 或 program_stream_directory 的节目流 PES 分组包，除 packet_start_code_prefix 外的所有字节被放入一个新 PES 分组包的 data_bytes 字段中。新 PES 分组包的 stream_id 值为 ancillary_stream (见表 17)。这样，这个新的 PES 分组包就被携带在传输流包中。

在传输流解码器中重建节目流时，对含 stream_id 为 ancillary_stream_id 的 PES 分组包，packet_start_code_prefix 被写入到正在重建的节目流中，其后跟有来自这些传输流 PES 分组包的 data_byte 字段。

GB/T 17191.1-1997 流首先通过以 GB/T 20090.1 的 PES 分组包头来取代 GB/T 17191.1-1997 分组包头的方式携带于传输流中。GB/T 17191.1-1997 分组包头字段值被复制到等价的 GB/T 20090.1 分组包头字段中。

program_packet_sequence_counter 字段被包括在携带有来自节目流或 GB/T 17191.1-1997 系统流的数据的每个 PES 分组包头中。这使初始节目流或初始 GB/T 17191.1-1997 系统流中 PES 分组包的顺序在解码器中可重现。

节目流或 GB/T 17191.1-1997 系统流的 pack_header() 字段被携带在传输流中紧跟着的 PES 分组包头中。

6.4 节目成分信息表

节目成分信息表(PEIT)提供节目成分识别和解复用所必需的基本信息。每个 PEIT 表对应一个 PEIT 区

间。每个 PEIT 区间由分属不同节目成分的传输包构成。节目成分信息表的主要功能是对其所对应的区间内传输包序列进行描述。描述的内容主要包括:所对应的 PEIT 区间所选取的节目成分排序方式,各个节目成分计数信息,各个节目成分的 PEID,以及其它信息。一个 PEIT 表由单独一个 PEIT 段承载。

解复用端首先通过搜索带有 PEIT_indicator 标识符的传输包来寻找 PEIT 段。PEIT_indicator 的具体定义及其在码流中的位置在 6.3.2 中阐述。在重建并得到 PEIT 表之后,解复用端使用 PSI 对应的 PEID 做为标识符在 PEIT 区间内搜索所需要的 PSI 传输包并解析 PSI 信息。PSI 信息中包含节目基本流对应的 PEID 和其它有用信息。获得节目相关的信息后,系统就可以使用节目基本流对应的 PEID 做为标识符在 PEIT 区间内搜索所需要的基本流传输包并送到相应的基本流处理器中。PEID 的定义参见表 22。详细的解复用过程的说明参见附录 L。

为了优化在不同传输情况下的传输效率,本标准设计两种复用模式:基本模式(basic mode)和高级模式(advanced mode)。基本模式适用于低误码和包按序抵达的传输信道,比如数字有线电视广播信道等。高级模式适用于高丢包或高误码,以及有包乱序的传输信道,比如 IPTV 网络和移动电视信道。有关复用模式的说明参见附录 M。

表 22 PEID 的定义

值	描 述
0x0000	PAT
0x0001	CAT
0x0002	TSDT
0x0003~0x000E	保留
0x000F	PEIT_indicator
0x0010~0x1FFE	可以赋给network_PEID, Program_map_PEID, elementary_PEID或作其它用途
0x1FFF	空的分组

注： PEID值为0x0000, 0x0001, 和0x0010~0x1FFE的传输分组允许携带transport_rate。

6.4.1 PEIT 语法

语 法	位 数	助记符
program_element_information_section() {		
table id	8	uimsbf
section_syntax_indicator	1	bslbf
Reserved	3	bslbf
PEIT_section_length	12	uimsbf
Reserved	4	bslbf

PEIT_interval_pointer	9	uimsbf
advanced_mode_enable	1	bslbf
num_of_anchors	2	uimsbf
if(advanced_mode_enable) { PEIT_seq_num }	8	uimsbf
total_TP_cnt	8	uimsbf
total_element_cnt	8	uimsbf
for (i=0; i< total_element_cnt;i++){		
TP_cnt	8	uimsbf
Reserved	3	bslbf
PEID	13	uimsbf
}		
CRC_32	32	rpchof
}		

6.4.2 PEIT 段中各字段的语义定义

表标识字段 **table_id**
8 位字段。值为 0x40。

段语法指示符字段 **section_syntax_indicator**
1 位指示符。值为 0。

PEIT 段长度字段 **PEIT_section_length**
12 位字段。指示紧跟在 **private_section_length** 字段后直到 **private_section** 结束的私有段剩余字节数。该字段值不应超过 4093(0xFFD)。

PEIT 区间指针字段 **PEIT_interval_pointer**
9 位字段。表示当前 PEIT 表的第一个传输包与当前 PEIT 表对应的 PEIT 区间的第一个传输包之间间隔的传输包个数，范围为 0-511。

高级模式启动字段 **advanced_mode_enable**
1 位标志。表示当前 PEIT 区间的复用模式.如果设置为'1',那么当前 PEIT 区间为高级复用模式,否则为基本复用模式.

定位数目字段 **num_of_anchors**
2 位字段。表示当前 PEIT 区间内传输包被链接的 AP 的数目。选取范围为 1-4。

PEIT 序列号字段 PEIT_seq_num
8 位字段。表示 PEIT 表的序列号,选取范围为 0-255.

传输包总数字段 total_TP_cnt
8 位字段。表示当前 PEIT 区间一共包含多少传输包. 范围为 0-255。

节目成分总数字段 total_element_cnt
8 位字段。表示当前 PEIT 表包含多少节目项，范围为 0-255. 不同节目项可以是同一节目。

传输包计数字段 TP_cnt
8 位字段。表示当前同一节目成分连续出现的个数,如果该成分在 PEIT 表中相应的节目项中列出,但是并没有出现,那么这个值为 0.

节目成分标识符字段 PEID
13 位字段。表示节目成分的标识符. 具体的定义参见表 22.

CRC 32 字段 CRC_32
32 位字段，它包含 CRC 值以在处理完整个私有段后在附录 A 中定义的解码器寄存器产生 0 输出值。

6.5 节目特定信息

节目特定信息(PSI)既包括 GB/T 20090.1 标准数据，也包括私有数据以使解码器能分离节目。节目由一个或多个基本流组成，每一个均由 PEID 标识。节目基本流或其中的各部分可以被加扰以用于条件访问，但 PSI 不应被加扰。

在传输流中，PSI 被分为 4 个表结构，如表 23 所示。虽然这些结构可以被看作简单的表，但它们应被进一步划分为各个段并插入到传输流包中，一些带有预先规定的 PEID，另一些带有用户可选的 PEID。GB/T 20090.1 定义的 PSI 表应被划分为携带于传输分组包中的一个或多个段。一个段就是一个用于把每个 GB/T 20090.1 定义的 PSI 表映射到传输流包中的语法结构。

除 GB/T 20090.1 定义的 PSI 表外，还可携带私有数据表。本标准未定义将私有信息携带于传输流包中的方法。可以用与携带 GB/T 20090.1 定义的 PSI 表相同的方式来对其进行结构化，从而，用来映射该私有数据的语法与用来映射 GB/T 20090.1 定义的 PSI 表的语法是一致的。为此定义了一个私有段。如果携带私有数据的传输流包和携带节目映射表的传输流包具有相同的 PEID，那么将使用 private_section 语法和语义。携带于 private_data_byte 中的数据可以被加扰，但 private_section 中其余字段不应被加扰。该 private_section 允许数据以最小结构传送。在不使用这一结构时，本部分未对传输流包中私有数据的映射作出定义。

表 23 节目特定信息

结构名	流类型	PEID值	描 述
节目关联表	GB/T 20090.1	0x0000	将节目号码与节目映射表PEID相关联
节目映射表	GB/T 20090.1	其值在PAT中指出	指定一个或多个节目成分的PEID值
网络信息表	专用	其值在PAT中指出	诸如FDM频率，卫星转发器频点等物理网络参数
条件访问表	GB/T 20090.1	0x0001	将一个或多个(私有)EMM流分别与唯一的PEID值相关联

传输流描述表	GB/T 20090.1	0x0002	将一个或多个表 40 中的描述子与整个传输流相关联
--------	--------------	--------	---------------------------

各个段可能长度不一。每一段的开始由传输流包有效负载中的 `pointer_field` 来标识。该字段语法见表 25。

携带 PSI 段的传输流包中还可能包含适应字段。

在一个传输流中，值为 `0xFF` 的分组填充字节可能会出现在一个段的末字节后。在这种情况下，直到传输流包结束的所有后续字节也都将是值为 `0xFF` 的填充字节。这些字节可能被解码器丢弃。此时，有相同 PEID 值的下一个传输流包有效负载，应以值为 `0x00` 的 `pointer_field` 作为起始码，以指出紧跟其后开始的下一个段。

每一传输流都包含一个或多个 PEID 值为 `0x0000` 的传输流包。这些传输流包一起将包含一个完整的节目关联表，以给出该传输流中所有节目的一个完整列表。`current_next_indicator` 置 '1' 的表的最近传输版本，总是作用于传输流中当前数据。传输流中携带的节目的任何变动将在 PEID 值为 `0x0000` 的传输流中携带的节目关联表的最新版本中加以描述。这些段均用到值为 `0x00` 的 `table_id`。在 PEID 值为 `0x0000` 的传输流包中只允许 `table_id` 值为 `0x00` 的段。要使节目关联表的新版本生效，所有带新 `version_number` 及 `current_next_indicator` 置 '1' 的段(在 `last_section_number` 中指出)必须退出 T-STD 中定义的 B_{sys} (见 6.2)。当用于结束表的那个段的最后一个字节离开 B_{sys} 后，节目关联表开始生效。

一旦传输流中一个或多个基本流被加扰，就要传送一个 PEID 值为 `0x0001` 的且包含一个完整的条件访问表的传输流包。表中包含与加扰流相关联的 `CA_descriptor`。传输的传输流包共同组成一个条件访问表的完整版本。`current_next_indicator` 置 '1' 的表的最近传送版本总是作用于传输流中的当前数据。在加扰中使现有表无效或不完整的任何变动应在条件访问表的一个最新版本中加以描述。这些段将均使用值为 `0x01` 的 `table_id`。在 PEID 值为 `0x0001` 的传输流包中只允许 `table_id` 值为 `0x01` 的段。要使 CAT 的新版本生效，所有带新 `version_number` 及 `current_next_indicator` 置 '1' 的段(在 `last_section_number` 中指出)必须退出 T-STD 中定义的 B_{sys} (见 6.2)。当用于结束表的那个段的最后一个字节离开 B_{sys} 后，CAT 开始生效。

每个传输流要包含 PEID 值在节目关联表中被标记为包含传输流节目映射段的传输流包的一个或多个传输流包。节目关联表中列出的每个节目在一个唯一的 `TS_program_map_section` 中描述。每个节目将在传输流自身中完全定义。在合适的节目映射表段中有关联 `elementary_PEID` 字段的私有数据是该节目的一段。其它私有数据可能存在于传输流中但未在节目映射表中列出。当 `current_next_indicator` 置 '1' 的 `TS_program_map_section` 的最近传送版本将总是作用于传输流的当前数据，传输流中携带的任何节目的定义的任何改动应在携带于传输流包中的节目映射表的相应段的最新版本中加以描述。该传输流包的 PEID 值被标识为用于那个特定节目的 `program_map_PEID`。在节目连续存在期间，包括它的相关事件，该 `program_map_PEID` 不应改变。节目定义不应该跨过多于一个的 `TS_program_map_section`。在具有一个新 `version_number` 且 `current_next_indicator` 为 '1' 的段的最后一个字节退出 B_{sys} 时，`TS_program_map_section` 的新版本开始生效。

`table_id` 值为 `0x02` 的段应包含节目映射表信息。这些段可能携带于有不同 PEID 值的传输流包中。

网络信息表是可选的，其内容也是专用的。如果存在的话，它将携带于一些传输流包中。这些传输流包有相同 PEID 值，称为 `network_PEID`。`network_PEID` 值由用户定义，且如果存在，将出现在节目关联表中，在保留 `program_number 0x0000` 之下。若存在网络信息表，它将以一个或多个 `private_section` 的形式出现。

GB/T 20090.1 定义的 PSI 表的最大字节数为 1024 字节，`private_section` 最大字节数为 4096 字节。

传输流描述表是可选的。如果存在该表，则它将被封装在 PEID 值为 2 的传输流包中，该表将作用于整个传输流。传输流描述段的 `table_id` 将使用表 24 中规定的 `0x03`。其内容限制为表 40 中规定的描述子。当存在于传输流描述段中表的最后一个字节存在于 B_{sys} 中的时候，该传输流描述段有效。

不论是本部分定义的还是专用的，对 PSI 数据中的起始码、同步字节或其它位模式的出现没有限制。关于如何使用 PSI 函数的说明信息参见附录 C。

6.5.1 指针

pointer_field 语法定义见表 25。

表 25 节目特定信息指针

语 法	位数	助记符
pointer_field	8	uimsbf

6.5.2 指针语法中各字段的语义定义

指针字段 pointer_field

8 位字段，其值表示紧跟其后的直到出现在传输流包的有效负载中的第一个段的首字节为止的字节数（因此值 0x00 表示该段紧跟在 pointer_field 之后开始）。当至少有一个段开始于给定的传输流包中时，payload_unit_start_indicator(见 6.3.2)应置为 1 且传输流包的有效负载的首字节应包含该指针。当没有段开始于给定的传输流包中时，payload_unit_start_indicator 应置为 0 且传输流包的有效负载将不传送指针。

6.5.3 节目关联表

节目关联表给出了携带节目定义的传输流包中 program_number 与该传输流包的 PEID 值的对应关系。program_number 是与一个节目相关联的数值标号。

整个表包含于具有下列语法的一个或多个段之中。它可以被分割以占用多个段。见表 26。

表 26 节目关联段

语 法	位数	助记符
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_PEID	13	uimsbf
}		

<pre>else { Program_map_PEID } } CRC_32 }</pre>	<p>13</p> <p>32</p>	<p>uimsbf</p> <p>rpchof</p>
---	---------------------	-----------------------------

6.5.4 table_id 赋值

table_id 字段标识了传输流 PSI 段的内容，如表 27 所示。

表 27 table_id 赋值的值

值	描 述
0x00	program_association_section
0x01	conditional_access_section(CA_section)
0x02	TS_program_map_section
0x03	TS_description_section
0x04~0x3F	在GB/T 17975.1-2000中定义
0x40	Program_element_information_section
0x41~0xFE	用户私有
0xFF	禁止

6.5.5 节目关联段中各字段的语义定义

表标识字段 table_id

8 位字段，值为 0x00。如表 27 所示。

段语法指示符字段 section_syntax_indicator

1 位字段，值为'1'。

段长度字段 section_length

12 位字段，前两位为 '00'，后 10 位规定紧跟在该字段之后且包括 CRC 的该段的字节数目。其值不能超过 1021(0x3FD)。

传输流标识字段 transport_stream_id

16 位字段，用作标识以识别来源于网络中任何其它复用流的传输流。其值由用户定义。

版本号字段 version_number

5 位字段。它是整个节目关联表的版本号码。一旦节目关联表发生变化，该字段将递增 1，并对 32 取模。在 current_next_indicator 为'1'时，版本号应该是当前适用的节目关联表的版本号；在 current_next_indicator 为'0'时，版本号应该是下一个适用的节目关联表的版本号。

当前下一个指示符字段 current_next_indicator

1 位指示符。置'1'时表示传送的节目关联表是当前适用的；置'0'时表示传送的节目关联表还不适用，它是下一个生效的表。

段号字段 section_number

8 位字段，给出了该段的号码。节目关联表中第一个段的 section_number 应为 0x00。节目关联表每增加一个段，它将递增 1。

末段号字段 last_section_number

8 位字段，给出整个节目关联表中最后一段(即有最高 section_number)的号码。

节目号字段 program_number

16 位字段，规定 program_map_PEID 可适用的节目。当值为 0x0000 时，其后的 PEID 参照将是网络 PEID。该字段的其它值由用户定义。在节目关联表的一个版本中，该字段取任何单个值不能超过一次。

注： program_number 可作为一个指示符号，例如用于广播信道。

网络 PEID 字段 network_PEID

13 位字段。仅与值为 0x0000 的 program_number 一起使用，以指定包含网络信息表的传输流包的 PEID。该字段值由用户定义，但只能取表 22 中规定的值。该字段的出现是可选的。

节目映射 PEID 字段 program_map_PEID

13 位字段。它表示包含可作用于 program_number 所指定节目的 program_map_section 的传输流包的 PEID。任何 program_number 都不应该有多个 program_map_PEID 值。program_map_PEID 值由用户定义，但只能取表 22 中规定的值。

CRC 32 字段 CRC_32

32 位字段，它包含 CRC 值以在处理完整个节目关联段后在附录 A 中定义的解码器寄存器产生 0 输出值。

6.5.6 条件访问表

条件访问表(CA)给出了一个或多个 CA 系统、它们的 EMM 流和任何与它们相关的特定参数之间的关系。表 28 中 descriptor()字段的定义见 8.16 。
该表包含在具有以下语法的一个或多个段之中。它可以被分割以占用多个段。

表 28 条件访问段

语 法	位数	助记符
CA_section() { table_id section_syntax_indicator ‘0’ Reserved section_length Reserved version_number current_next_indicator section_number last_section_number for(i=0;i<N;i++) { descriptor() } CRC_32 }	 8 1 1 2 12 18 5 1 8 8 32	 uimbsbf bslbf bslbf bslbf uimbsbf bslbf uimbsbf uimbsbf rpchof

6.5.7 条件访问段中各字段的语义定义

表标识字段 table_id

8 位字段。表 27 中规定其值为 0x01。

段语法指示符字段 section_syntax_indicator

1 位字段，值为‘1’。

段长度字段 section_length

12 位字段，前两位为 ‘00’，后 10 位规定其后紧跟的段(包括 CRC)的字节数。该字段值不应超过 1021(0x3FD)。

版本号字段 version_number

5 位版本号。它是整个条件访问表的版本号码。一旦条件访问表发生变化，该字段将递增 1，并对 32 取模。在为'1'时，版本号应该是当前适用的条件访问表的版本号；在 current_next_indicator 为'0'时，版本号应该是下一个适用的条件访问表的版本号。

当前下一个指示符字段 current_next_indicator

1 位指示符。置'1'时表示传送的条件访问表是当前适用的；置'0'时表示传送的条件访问表还不适用，它是下一个生效的表。

段号字段 section_number

8 位字段，给出了该段的号码。条件访问表中第一个段的 section_number 应为 0x00。条件访问表每增加一个段，它将递增 1。

末段号字段 last_section_number

8 位字段，给出整个条件访问表中最后一段(即有最高 section_number)的号码。

CRC 32 字段 CRC_32

32 位字段，它包含 CRC 值以在处理完整个条件访问段后在附录 A 中定义的解码器寄存器产生 0 输出值。

6.5.8 节目映射表

节目映射表给出了节目号码和组成它们的节目元素之间的映射。该映射的一个单一实例被称为“节目定义”。节目映射表是一个传输流中所有节目定义的完整集合。该表在传输流包中被传输，传输流包的 PEID 值由编码器选择。需要时可能用到多个 PEID。该表包含在具有以下语法的一个或多个段中。它可以被分割以占用多个段。在每一段中，段号码字段值应置 0。段由 program_number 字段标识。见表 29。可以在 2.6 中找到 descriptor()字段的定义。

表 29 传输流节目映射段

语 法	位 数	助记符
TS_program_map_section(){		
Table_id	8	uimsbf
section_syntax_indictor	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_indictor	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
Reserved	1	bsbf
clock_recovery_flag	2	bslbf
clock_recovery_PEID	13	uimsbf
Reserved	4	bsbf

program_info_length for (i=0;i<N;i++){ descriptor() }	12	uimsbf
for (i=0;i<N1;i++){		
stream_type	8	uimsbf
Reserved	3	bslbf
elementary_PEID	13	uimsbf
Reserved	4	bslbf
ES_info_length	12	uimsbf
For (i=0;i<N2;i++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

6.5.9 传输流节目映射段中各字段的语义定义

表标识字段 **table_id**

8 位字段，值为 0x02。如表 27 所示。

段语法指示符字段 **section_syntax_indicator**

1 位字段，值为'1'。

段长度字段 **section_length**

12 位字段，前两位为'00'，后 10 位规定紧跟在该字段之后且包括 CRC 的该段的字节数目。其值不能超过 1021(0x3FD)。

节目号字段 **program_number**

16 位字段，规定 **program_map_PEID** 可适用的节目。一个节目定义仅会包含于一个传输流节目映射段。这意味着一个节目定义不会长于 1016(0x3FD)。处理长度不够情况的方法参见附录 C。**program_number** 可作为广播信道的指示符号。通过描述属于一个节目的不同节目元素，来自于不同源(例如连续事件)的数据可以被连接在一起以形成使用一个 **program_number** 的流的连续集合。应用示例参见附录 C。

版本号字段 **version_number**

5 位版本号。它是传输流节目映射段的版本号码。一旦该部分中的信息发生变化，该字段将递增 1，并对 32 取模。版本号涉及单个节目的定义，从而涉及单个段。在 **current_next_indicator** 为'1'时，版本号应该是当前适用的传输流节目映射段的版本号；在 **current_next_indicator** 为'0'时，版本号应该是下一个适用的传输流节目映射段的版本号。

当前下一个指示符字段 **current_next_indicator**

1 位指示符。置'1'时表示传送的传输流节目映射段是当前适用的；置'0'时表示传送的传输流节目映射段还不适用，它是下一个生效的段。

段号字段 **section_number**

8 位字段，值应为 0x00。

末段号字段 last_section_number

8 位字段，值应为 0x00。

时钟恢复标志字段 clock_recovery_flag

2 位标志。置'11'时表示该节目包含 PCR 时钟恢复信息，置'00'时表示该节目包含传输率时钟恢复信息，置'10'时表示该节目既包含 PCR 时钟恢复信息，又包含传输率时钟恢复信息。'01'是保留值。

时钟恢复 PEID 字段 clock_recovery_PEID

13 位的字段，指出包含时钟恢复信息的传输流包的 PEID。这些传输流包包含对由 program_number 所指定的节目有效的时钟恢复信息。若没有时钟恢复信息与用于私有流的节目定义相关联，那么该字段取值应为 0x1FFF。对选择 PEID 值的限制见表中 transport_rate 的语义定义和表 22。

节目信息长度字段 program_info_length

12 位的字段。起始的两位应为 '00'。其余的 10 位规定了紧跟在该字段之后的描述子的字节数。

流类型字段 stream_type

8 位的字段。它指出了携带在分组中的节目元素的类型，这些分组的 PEID 由 elementary_PEID 所指出。表 30 中规定了 stream_type 的值。

注：对于除音频、视频和 DSM CC 以外的由本部分定义的数据类型，诸如节目流目录和节目流映射等，GB/T 17975.1-2000、GB/T 20090.1 辅助流是可以适用的。

表 30 流类型指定

值	描 述
0x00	保留
0x01	GB/T 17191.2-1997视频
0x02	GB/T 17975.2-2000视频或 GB/T 17191.2-1997受限参数视频流
0x03	GB/T 17191.3-1997音频
0x04	GB/T 17975.3-2002音频
0x05	GB/T 17975.1-2000 private_sections
0x06	包含私有数据的 GB/T 17975.1-2000 PES分组包
0x07	ISO/IEC 13522 MHEG
0x08	附录A—DSM CC
0x09	ITU-T Rec.H.222.1
0x0A	ISO/IEC 13818-6类型A
0x0B	ISO/IEC 13818-6类型B
0x0C	ISO/IEC 13818-6类型C
0x0D	ISO/IEC 13818-6类型D
0x0E	GB/T 17975.1-2000辅助
0x42	GB/T 20090.2-2006视频
0x43	GB/T 20090.3音频
0x44	GB/T 20090.1 private_sections

0x45	包含私有数据的 GB/T 20090.1 PES分组包
0x46	GB/T 20090.1辅助

表 30 （续）

值	描 述
0x47~0x7F	ISO/IEC 13818保留
0x80~0xFF	用户私有

基本 PEID 字段 elementary_PEID

13 位字段，指出了携带相关节目元素的传输流包的 PEID。

ES 信息长度字段 ES_info_length

12 位字段。起始两位应为 ‘00’，剩余的 10 位指出了紧跟在该字段之后的相关节目元素的描述子的字节数。

CRC 32 字段 CRC_32

32 位字段，它包含 CRC 值，以在处理完整个传输流节目映射段后在附录 A 中定义的解码器寄存器产生 0 输出值。

6.5.10 私有段的语法

当私有数据在以 PEID 值在节目关联表中被指定为节目映射表 PEID 的传输流包传送时，将使用 private_section。该段在使得解码器能对流进行分析时允许数据以最小结构传输。可以用两种方法来使用该段：若 section_syntax_indicator 置‘1’，将用到所有表公共的整个结构；若置‘0’，则仅有从 ‘table_id’ 到 ‘private_section_length’ 之间的字段服从公共结构语法和语义。private_section 其余字段可采取用户定义的任何形式。附录 H 中有该语法扩充使用的示例。

私有表可以由几个 private_section 组成，各个段有相同的 table_id。见表 31。

表 31 私有段

语 法	位 数	助记符
private_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
Reserved	2	bslbf
private_section_length	12	uimsbf
if (section_syntax_indicator == ‘0’){		
for (I=0;i<N;I++) {		
private_data_byte	8	bslbf
}		
}		
else {		
table_id_extension	16	uimsbf
Reserved	2	bslbf
version_number	5	uuimsbf

表 31（续）

语 法	位 数	助记符
current_next_indicator	1	bslbf
section_unmber	8	uimsbf
last_section_number	8	uimsbf
for (i=0;I<private_section_length-9;i++){ private_data_byte }	8	bslbf
CRC_32	32	rpchof
}		

6.5.11 私有段中各字段的语义定义

表标识字段 **table_id**

8 位字段。其值表示该段所属的私有表。只能使用在表 27 中被定义为‘用户私有’的值。

段语法指示符字段 **section_syntax_indicator**

1 位指示符。置'1'时表示 **private_section_length** 字段之后的私有段服从通用的段语法；置'0'时表示 **privata_data_byte** 紧跟在 **private_section_length** 字段之后。

私有指示符字段 **private_indicator**

1 位用户可定义的标志位。该字段在将来的标准中不会再定义。

私有段长度字段 **private_section_length**

12 位字段。指示紧跟在 **private_section_length** 字段后直到 **private_section** 结束的私有段剩余字节数。该字段值不应超过 4093(0xFFD)。

私有数据字节字段 **private_data_byte**

用户可定义的字段。该字段在将来的标准中不会再定义。

表标识扩展字段 **table_id_extention**

16 位字段。它的使用与值由用户定义。

版本号字段 **version_number**

5 位字段。它是私有段的版本号码。一旦私有段中携带的信息发生变化，该字段将递增 1，并对 32 取模。在 **current_next_indicator** 为'0'时，版本号应该是下一个适用的有相同 **section_number** 和 **table_id** 的私有段的版本号。

当前下一个指示符字段 **current_next_indicator**

1 位指示符。置'1'时表示传送的私有段是当前适用的；当该字段为'1'，版本号应该是当前适用的私有段的版本号。置'0'时表示传送的私有段还不适用，它是下一个生效的有相同 **section_number** 和 **table_id** 的私有段。

段号字段 **section_number**

8 位字段，给出了该段的号码。私有段中第一个段的 **section_number** 应为 0x00。私有段每增加一个段，它将递增 1。

末段号字段 **last_section_number**

8 位字段，指出了本段是其中一段的私有表的最后一段(即有最高 **section_number** 的段)的号码。

CRC 32 字段 **CRC_32**

32 位字段，它包含 CRC 值以在处理完整个私有段后在附录 A 中定义的解码器寄存器产生 0 输出值。
 \

7 节目流比特流的要求

7.1 节目流编码结构与参数

GB/T 17975.1-2000、GB/T 20090.1 节目流编码层允许将具有一个或多个基本流的一个节目组合成一个单一流。来自于各基本流的数据与那些允许节目中的基本流同步播放的信息复用一起。

一个节目流包含来自于一个节目的复用在一起的一个或多个基本流。基本流数据携带于 PES 分组包中，PES 分组包由分组头及其后的分组包数据组成。

PES 分组包头以一个 32 位起始码开始，它也标识了该分组数据属于哪个流(见表 17)。PES 分组包头可能仅含有展现时间戳(PTS)，或既有展现时间戳(PTS)又有解码时间戳(DTS)。PES 分组包头还含其它可选字段的标识位，分组数据包括来自一个基本流的数量可变的连续字节。

在节目流中，PES 分组包被组合在分组中。分组以分组头开始，其后跟有零个或多个 PES 分组包。分组头以 32 位 start_code 开始，用于存放时间信息和比特率信息。

节目流以一个可重复的系统头开始，系统头中含有流中定义的所有系统参数。

本部分未指定可用作条件访问系统的一部分的编码数据。但本部分为节目服务供应商提供了一些机制以传输和标识数据供解码器处理及正确地引用此处定义的数据。

7.2 节目流系统目标解码器

节目流的语义及对这些语义的限制要求对解码事件及其发生时刻进行精确定义。在本部分中用一个称为节目流系统目标解码器(P-STD)的假想的解码器来陈述这些所需的定义。

P-STD 是一个概念化模型，用来精确定义这些术语并用于在构造节目流期间为解码过程建模。P-STD 仅为此而定义。所描述的 P-STD 的体系结构和时钟并不排除各种具有不同结构或定时机制的解码器对节目流进行不间断地同步回放。

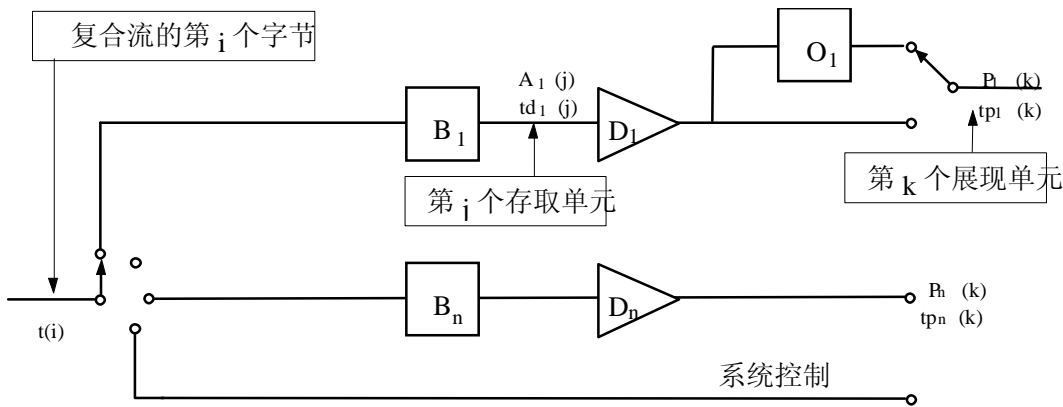


图 7 节目流系统目标解码器框图

以下记号用来描述节目流系统目标解码器。图 7 说明了其中的一部分。

- i,i' 节目流中的字节索引。第一字节的索引为 0。
- j 基本流中访问单元的索引。
- k, k', k'' 基本流中展现单元的索引。
- n 基本流的索引。

$t(i)$ 节目流中第 i 个字节进入系统目标解码器的时刻，以秒为单位表示。 $t(0)$ 的值是一个任意常数。

$SCR(i)$ 以 27MHz 系统时钟为单位来度量的 SCR 字段中的时间。这里的 i 指字段 `system_clock_reference_base` 最后一个字节的索引。

$A_n(j)$ 基本流 n 中第 j 个访问单元。注意， $A_n(j)$ 是按解码顺序给出索引的。

$td_n(j)$ 基本流 n 的第 j 个访问单元在系统目标解码器中的解码时间，该时间按秒度量。

$P_n(k)$ 基本流 n 中的第 k 个展现单元。 $P_n(k)$ 是按展现顺序进行索引的。

$tp_n(k)$ 基本流 n 的第 k 个展现单元在系统目标解码器中的展现时间，该时间按秒度量。

t 按秒度量的时间。

$F_n(t)$ 系统目标解码器中基本流 n 的输入缓冲区在时间 t 的占用度，按字节度量。

B_n 系统目标解码器中基本流 n 的输入缓冲区。

BS_n 系统目标解码器中基本流 n 的输入缓冲区的容量，按字节度量。

D_n 基本流 n 的解码器。

O_n 用于视频基本流 n 的重排序缓冲区。

7.2.1 系统时钟频率

P-STD 中引用的时间信息携带于本部分定义的一些数据字段中。在 6.3.6 及 7.3.3 中对这些字段作了定义。这些信息被编码为系统时钟的采样值。

系统时钟频率的值以 Hz 为单位且应满足以下约束：

$$27\,000\,000 - 810 \leq \text{system_clock_frequency} \leq 27\,000\,000 + 810$$

$$\text{system_clock_frequency 随时间的变化率} \leq 75 \times 10^{-3} \text{ Hz/s}$$

在本部分中有几处使用了记号 ‘system_clock_frequency’ 以表示满足以上要求的时钟频率。为了表达方便，有 SCR、PTS 或 DTS 的方程所得到的时间值均为 $(300 \times 2^{33} / \text{system_clock_frequency})$ 秒的整数倍。这是因为 SCR 时间信息被编码为 33 位的系统时钟频率的 1/300，另加 9 位的余数。PTS 及 DTS 被编码为 33 位，计数频率为系统时钟频率的 1/300。

7.2.2 节目流系统目标解码器的输入

来自节目流的数据进入系统解码器。第 i 个字节在时刻 $t(i)$ 进入，通过对输入的系统参考时钟(SCR)字段及编码在分组头中的 `program_mux_rate` 字段进行解码可从输入流中恢复该字节进入系统解码器的时间。式(18)定义的 SCR 分两部分编码：一部分以系统时钟频率的 1/300 周期为单位，称为 `system_clock_reference_base` (见式(19))，另一部分以系统时钟周期为单位，称为 `system_clock_reference_ext` (见式(20))。在下文中，这些字段中的编码值被表示为 $SCR_base(i)$ 及 $SCR_ext(i)$ 。SCR 字段中编码值表示时间 $t(i)$ ，其中 i 指的是包含 `system_clock_reference_base` 字段最后一位的字节。

特别地：

$$SCR(i) = SCR_base(i) \times 300 + SCR_ext(i) \quad (18)$$

式中：

$$SCR_base(i) = ((\text{system_clock_frequency} \times t(i)) \text{ DIV } 300) \% 2^{33} \quad (19)$$

$$SCR_ext(i) = ((\text{system_clock_frequency} \times t(i)) \text{ DIV } 1) \% 300 \quad (20)$$

式(21)给出的输入到达时间 $t(i)$ ，对其余所有字节均从 $SCR(i)$ 及数据到达率构造。其中每一个包的到达率由该分组头中 `program_mux_rate` 字段表示。

$$t(i) = SCR(i') / \text{system_clock_frequency} + (i - i') / (\text{program_mux_rate} \times 50) \quad (21)$$

式中：

i' ——分组头中包含 `system_clock_reference_base` 字段的最后一位的字节索引；

i ——包中任何字节的索引，包括分组头；

$SCR(i')$ ——以系统时钟为单位的系统参考时钟基础和扩展字段的编码值；

program_mux_rate——7.3.3 中定义的字段。

在传输完一个分组的最后字节之后，可能会有一个时间间隔。在这期间，没有字节被传送到 P-STD 的输入。

7.2.3 缓冲

来自基本流 n 的 PES 分组包数据，传送给流 n 的输入缓冲区 B_n 。字节 i 从 STD 到 B_n 的传输是瞬时的，从而字节 i 在时刻 $t(i)$ 进入大小为 BS_n 的流 n 的缓冲区。节目流的分组头、系统头、节目流映象、节目流目录或 PES 分组包头中的字节，诸如 SCR、DTS、PTS 和 packet_length 等字段，不被送入任何缓冲区，但可能被用于控制系统。

从 BS_1 到 BS_n 的输入缓冲区大小在式(16)和(17)的语法中，由 P-STD 缓冲区大小参数给出。

在解码时刻 $td_n(j)$ ，所有在缓冲区中时间最长的访问单元 $A_n(j)$ 以及任何在时刻 $td_n(j)$ 出现在缓冲区中的数据之前的填充字节，在时刻 $td_n(j)$ 被立即移走。解码时刻 $td_n(j)$ 由 DTS 或 PTS 字段指定。紧跟在访问单元 j 之后的无编码 DTS 或 PTS 字段的访问单元的解码时间 $td_n(j+1)$ ， $td_n(j+2)$ ，... 可从基本流信息推导得出，参见 GB/T 20090 的第 2、3 部分及 GB/T 17975.2-2000，GB/T 17975.3-2002，GB/T 17191.2-1997 或 GB/T 17191.3-1997 的附录 C 及 9.5。当访问单元从缓冲区中移出时被立即解码为展现单元。

节目流的构造和 $t(i)$ 的选择应使得大小从 BS_1 到 BS_n 的输入缓冲区在节目流系统解码器中不发生上溢和下溢。

即对于所有的 t 和 n

$$0 \leq F_n(t) \leq BS_n$$

以及在 $t=t(0)$ 前的瞬间

$$F_n(t) = 0$$

$F_n(t)$ 是 P-STD 缓冲区 B_n 的瞬间占用度。

该条件的一个例外是，当视频序列头中的 low_delay 标志被设定为 '0' (见 6.2.6) 或 trick_mode 的状态为真时，P-STD 缓冲区 B_n 可能会下溢。

对所有节目流，系统目标解码器输入缓冲区引起的延迟应不大于 1s，除非是静态图像视频数据，输入缓冲区延迟是指一个字节从进入缓冲区到它被解码的时间差值。

特别地，对于包含在访问单元 j 中的所有数据，在没有静态图像视频数据的情况下，该延迟受以下约束：

$$td_n(j) - t(i) \leq 1s$$

而在有静态图像视频数据的情况下，该延迟受以下约束：

$$td_n(j) - t(i) \leq 60s$$

对节目流，每个包的所有字节均应在其后续包之前进入 P-STD。

若视频序列扩展中的 low_delay 标志置 '1' (参见 GB/T 20090.2-2006 的 7.2.2.1)，bbv 缓冲区可能下溢。

对于这种情况，当在 $td_n(j)$ 所规定的时刻对基本流缓冲区 B_n 进行检查时，该访问单元的全部数据未必已经在 B_n 中。当此情况出现时，应在两个字段周期期间重新检查缓冲区，直到整个访问单元的数据完全进入 B_n ，此时整个访问单元将从 B_n 中立即移出。

bbv 缓冲区下溢允许无限制连续。P-STD 解码器应在与上文及编码在该比特流中的 DTS 或 PTS 值一致的最早时刻从 B_n 中移去访问单元数据。解码器可能不能够按 DTS 和 PTS 的规定重建正确的解码和显示时间，直到 bbv 缓冲区下溢终止，且在该比特流中找到 PTS 或 DTS 字段。

7.2.4 PES 流

构造一个 PES 分组包的连续数据流是有可能的，其中的每个分组含有同一基本流的且有相同 stream_id 值的数据，这种流称为 PES 流。一个 PES 流的 PES-STD 模型与节目流的模型一样，除了用基本流参考时钟(ESCR)取代 SCR，用 ES_rate 取代 program_mux_rate。分流器只把数据送到一个基本流缓冲区。PES-STD 模型中缓冲区的大小 BS_n 定义如下：

对于 GB/T 17975.2-2000 或 GB/T 20090.2-2006 视频：

$$BS_n = bbv_{max}[profile,level] + BS_{oh}$$

$BS_{oh} = (1/750)s \times R_{max}[profile,level]$ ，其中 $bbv_{max}[profile,level]$ 和 $R_{max}[profile,level]$ 分别是 GB/T 17975.2-2000 的表 8-14 和 8-13 中定义的每种档次或 GB/T 20090.2-2006 的表 B.4 和表 B.5 中定义的每种档次、等级和层次的最大 bbv 大小和比特率。 BS_{oh} 被分配用作 PES 分组包头的开销。

对于 GB/T 17191.2-2000 视频：

$$BS_n = BBV_{max} + BS_{oh}$$

$BS_{oh} = (1/750)s \times R_{max}$ ，其中 BBV_{max} 和 R_{max} 分别代表 GB/T 17191.2-1997 中受限参数比特流的最大比特率和最大 bbv_buffer_size 。

对于 GB/T 17191.3-1997、GB/T 17975.3-2002 或 GB/T 20090.3 音频：

$$BS_n = 2848 \text{ 字节}$$

7.2.5 解码和展现

节目流系统目标解码器的解码和展现分别与 6.2.4 及 6.2.5 中定义的传输流系统目标解码器相同。

7.3 节目流语法语义规定

以下语法描述了一个字节流。节目流语法图形表示见附录 F。

7.3.1 节目流

节目流见表 32。

表 32 节目流

语 法	位数	助记符
AVS_program_stream(){ do{ pack() } while (nextbits()==pack_start_code) AVS_program_end_code }	32	bslbf

7.3.2 节目流中各字段的语义定义

AVS 节目终止码 AVS_program_end_code

一个值为 ‘0000 0000 0000 0000 0000 0001 1011 1001’ (0x000001B9)的位串，用来终止节目流。

7.3.3 节目流分组层

节目流分组见表 33，节目流分组头见表 33。

表 33 节目流分组

语 法	位数	助记符
pack() { pack_header() }		

```
while (nextbits()==packet_start_code_prefix) {
```

表 33 （续）

语 法	位数	助记符
PES_packet() } }		

7.3.4 节目流分组中各字段的语义定义

分组起始码字段 pack_start_code

值为 ‘0000 0000 0000 0000 0000 0001 1011 1010’ (0x000001BA)的位串，用来标志一个分组的开始。

系统参考时钟字段 system_clock_reference_base, system_clock_reference_extenstion

系统参考时钟(SCR)是一个分两部分编码的 42 位字段。第一部分 system_clock_reference_base 是一个长度为 33 位的字段，其值 SCR_base(i)由式(19)给出；第二部分 system_clock_reference_extenstion 是一个长度为 9 位的字段，其值 SCR_ext(i)由式(20)给出。SCR 字段指出了基本流中包含 ESCR_base 最后一位的字节到达节目目标解码器输入端的期望时间。

SCR 字段的编码频率要求见 9.1 。

标记位字段 marker_bit

1 位字段，取值'1'。

节目复用速率字段 program_mux_rate

一个 22 位整数，规定 P-STD 在包含该字段的包期间接收节目流的速率。其值以 50 字节/秒为单位。不允许取 0 值。该字段所表示的值用于在 7.2 中定义 P-STD 输入端的字节到达时间。该字段值在本部分中的节目多路复用流的不同包中取值可能不同。

分组填充长度字段 pack_stuffing_length

3 位整数，规定该字段后填充字节的个数。

填充字节字段 stuffing_byte

8 位字段，取值恒为 ‘1111 1111’。该字段能由编码器插入，例如为了满足信道的要求。它由解码器丢弃。在每个分组头中最多只允许有 7 个填充字节。

表 34 节目流分组头

语法	位数	助记符
pack_header() { pack_start_code '01' system_clock_reference_base[32..30] marker_bit system_clock_reference_base[29..15] marker_bit system_clock_reference_base[14..0] marker_bit system_clock_reference_extension	 32 2 3 1 15 1 15 1 9	 bslbf bslbf bslbf bslbf bslbf bslbf uimsbf

表 34 （续）

语 法	位数	助记符
marker_bit	1	bslbf
program_mux_rate	22	uimsbf
marker_bit	1	bslbf
marker_bit	1	bslbf
Reserved	5	bslbf
pack_stuffing_length	3	uimsbf
for (i=0;i<pack_stuffing_length;i++){ stuffing_byte }	8	bslbf
if (nextbits()==system_header_start_code) { system_header() }		
}		

7.3.5 系统头

节目流系统头见表 35。

表 35 节目流系统头

语 法	位数	助记符
system_header() { system_header_start_code header_length marker_bit rate_bound marker_bit audio_bound fixed_flag CSPS_flag system_audio_lock_flag system_video_lock_flag marker_bit vedio_bound packet_rate_restriction_flag Reserved_bits	 32 16 1 22 1 6 1 1 1 1 1 5 1 7	 bslbf uimsbf bslbf uimsbf bslbf uimsbf bslbf bslbf bslbf bslbf uimsbf bslbf bslbf

while (nextbits()=='1') {		
---------------------------	--	--

表 35 （续）

语 法	位数	助记符
stream_id	8	uimsbf
'11'	2	bslbf
P-STD_buffer_bound_scale	1	bslbf
P-STD_buffer_size_bound	13	uimsbf
}		
}		

7.3.6 系统头中各字段的语义定义

系统头起始码字段 system_header_start_code

取值 ‘0000 0000 0000 0000 0000 0001 1011 1011’ (0x000001BB)的位串，指出系统头的开始。

头长度字段 header_length

16 位字段。指出该字段后的系统头的字节长度。在本部分将来的扩充中可能扩展该字段。

速率上限字段 rate_bound

22 位字段。取值不小于编码在节目流的任何包中的 program_mux_rate 字段的最大值。该字段可被解码器用于估计是否有能力对整个流解码。

音频上限字段 audio_bound

6 位字段，取值是在从 0 到 32 的闭区间中的整数，且不小于节目流中解码过程同时有效的 GB/T 17975.3-2002、GB/T 20090.3 和 GB/T 17191.3-1997 音频流的最大数目。在本节中，若 STD 缓冲区非空或展现单元正在 P-STD 模型中展现，则 GB/T 17975.3-2002、GB/T 20090.3 和 GB/T 17191.3-1997 音频流的解码过程是有效的。

固定标志字段 fixed_flag

1 位标志位。置'1'时表示比特率恒定的操作；置'0'时，表示操作的比特率可变。在恒定比特率的操作期间，复用的 GB/T 20090 节目流中的 system_clock_reference 字段值应遵从下面的线性公式：

$$SCR_base(i)=((c1\times i+c2)\text{ DIV }300)\%2^{33}\tag{22}$$

$$SCR_ext(i)=((c1\times i+c2)\text{ DIV }300)\%300\tag{23}$$

式中：

c1——对所有 i 均有效的实型常数；

c2——对所有 i 均有效的实型常数；

i——在 GB/T 20090 复用的节目流中包含任何 system_clock_reference 字段的最后一位的字节索引。

CSPS 标志字段 CSPS_flag

1 位字段。置'1'时，节目流符合 9.8 中定义的限制。

系统音频锁定标志字段 system_audio_lock_flag

1 位字段。表示在系统目标解码器的音频采样率和 system_clock_frequency 之间存在规定的比率。system_clock_frequency 在 7.2.1 中定义而音频采样率由 GB/T 17975.3-2002 或 GB/T 20090.3 规定。如果对节目流中所有音频基本流的所有展现单元，system_clock_frequency 和实际音频采样率的比例 SCASR 是恒定的，且对音频流中所指出的标准采样率 and 下表中数值相等，则该字段只能为'1'。

$$SCASR=(system_clock_frequency)/audio_sample_rate_in_the_P-STD\tag{24}$$

记号 X/Y 表示实数除法。

标准音频采样频率(kHz)	22.05	44.1	24	48	96
SCASR	27 000 000	27 000 000	27 000 000	27 000 000	27 000 000
	-----	-----	-----	-----	-----
	22 050	44 100	24 000	48 000	96 000

系统视频锁定标志字段 system_video_lock_flag

1 位字段。表示在系统目标解码器的视频帧速率和 system_clock_frequency 之间存在规定的比率。system_clock_frequency 在 7.2.1 中定义而视频帧速率由 GB/T 17975.2-2000 或 GB/T 20090.2-2006 规定。如果对 GB/T 17975.1-2000 或本部分中所有视频基本流的所有展现单元，system_clock_frequency 和实际视频帧速率的比例 SCFR 是恒定的，且对视频流中所指出的标准帧速率和下表中数值相等，则该字段只能为'1'。

SCFR=system_clock_frequency / frame_rate_in_the_P-STD (25)

标准帧速率 (Hz)	23.976	24	25	29.97	30	50	59.94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

比率 SCFR 的值是精确的。对于 23.976、29.97 或 59.94 帧/秒的标准速率，实际的帧速率与标准速率略有不同。

视频上限字段 video_bound

5 位字段，取值是在从 0 到 16 的闭区间中的整数且不小于节目流中解码过程同时有效的 GB/T 17975.2-2000、GB/T 20090.2-2006 和 GB/T 17191.2-1997 流的最大数目。在本节中，若 P-STD 缓冲区非空或展现单元正在 P-STD 模型中展现，则 GB/T 17975.2-2000、GB/T 20090.2 和 GB/T 17191.2-1997 视频流的解码过程是有效的。

分组速率限制标志字段 packet_rate_restriction_flag

1 位标志位。若 CSPS 标识为'1'，则该字段表示 9.8 中规定的哪个限制适用于分组速率。若 CSPS 标识为'0'，则该字段的含义未定义。

保留位字段 reserved_bits

7 位字段。被保留供 ISO/IEC 和 AVS 将来使用。它的值应为 '111 1111'，除非 ISO/IEC 和 AVS 对它作出其它规定。

流标识字段 stream_id

8 位字段。指示其后的 P-STD_buffer_bound_scale 和 P-STD_buffer_size_bound 字段所涉及的流的编码和基本流号码。

若取值 '1011 1000'，则其后的 P-STD_buffer_bound_scale 和 P-STD_buffer_size_bound 字段指节目流中所有的音频流。

若取值 '1011 1001'，则其后的 P-STD_buffer_bound_scale 和 P-STD_buffer_size_bound 字段指节目流中所有的视频流。

若 stream_id 取其它值，则应该是大于或等于'1011 1100'的一字节值且应根据表 17 解释为流的编码和基本流号码。

节目流中的每个基本流应在每个系统头中通过这种机制精确地规定一次它的 P-STD_buffer_bound_scale 和 P-STD_buffer_size_bound。

P-STD 缓冲区界限比例字段 P-STD_buffer_bound_scale

1 位字段。表示用于解释后续 P-STD_buffer_size_bound 字段的比例系数。若前面的 stream_id 表示一个音频流，则该字段值为'0'。若表示一个视频流，则该字段值为'1'。对于所有其它的流类型，该字段值可以为'0'也可以为'1'。

P-STD 缓冲区大小界限字段 P-STD_buffer_size_bound

13 位无符号整数，取值不小于节目流中流 n 的所有分组的 P-STD 缓冲区大小 BS_n 的最大值。若 P-STD_buffer_bound_scale 的值为 '0'，则该字段以 128 字节为单位来度量缓冲区大小的边界。若 P-STD_buffer_bound_scale 的值为 '1'，则该字段以 1024 字节为单位来度量缓冲区大小的边界。因此：

```

if (P-STD buffer bound scale == 0)

```

$$BS_n \leq P_STD_buffer_size_bound \times 128$$

else

$$BS_n \leq P_STD_buffer_size_bound \times 1024$$

7.3.7 节目流分组包层

节目流的分组包层由 6.3.6 中的 PES 分组包层来定义。

7.4 节目流映射

节目流映射(PSM)对节目流中的各基本流及它们之间的相互关系作了描述。当携带在传输流中时,该结构不应该被修改。当 stream_id 值为 0xBC 时, PSM 作为 PES 分组包出现(见表 17)。

注：该语法与 6.3.6 中描述的 PES 分组包语法不同。

可以在第 8 章中找到 `descriptor()`字段的定义。

7.4.1 节目流映射语法

节目流映射语法见表 36。

表 36 节目流映射

语 法	位数	助记符
program_stream_map() {		
packet_start_code_prefix	24	bslbf
map_stream_id	8	uimsbf
program_stream_map_length	16	uimsbf
current_next_indicator	1	bslbf
Reserved	2	bslbf
program_stream_map_version	5	uimsbf
Reserved	7	bslbf
marker_bit	1	bslbf
program_stream_info_length	16	uimsbf
For (i=0;i<N;i++){		
descriptor()		
}		
elementary_stream_map_length	16	uimsbf
For (i=0;i<N1;i++){		
stream_type	8	uimsbf
elementary_stream_id	8	uimsbf

elementary_stream_info_length For (i=0;i<N2;i++) {	16	uimsbf
--	-----------	---------------

表 36 （续）

语 法	位数	助记符
<pre>descriptor() } } CRC_32 }</pre>	32	rpchof

7.4.2 节目流映射中各字段的语义定义

分组起始码前缀字段 `packet_start_code_prefix`
24 位码，它和跟随其后的 `map_stream_id` 共同组成一个分组起始码以标志分组的开始。该字段是值为 ‘0000 0000 0000 0000 0000 0001’ (0x000001)的位串。

映射流标识字段 `map_stream_id`
8 位字段，值为 0xBC。

节目流映射长度字段 `program_stream_map_length`
16 位字段，指示紧跟在该字段后的 `program_stream_map` 中的字节数。该字段的最大值为 1018(0x3FA)。

当前下一个指示符字段 `current_next_indicator`
1 位字段，置‘1’时表示传送的节目流映射当前是适用的。置‘0’时表示传送的节目流映射还不适用，但它将是下一个生效的表。

节目流映射版本字段 `program_stream_map_version`
5 位字段，表示整个节目流映射的版本号。一旦节目流映射的定义发生变化，该字段将递增 1，并对 32 取模。在 `current_next_indicator` 为‘1’时，该字段应该是当前适用的节目流映射的版本号；在 `current_next_indicator` 为‘0’时，该字段应该是下一个适用的节目流映射的版本号。

节目流信息长度字段 `program_stream_info_length`
16 位字段，指出紧跟在该字段后的描述子的总长度。

标记位字段 `marker_bit`
1 位字段，取值为‘1’。

基本流映射长度字段 `elementary_stream_map_length`
16 位字段，指出在该节目流映射中的所有基本流信息的字节长度。它包括 `stream_type`、`elementary_stream_id` 和 `elementary_stream_info_length` 字段。

流类型字段 `stream_type`
8 位字段，根据表 30 规定了流的类型。该字段只能标志包含在 PES 分组中的基本流且取值不能为 0x05。

基本流标识字段 `elementary_stream_id`
8 位字段，指出该基本流所在 PES 分组包的头中 `stream_id` 字段的值。

基本流信息长度字段 `elmentary_stream_info_length`
16 位字段，指出紧跟在该字段后的描述子的字节长度。

CRC 32 字段 `CRC_32`
32 位字段，它包含 CRC 值以在处理完整个节目流映射后在附录 A 中定义的解码器寄存器产生 0 输出值。

7.5 节目流目录

一个完整流的目录是由所有由 `directory_stream_id` 标识的 PES 分组包中的目录数据组成的。

program_stream_directory 分组的语法定义见表 37。

注 1： 该语法不同于 6.3.6 中描述的 PES 分组包语法。

可能需要目录项以索引 GB/T 17975.2-2000、GB/T 20090.2-2006 和 GB/T 17191.2-1997 中定义的视频流的 I 图像。如果目录项中索引的 I 图像前面是未插入图像头的序列头，则目录项应指向序列头的首字节。如果目录项中索引的 I 图像前面是一组未插入图像头且前面没有序列头的图像头，则目录项应指向这组图像头的首字节。索引其它任何图像的目录项应指向该图像头的首字节。

注 2： 建议在目录结构中应索引那些紧跟在序列头后的 I 图像，以使得在解码器可能被完全重启的每个点上目录中都包含一个入口。

对 GB/T 17975.3-2002、GB/T 20090.3 和 GB/T 17191.3-1997 中定义的音频流的目录索引应该是音频帧的同步字。

注 3： 建议相邻两个被索引的访问单元的间隔不要超过 0.5s。

在 program_stream_directory 分组中对访问单元进行索引的次序应与它们在比特流中出现的次序相同。

7.5.1 节目流目录分组语法

节目流目录分组语法见表 37。

表 37 节目流目录分组

marker_bit	1	bslbf
audio_bound	6	uimsbf
fixed_flag	1	bslbf
CSPS_flag	1	bslbf
system_audio_lock_flag	1	bslbf
system_video_lock_flag	1	bslbf
marker_bit	1	bslbf
vedio_bound	5	uimsbf
packet_rate_restriction_flag	1	bslbf
reserved_bits	7	bslbf
while (nextbits()=='1') { stream_id	8	uimsbf
directory_PES_packet() { packet_start_code_prefix	24	bslbf
directory_stream_id	8	uimsbf
PES_packet_length	16	uimsbf
number_of_access_units	15	uimsbf
marker_bit	1	bslbf
prev_directory_offset[44..30]	15	uimsbf
marker_bit	1	bslbf
prev_directory_offset[29..15]	15	uimsbf

表 36 (续)

语 法	位数	助记符
marker_bit	1	bslbf
prev_directory_offset[14..0]	15	uimsbf
marker_bit	1	bslbf
next_directory_offset[44..30]	15	uimsbf
marker_bit	1	bslbf
next_directory_offset[29..15]	15	uimsbf
marker_bit	1	bslbf
next_directory_offset[14..0]	15	uimsbf
marker_bit	1	bslbf
for (i=0;i<number_of_access_units;i++) {		
packet_stream_id	8	uimsbf
PES_header_position_offset_sign	1	tcimsbf
PES_header_position_offset[43..30]	14	uimsbf
marker_bit	1	bslbf
PES_header_position_offset[29..15]	15	uimsbf
marker_bit	1	bslbf
PES_header_position_offset[14..0]	15	uimsbf
marker_bit	1	bslbf
reference_offset	16	uimsbf
marker_bit	1	bslbf
Reserved	3	bslbf
PTS[32..30]	3	uimsbf
marker_bit	1	bslbf
PTS[29..15]	15	uimsbf
marker_bit	1	bslbf
PTS[14..0]	15	uimsbf
marker_bit	1	bslbf
bytes_to_read[22..8]	15	uimsbf
marker_bit	1	bslbf
bytes_to_read[7..0]	8	uimsbf
marker_bit	1	bslbf
intra_coded_indicator	1	bslbf
coding_parameters_indicator	2	bslbf

表 36 （续）

语 法	位数	助记符
Reserved	4	bslbf
<pre> } } </pre>		

7.5.2 节目流目录中各字段的语义定义

分组起始码前缀字段 `packet_start_code_prefix`

24 位字段。它和跟随其后的 `stream_id` 共同组成一个分组起始码以标志分组的开始。该字段是值为‘0000 0000 0000 0000 0001’ (0x000001)的位串。

目录流标识字段 `directory_stream_id`

8 位字段，值为‘1111 1111’ (0xFF)。

PES 分组包长度字段 `PES_packet_length`

16 位字段。指示紧跟在该字段后的 `program_stream_directory` 中的字节数(见表 17)。

访问单元数字段 `number_of_access_units`

15 位字段。表示该目录 PES 分组包中索引的访问单元数。

前一个目录偏移字段 `prev_directory_offset`

45 位无符号整数。给出了前一个节目流目录分组中分组起始码的首字节的地址偏移字节数。

下一个目录偏移字段 `next_directory_offset`

45 位无符号整数。给出了下一个节目流目录分组中分组起始码的首字节的地址偏移字节数。该地址偏移是相对于包含该字段的分组的起始码的首字节而言的。值'0'表示不存在下一个节目流目录分组。

分组流标识字段 `packet_stream_id`

8 位字段。是包含本目录项所引用的访问单元的基本流的 `stream_id`。

PES 头位置偏移符号字段 `PES_header_position_offset_sign`

1 位字段。它用作紧跟其后描述的 `PES_header_position_offset` 的算术符号。值'0'表示 `PES_header_position_offset` 是一个正的偏移量；值'1'表示 `PES_header_position_offset` 是一个负的偏移量。

PES 头位置偏移字段 `PES_header_position_offset`

44 位无符号整数。给出了包含被索引访问单元的 PES 分组包的首字节的地址偏移字节数。该地址偏移是相对于包含该字段的分组的起始码的首字节而言的。值'0'表示不存在被索引访问单元。

参考偏移字段 `reference_offset`

16 位无符号整数。给出了被索引访问单元的首字节的位置。该字段是相对于包含被索引访问单元的 PES 分组包的首字节而言的，且以字节为单位。

展现时间戳字段 `PTS(presentation_time_stamp)`

33 位字段，表示被引用访问单元的 PTS。PTS 字段的编码语义见 6.3.6 中的描述。

待读字节字段 `bytes_to_read`

23 位无符号整数。表示节目流中在 `reference_offset` 指向的字节之后为了对访问单元完全解码所需的字节数。该值包含复用在系统层中的任何字节，包括来自于其它流的信息。

帧内编码指示符字段 `intra_coded_indicator`

1 位标志。置'1'时表示被引用的访问单元未进行预测编码。它独立于解码访问单元所需的其余编码参数。例如，对于视频 I 帧，该字段应编码为'1'；而对于 P 帧和 B 帧，该字段应编码为'0'。对于包含来源于 GB/T 17975.2-2000 或 GB/T 20090.2-2006 以外数据的所有 PES 分组包，该字段未定义(见表 38)。

表 38 帧内编码指示符

值	意义
0	非帧内
1	帧内

编码参数指示符字段 coding_parameters_indicator
2 位字段，用于指示解码被索引访问单元所需编码参数的位置。例如，该字段可用来决定视频帧量化矩阵的位置。见表 39。

表 39 编码参数指示符

值	意 义
00	所有编码参数置为缺省值
01	访问单元中的所有编码参数均被设定，其中至少有一个编码参数不为缺省值
10	该访问单元中部分编码参数被设置
11	该访问单元中无编码参数被设置

8 节目和节目元素描述子

节目和节目元素描述子是用于扩展节目和节目元素定义的结构，所有描述子的格式均以 一个 8 位标签值为开始，后跟 8 位描述子长度及数据字段。

8.1 节目和节目元素描述子中各字段的语义定义

以下语义适用于从 8.2 到 8.34 中定义的描述子。

描述子标签字段 descriptor_tag

8 位字段，用于标识每一描述子。

表 40 给出了 GB/T 20090.1 定义和保留的且用户可用的描述子标签值。TS 或 PS 栏中的 X 表示该描述子可分别用于传输流或节目流。注意，描述子字段含义可能取决于它用于的流。以下描述子语义对每种情况作了规定。

表 40 节目和节目元素描述子

描述子标签值	TS	PS	标 识
0	n/a	n/a	保留
1	n/a	n/a	保留
2	X	X	video_stream_descriptor
3	X	X	audio_stream_descriptor
4	X	X	hierarchy_descriptor
5	X	X	registration_descriptor
6	X	X	data_stream_alignment_descriptor
7	X	X	target_background_grid_descriptor
8	X	X	video_window_descriptor

表 40 （续）

描述子标签值	TS	PS	标 识
9	X	X	CA_descriptor
10	X	X	ISO_639_language_descriptor
11	X	X	System_clock_descriptor
12	X	X	multiplex_buffer_utilization_descriptor
13	X	X	copyright_descriptor
14	X		最大比特率描述子
15	X	X	私有数据指示符描述子
16	X	X	平滑缓冲区描述子
17	X	X	STD_descriptor
18	X	X	IBP描述子
43~62	n/a	n/a	GB/T 20090.1保留
63	X	X	AVS_Video_descriptor
64~255	n/a	n/a	用户私有

描述子长度字段 descriptor_length

8 位字段。规定了紧跟在该字段之后的描述子的字节数。

8.2 视频流描述子

8.2.1 非 AVS 视频流描述子

视频流描述子给出了一些基本信息以标识 GB/T 17975.2-2000 或 GB/T 17191.2-1997 中描述的视频基本流和编码参数。见表 41。

表 41 非 AVS 视频流描述子

语 法	位数	助记符
Video_stream_descriptor() {		
descriptor_tag	8	Uimsbf
descriptor_length	8	Uimsbf
multiple_frame_rate_flag	1	Bslbf
frame_rate_code	4	Uimsbf
MPEG_1_only_flag	1	Bslbf
constrained_parameter_flag	1	Bslbf
still_picture_flag	1	Bslbf
If (MPEG1_only_flag == '0') {		
profile_and_level_indication	8	Uimsbf

chroma_format	2	Uimsbf
frame_rate_extension_flag	1	Bslbf

表 41 （续）

语 法	位数	助记符
Reserved	5	bslbf
}		
}		

8.2.2 AVS 视频流描述子

视频流描述子给出了一些基本信息以标识 GB/T 20090.2-2006 中描述的视频基本流和编码参数。见表 42。

表 42 AVS 视频流描述子

语 法	位数	助记符
AVS_video_descriptor () {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
profile_id	8	uimsbf
level_id	8	uimsbf
multiple_frame_rate_flag	1	bslbf
frame_rate_code	4	uimsbf
AVS_still_present	1	bslbf
chroma_format	2	uimsbf
sample_precision	3	uimsbf
reserved	5	bslbf
}		

8.3 视频流描述子中各字段的语义定义

8.3.1 非 AVS 视频流描述子中各字段的语义定义

复合帧速率标志字段 multiple_frame_rate_flag

1 位字段，置'1'时表示视频流中可能有多个帧速率，置'0'时表示只有单一帧速率。

帧速率码字段 frame_rate_code

4 位字段，和 GB/T 20090.2-2006 的 7.2.2.1 中定义、GB/T 17975.2-2000 的 0 中定义相类似。不同点在于 multiple_frame_rate_flag 字段置'1'时，一个特定的帧速率意味着视频流中允许有某些其它的帧速率。如表 43 所示。

表 43 帧速率码

编码速率	同时允许的速率
23.976	
24.0	23.976

25.0	
29.97	23.976
30.0	23.976 24.0 29.97
50.0	25.0
59.94	23.976 29.97
60.0	23.976 24.0 29.97 30.0 59.94

仅含 MPEG1 标志字段 `MPEG_1_only_flag`

1 位字段。置'1'时表示该视频流仅含 GB/T 17191.2-1997 数据；置'0'时表示该视频流可能包含 GB/T 20090.2-2006 或 GB/T 17975.2-2000 视频数据和受限参数 GB/T 17191.2-1997 视频数据。

受限参数标志字段 `constrained_parameter_flag`

1 位字段。置'1'时表示视频流不包括非受限 GB/T 17191.2-1997 视频数据；置'0'时表示该视频流可能包含受限参数和非受限的 GB/T 17191.2-1997 视频数据。若 `MPEG_1_only_flag` 为'0'，则本字段应为'1'。

静态图象标志字段 `still_picture_flag`

1 位字段。置'1'时表示该视频流只含静态图象；置'0'时，则可包含运动的或静态的图象数据。

档次与等级指示符字段 `profile_and_level_indication`

8 位字段。该字段与非 GB/T 17191.2-1997 视频流中 `profile_and_level_indication` 字段相同。该字段的值表示一个档次和等级。在本小节中，一个 GB/T 17191.2-1997 受限参数流被认为是在低等级上的一个主档次(MP @ LL)。

色度格式字段 `chroma_format`

2 位字段。该字段与 GB/T 20090.2-2006 或 GB/T 17975.2-2000 视频流中 `chroma_format` 字段编码方式相同。该字段值应等于相关视频流的任何视频序列中的 `chroma_format` 字段值。在本小节中，认为 GB/T 17191.2-1997 视频流有一个值为 '01' 的 `chroma_format` 字段，表示 4:2:0。

帧速率扩展标志字段 `frame_rate_extension_flag`

1 位字段。置'1'时表示 GB/T 17975.2-2000 视频流的任何序列中的 `frame_rate_extension_n` 和 `frame_rate_extension_d` 字段不能同时为 0。在本小节中，限定 GB/T 17191.2-1997 视频流的这两个字段均为 0。

8.3.2 AVS 视频流描述子中各字段的语义定义

`profile_id`

8 位字段。表示比特流的档次，该字段与 GB/T 20090.2-2006 视频流中 `profile_id` 字段相同。

`level_id`

8 位字段。表示比特流的等级。该字段与 GB/T 20090.2-2006 视频流中 `level_id` 字段相同。

复合帧速率标志字段 `multiple_frame_rate_flag`

1 位字段，置'1'时表示视频流中可能有多个帧速率，置'0'时表示只有单一帧速率。

帧速率码字段 `frame_rate_code`

4 位字段，该字段与 GB/T 20090.2-2006 视频流中 `frame_rate_code` 字段定义相同。不同点在于 `multiple_frame_rate_flag` 字段置'1'时，一个特定的帧速率意味着视频流中允许有某些其它的帧速率。如表 43 所示。

`AVS_still_present`

1 位字段。置'1'时表示该视频流只含静态图象；置'0'时，则可包含运动的或静态的图象数据。

`chroma_format`

2 位字段。规定色度分量的格式。该字段与 GB/T 20090.2-2006 视频流中 `chroma_format` 字段编码方式相同。

`sample_precision`

3 位字段。规定亮度和色度样本的精度。该字段与 GB/T 20090.2-2006 视频流中 sample_precision 字段编码方式相同。

8.4 音频流描述子

音频流描述子给出了基本信息以表示 GB/T 20090.3、GB/T 17975.3-2002 或 GB/T 17191.3-1997 中所描述的音频基本流的编码版本。见表 44。

表 44 音频流描述子

语 法	位数	助记符
audio_stream_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
free_format_flag	1	bslbf
ID	1	bslbf
Layer	2	bslbf
variable_rate_audio_indicator	1	bslbf
Reserved	3	bslbf
}		

8.5 音频流描述子中各字段的语义定义

自由格式标志字段 free_format_flag

1 位字段。置'1'时表示音频流或包含一或多个 bitrate_index 置为 '0000' 的音频帧；置'0'时音频流的任何音频帧中的 bitrate_index 不为 '0000'。(参见 GB/T 20090. 3 或 GB/T 17975.3-2002 中的 2.4.2.3)

ID 字段 ID

1 位字段。设置成与音频流中 ID 字段相同的值。(参见 GB/T 20090.3 或 GB/T 17975.3-2002 中的 2.4.2.3)。

层字段 layer

2 位字段。与 GB/T 20090.3 音频流、GB/T 17975.3-2002 或 GB/T 17191.3-1997 音频流中 layer 字段编码方式相同(参见 GB/T 20090.3 或 GB/T 17975.3-2002 中的 2.4.2.3)。这一字段指出的层应等于或高于音频流的任何音频帧中所指出的最高层。

可变速率音频指示符字段 variable_rate_audio_indicator

1 位标志。置'1'时表示在连续的音频帧之间相关音频流的比特速率可变。连续编码的可变速率的音频在展现时不应该不连续。

8.6 层次描述子

层次描述子，给出了信息以表示包含分层编码音频和视频及私有流的节目元素。这些节目元素被复合在多个流中。如本部分、GB/T 17975.1-2000、GB/T 20090.3、GB/T 17975.2-2000、GB/T 17975.3-2002 和 GB/T 17191. 3-1997 中所述。见表 45。

表 45 层次描述子

语 法	位数	助记符
-----	----	-----

hierarchy_descriptor(){		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
Reserved	4	bslbf
hierarchy_type	4	uimsbf

表 45 （续）

语 法	位数	助记符
Reserved	2	bslbf
hierarchy_layer_index	6	uimsbf
Reserved	2	bslbf
hierarchgy_embedded_layer_index	6	uimsbf
Reserved	2	bslbf
hierarchy_channel	6	uimsbf
}		

8.7 层次描述子中各字段的语义定义

层次类型 hierarchy_type

相关层次和它的分层嵌入层之间的层次关系见表 46。

表 46 hierarchy_type 字段值

值	描 述
0	保留
1	GB/T 17975.2-2000空间可伸缩性
2	GB/T 17975.2-2000SNR可伸缩扩展性
3	GB/T 17975.2-2000当前可伸缩扩展性
4	GB/T 17975.2-2000数据分割
5	GB/T 20090.3或者GB/T 17975.3-2002扩展比特流
6	GB/T 20090.1或者GB/T 17975.1-2000私有流
7	GB/T 17975.2-2000多视角档次
8~14	保留
15	基本层

层次层索引字段 hierarchy_layer_index

6 位字段。定义了编码层层次表中相关节目元素的唯一索引。在一个节目定义中索引是唯一的。

分层嵌入层索引字段 hierarchy_embedded_layer_index

6 位字段。定义了节目元素的层次表索引。在解码与层次描述子相关的基本流之前必须访问该节目元素。若 hierarchy_type 值为 15(基本层)，则该字段未定义。

层次信道字段 hierarchy_channel

6 位字段，指示在传输信道有序集合中，相关节目元素的期望信道号。该字段的最低值定义了最健壮的传输信道。

注： 一个给定的该字段可能同时被赋给几个节目元素。

8.8 注册描述子

注册描述子提供了一种方法以唯一地、明确地标识私有数据的格式(见表 47)。

表 47 注册描述子

语 法	位数	助记符
<pre> registration_descriptor() { descriptor_tag descriptor_length format_identifier for (i=0;i<N;i++) { additional_identification_info } } </pre>	<p>8</p> <p>8</p> <p>32</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p> <p>bslbf</p>

8.9 注册描述子中各字段的语义定义

格式标识符字段 `format_identifier`

32 位值，从国际或国内标准化组织指定的注册机构处获得。

附加鉴定信息字段 `additional_identification_info`

若有该字段的话，则其含义由 `format_identifier` 授予者定义，一旦定义后将不再变更。

8.10 数据流对齐描述子

该描述子描述了相关基本流中出现的对齐类型。如果 PES 分组包头中的 `data_alignment_indicator` 为'1'，且有该描述子，则该描述子所指明的对齐是必须的。见表 48。

表 48 数据流对齐描述子

语 法	位数	助记符
<pre> data_stream_alignment_descriptor() { descriptor_tag descriptor_length alignment_type } </pre>	<p>8</p> <p>8</p> <p>8</p>	<p>uimsbf</p> <p>uimsbf</p> <p>uimsbf</p>

8.11 数据流对齐描述子中各字段的语义定义

对齐类型字段 `alignment_type`

表 49 描述了当 PES 分组包头中 `data_alignment_indicator` 为'1'时的视频对齐类型。对于该字段的每种取值，跟在 PES 包头之后的 `PES_packet_data_type` 应该是具有表 49 中所指出类型的起始码的首字节。在视频序列起始处，对齐应发生在第一个序列头的起始码处。

注：指定表 49 的对齐类型'01'并不排除从 GOP 或 SEQ 头的开始处对齐。

视频数据访问单元的定义见 0。

表 49 视频流对齐值

对齐类型	描 述
00	保留
01	条带或视频访问单元
02	视频访问单元
03	GOP或SEG
04	SEQ
05~FF	保留

表 50 描述了在 PES 分组包头中 data_alignment_indicator 值为'1'时的音频流对齐类型。在这种情况下，紧跟着 PES 包头的第一个 PES_packet_data_byte 是音频同步字的首字节。

表 50 音频流对齐值

对齐类型	描 述
00	保留
01	同步字
02~FF	保留

8.12 目标背景栅格描述子

可能有一个或多个视频流在解码后不占据整个显示区域（例如监视器）。target_background_grid_descriptor 和 video_window_descriptor 的组合可使这些视频窗口在期望的区域显示。前者用于描述投影于显示区域的单元像素的栅格而后者用于描述相关流的显示窗口的左上角的像素位置或视频展现单元的显示矩形在栅格上的显示位置。图 8 对此作了表示。

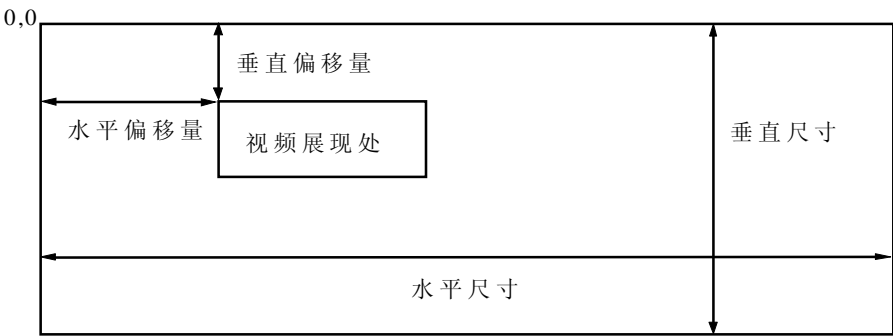


图 8 目标背景栅格描述子显示区域

8.13 目标背景栅格描述子中各字段的语义

- 水平尺寸字段 horizontal_size
以像素为单位的目标背景栅格水平尺寸。
- 垂直尺寸字段 veritical_size
以像素为单位的目标背景栅格垂直尺寸。

宽高比信息字段 aspect_ratio_information
规定目标背景栅格的采样宽高比或显示宽高比。该字段在 GB/T 20090.2-2006 或 GB/T 17975.2-2000 中定义。见表 51。

表 51 目标背景栅格描述子

语 法	位数	助记符
target_background_grid_descriptor() { descriptor_tag descriptor_length horizontal_size vertical_size Aspect_ratio_information }	 8 8 14 14 4	 uimsbf uimsbf uimsbf uimsbf uimsbf

8.14 视频窗描述子
视频窗描述子用于描述相关视频基本流的窗口特性。其值以与同一个流相应的目标背景栅格描述子为参考。见 8.12 中的 target_background_grid_descriptor。见表 52。

表 52 视频窗描述子

语 法	位数	助记符
video_window_descriptor() { descriptor_tag descriptor_length horizontal_offset vertical_offset window_priority }	 8 8 14 14 4	 uimsbf uimsbf uimsbf uimsbf uimsbf

8.15 视频窗描述子中各字段的语义定义

水平偏移字段 horizontal_offset
指示当前视频显示窗或显示矩形的左上角像素在目标背景栅格描述子 target_background_grid_descriptor 中定义的用于显示的目标背景栅格上的水平位置。视频窗的左上角像素须为目标背景栅格的像素之一(见图 8)。

垂直偏移字段 vertical_offset
指示当前视频显示窗或显示矩形的左上角像素在目标背景栅格描述子 target_background_grid_descriptor 中定义的用于显示的目标背景栅格上的垂直位置。视频窗的左上角像素须为目标背景栅格的像素之一(见图 8)。

窗口优先级字段 window_priority
指出窗口如何覆盖。值从 0 到 15 优先级依次递增，即优先级为 15 的窗口总是可见的。

8.16 条件访问描述子

该描述子既用于规定如 EMM 这样的系统范围条件访问管理信息，又用于规定如 ECM 这样的基本流特定信息。可用于传输流 program_map_section(见 6.5.8)和 program_stream_map(见 7.3)。如果任一基本流是加扰的，则包含该基本流的节目需要一个 CA 描述子。若一个传输流中有任一系统范围的条
件访问管理信息，则条件访问表中应有 CA 描述子。

当 TS_program_map_section 中含 CA 描述子时(table_id = 0x02)，CA_PEID 指向含有节目相关访问信息的分组，例如 ECM。它作为节目信息出现表示对整个节目都适用。同样，它作为扩展 ES 信息出现表示对相关节目元素也适用。对私有流也作了规定。

当 CA_section 含 CA 描述子时(table_id = 0x01)，CA_PEID 指向包含系统范围及(或)访问控制管理信息(如 EMM)的包。

包含条件访问信息的传输流包的内容有专门的定义。见表 53。

表 53 条件访问描述子

语 法	位数	助记符
CA_descriptor() { descriptor_tag descriptor_length CA_system_ID Reserved CA_PEID for (i=0;i<N;i++) { private_data_byte } }	 8 8 16 3 13 8	 uimbsf uimbsf uimbsf bslbf uimbsf

8.17 条件访问描述子中字段的语义定义

CA 系统 ID 字段 CA_system_ID

16 位字段，指出用于相关的 ECM 及(或)EMM 流的 CA 系统类型。它的编码方式是专门定义的，而不是由国际标准化组织指定。

CA PEID 字段 CA_PEID

13 位字段。指出包含用于相关 CA_system_ID 所规定的 CA 系统的 ECM 或 EMM 信息的传输流包的 PEID。CA_PEID 所指示的包的内容(ECM 或 EMM)由 CA_PEID 所出现的上下文决定，即传输流中的 TS_program_map_section 或 CA 表，或节目流中的 stream_id 字段。

8.18 ISO 639 语言描述子

该描述子用于规定相应节目元素的语言，见表 54。

表 54 ISO 639 语言描述子

语 法	位数	助记符
ISO_639_language_descriptor() {		

descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0;i<N;i++) {		
ISO_639_language_code	24	bslbf
audio_type	8	bslbf

表 53 （续）

语 法	位数	助记符
} }		

8.19 ISO 639 语言描述子中各字段的语义定义

ISO 639 语言码字段 ISO_639_language_code

标识相关节目元素使用的一种或几种语言。它包含 ISO 639 中第 2 部分规定的 3 字符代码。根据 GB/T 15273.1-1994，每个字符被编码为 8 位并按顺序插入到该 24 位的字段中。在多语言音频流中，该字段的顺序应反映音频流的内容。

音频类型字段 audio_type

8 位字段，规定表 55 定义的流类型。

表 55 音频类型值

值	描 述
0x00	未定义
0x01	无语音效果clean effects
0x02	受损听力hearing impaired
0x03	受损视觉描述visual_impaired_commentary
0x04~0xFF	保留

无语音 clean effects

表示所涉及的节目元素无语言。

受损听觉 hearing impaired

表示所涉及的节目元素适用于听觉受损者。

受损视觉描述 visual_impaired_commentary

表示所涉及的节目元素适用于视觉受损者。

8.20 系统时钟描述子

该描述子传达用于产生时间戳的信息。

若使用外部时钟参考，则 external_clock_reference_indicator 可能被置'1'。此时，解码器可使用同一个外部时钟参考。

若系统时钟比要求的 30×10^{-6} (30ppm)更精确，则其精度可编码在 clock_accuracy 字段中，被传送的时钟频率精度为：

$$\text{clock_accuracy_integer} \times 10^{-\text{clock_accuracy_exponent}} \times 10^{-6} \tag{26}$$

若 clock_accuracy_integer 置 0，则系统时钟精度为 30×10^{-6} (30ppm)；

若 external_clock_reference_indicator 置'1'，则时钟精度属于外部参考时钟。见表 56。

表 56 系统时钟描述子

语 法	位数	助记符
system_clock_descriptor() {		

descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf

表 56 （续）

语 法	位数	助记符
external_clock_reference_indicator	1	bslbf
Reserved	1	bslbf
clock_accuracy_integer	6	uimsbf
clock_accuracy_exponent	3	uimsbf
Reserved	5	bslbf
}		

8.21 系统时钟描述子中字段的语义定义

外部时钟参考指示符字段 external_clock_reference_indicator

1 位指示符。置'1'时表示系统时钟来源于解码器中可能有的外部频率参考。

时钟精度整数字段 clock_accuracy_integer

6 位整数。与 clock_accuracy_exponent 一起给出以 ppm 为单位的系统时钟频率的分数精度。

时钟精度指数字段 clock_accuracy_exponent

3 位整数。与 clock_accuracy_integer 一起给出以 ppm 为单位的系统时钟频率的分数精度。

8.22 复用缓冲区应用描述子

该描述子给出了 STD 复用缓冲区的占用边界。该信息用于再复用器之类的设备，以支持所期望的再复用策略。见表 57。

表 57 复用缓冲区应用描述子

语 法	位数	助记符
multiplex_buffer_utilization_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
bound_valid_flag	1	bslbf
LTW_offset_lower_bound	15	uimsbf
Reserved	1	bslbf
LTW_offset_upper_bound	15	uimsbf
}		

8.23 复用缓冲区应用描述子中各字段的语义定义

边界有效标志字段 bound_valid_flag

置'1'时表示 LTW_offset_lower_bound 和 LTW_offset_upper_bound 字段有效。

LTW 偏移下界字段 LTW_offset_lower_bound

15 位字段，只有 bound_valid_flag 为'1'时才有定义。在有定义时，该字段以(27 MHz / 300)的时钟周期为单位，与 LTW_offset 的定义相似(见 6.3.4)。如果该描述子所涉及的流的每一个分组都有 LTW_offset 字段编码的话，则它表示任何 LTW_offset 字段能取到的最小值。实际上，在有复用缓冲区应用描述子

时，LTW_offset 字段不一定编码在比特流中。该边界始终有效，直到再次出现该描述子。

LTW 偏移上界字段 LTW_offset_upper_bound

15 位字段，仅当 bound_valid 值为'1'时才有定义。当有定义时，该字段以 27 MHz/300 时钟周期为单位，与 LTW_offset 的定义相似(见 6.3.4)。如果该描述子所涉及的流的每一个分组都有 LTW_offset 字段编码的话，则它表示任何 LTW_offset 字段能取到的最大值。实际上，在有复用缓冲区应用描述子时，LTW_offset 字段不一定编码在比特流中。该边界始终有效，直到再次出现该描述子。

8.24 版权描述子

该描述子提供了一种方法以标识音像作品。它适用于各种节目或节目中的节目元素。见表 58。

表 58 版权描述子

语 法	位数	助记符
copyright_descriptor() { descriptor_tag descriptor_length copyright_identifier for(i=0;i<n;i++) { additional_copyright_info } }	 8 8 32 8	 uimsbf uimsbf uimsbf bslbf

8.25 版权描述子中各字段的语义定义

版权标识符字段 copyright_identifier

该字段是一个从注册机构获得的 32 位值。

附加版权信息字段 additional_copyright_info

该字段的含义，如果有的话，由 copyright_identifier 的接收者定义，且一旦定义就不再变动。

8.26 最大比特率描述子

最大比特率描述子见表 59。

表 59 最大比特率描述子

语 法	位数	助记符
maximum_bitrate_descriptor() { descriptor_tag descriptor_length Reserved maximum_bitrate }	 8 8 2 22	 uimsbf uimsbf bslbf uimsbf

8.27 最大比特率描述子中字段的语义定义

最大比特率字段 `maximum_bitrate`

22 位正整数字段，指示节目元素或节目中比特率上界(包括传输开销)，以 50 字节/秒为单位。该字段包含于节目映射表中。作为扩展节目信息，它适用于整个节目；作为 ES 信息，它适用于相关节目元素。

8.28 私有数据指示符描述子

私有数据指示符描述子见表 60。

表 60 私有数据指示符描述子

语 法	位数	助记符
<code>private_data_indicator_descriptor() {</code> descriptor_tag descriptor_length private_data_indicator <code>}</code>	 8 8 32	 uimsbf uimsbf uimsbf

8.29 私有数据指示符描述子中各字段的语义定义

私有数据指示符字段 `private_data_indicator`

私有字段，由使用者自定义。

8.30 平滑缓冲区描述子

平滑缓冲区描述子见表 60。该描述子为可选的，且为它所指的节目元素表示与该描述子相关的平滑缓冲区 SB_n 的大小，和与该缓冲区相关的流出率的信息。

对于传输流，相关节目元素(n)的传输流包字节在由式(4)定义的时刻进入大小为 `sb_size` 的缓冲区 SB_n 中。

对于节目流，相关基本流的所有 PES 分组包字节在由式(21)定义的时刻进入大小为 `sb_size` 的缓冲区 SB_n 中。

当缓冲区中已有数据时，字节以速率 `sb_leak_rate` 从中流出。 SB_n 不能上溢。在一个节目的持续过程中，该节目中不同节目元素的平滑缓冲区描述子的值不变。

只在 PMT 或节目流映射中包含该描述子时，它才有定义。

对于传输流，若该描述子存在于 PMT 的 ES 信息中，则具有该节目元素 PEID 的所有传输流分组均进入平滑缓冲区。

对于传输流，若该描述子存在于节目信息中，则以下传输流包进入平滑缓冲区：

在扩充节目信息中，具有作为 `elementary_PEID` 列出的 PEID 的所有传输流包。

具有与本节中 `PMT_PEID` 相等 PEID 的所有传输流包。

节目中有 `clock_recovery_PEID` 的所有传输流包。

进入相关缓冲区的字节都会流出缓冲区。

在给定时刻，最多只有一个描述子指向任一单独的节目元素，最多只有一个描述子指向该节目实体。

表 61 平滑缓冲区描述子

语 法	位数	助记符
<code>smoothing_buffer_descriptor() {</code>		

descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
Reserved	2	bslbf
sb_leak_rate	22	uimsbf
Reserved	2	bslbf

表 61 （续）

语 法	位数	助记符
sb_size }	22	uimsbf

8.31 平滑缓冲区描述子中各字段的语义定义

平滑缓冲区泄漏速率字段 **sb_leak_rate**

22 位正整数字段。以 400 比特/秒为单位，表示相关基本流或其它数据流出 **SB_n** 缓冲区的速率。

平滑缓冲区大小字段 **sb_size**

22 位正整数字段。以字节为单位，表示相关基本流或其它数据的平滑缓冲区 **SB_n** 的大小。

8.32 STD 描述子

该描述子为可选的，仅适用于 T-STD 模型及视频基本流。其使用见 6.2 。该描述子不适用于节目流。
见表 62。

表 62 STD 描述子

语 法	位数	助记符
STD_descriptor() { descriptor_tag descriptor_length Reserved leak_valid_flag }	 8 8 7 1	 uimsbf uimsbf bslbf bslbf

8.33 STD 描述子中各字段的语义定义

泄漏有效标志字段 **leak_valid_flag**

1 位标志位。值为'1'时，在 T-STD 中从缓冲区 **MB_n** 至 **EB_n** 的数据传输使用 6.2.3 定义的泄漏方式；值为'0'时并且相关视频流中的 **bbv_delay** 字段不为 0xFFFF，则从缓冲区 **MB_n** 至 **EB_n** 的数据传输使用 6.2.3 定义的 **bbv_delay** 方式。

8.34 IBP 描述子

该可选描述子提供了视频序列中帧类型序列的某些特征信息。见表 63。

表 63 IBP 描述子

语 法	位数	助记符
ibp_descriptor() { descriptor_tag descriptor_length closed_gop_flag }	 8 8 1	 uimsbf uimsbf uimsbf

identical_gop_flag	1	uimsbf
max_gop_length	14	uimsbf
}		

8.35 IBP 描述子中各字段的语义定义

闭合图像组标志字段 **closed_gop_flag**

1 位标志。在 GB/T 20090.2-2006 中此位不作定义；在 GB/T 17975.2-2000 中，置'1'时表示图像头组在每个 I 帧之前编码，且在视频序列中所有图像组头的 **closed_gop** 标志应设置为'1'。

相同图像组标志字段 **identical_gop_flag**

1 位标志。置'1'时表示 I 帧之间的 P 帧与 B 帧的数目，且图像编码类型及 I 图像之间的图像类型序列在整个序列中相同。

最大图像组长度字段 **max_gop_length**

14 位无符号数。表示序列中任意两个连续的 I 图像之间的最大编码图像数。该字段取值不能为'0'。

9 对多路复用流语义的约束

9.1 系统参考时钟的编码频率

在节目流的编码中，应使得相继包中包含 **system_clock_reference_base** 字段末位的字节间的时间间隔均小于或等于 0.7s，也即对任何 i 和 i' 都有

$$|t(i) - t(i')| \leq 0.7s$$

式中：

i 和 i' ——包含相继出现的 **system_clock_reference_base** 字段末位的字节下标。

9.2 节目参考时钟的编码频率

在传输流的编码中，应使得对每一个节目而言，其 **clock_recovery_PEID** 传输流包中相继 **transport_rate** 字段末位的字节间的时间间隔均小于或等于 0.1 s，也即对任何 i 和 i' 都有

$$|t(i) - t(i')| \leq 0.1s$$

式中：

i 和 i' ——每一个节目 **clock_recovery_PEID** 的传输分组中包含相继出现的 **transport_rate** 字段末位的字节下标。

从节目流中指定的 **clock_recovery_PEID** 开始，在相继两个 **transport_rate** 不连续点(见 6.3.4)之间至少应有两个 **transport_rate**，以便于保证相位锁定和计算字节传输时间。

9.3 基本流参考时钟的编码频率

在节目流和传输流的编码构造中，若基本流参考时钟字段编码在任意包含一给定基本流数据的 PES 分组包中，则 **PES_STD** 中包含相继 **ESCR_base** 字段的字节之间的时间间隔应不大于 0.7s。在 PES 流中，**ESCR** 编码也要求同样的时间间隔。即对所有的 i 和 i' ，

$$|t(i) - t(i')| \leq 0.7s$$

式中：

i 和 i' ——包含相继出现的 **ESCR_base** 字段末位的字节下标。

注：基本流参考时钟字段的编码是可选的，它不一定被编码，但如果被编码，就要遵循以上限制。

9.4 展现时间戳的编码频率

在节目流和传输流的编码构造中，指向每一基本音频或视频流显示时间之间的最大时间间隔应不大于 0.7s。即

$$|tp_n(k) - tp_n(k'')| \leq 0.7s$$

对所有的 n , k 和 k'' 满足:

$P_n(k)$ 和 $P_n(k'')$ 是指编码展现时间戳对应的展现单元;

k 和 k'' 的选择应使得不存在满足 $k < k' < k''$ 编码展现时间戳的展现单元 $P_n(k')$, 而且

在基本流 n 中, $P_n(k)$ 和 $P_n(k'')$ 之间没有解码不连续点。

对静态图像, 0.7s 的限制并不适用。

9.5 时间戳的条件编码

对传输流或节目流的每一基本流而言, 应该为第一个访问单元进行展现时间戳(PTS)编码。

如果基本流 n 中一个访问单元 $A_n(j)$ 的解码时间大于 `system_clock_frequency` 容限下的最大允许值, 则该访问单元的起始处存在解码不连续。对视频而言, 除非处于特技方式或 `low_delay` 标志置'1', 该情况只允许出现在视频序列开始处。如果在节目流或传输流的音频或视频基本流中存在解码不连续, 除非处于特技模式, 每一解码不连续点后的第一个访问单元应有对应的 `relative_display_time` 编码。

当 `low_delay` 置'1'时, PTS 应对 EB_n 或 B_n 下溢后的第一个访问单元编码。

如果一个图像起始码或音频访问单元的首字节包含在一个 GB/T 20090.2-2006 视频或 GB/T 20090.3 音频基本流 PES 分组包中, 则 PTS 编码可能仅出现在该分组包头中。

9.6 PES 分组包头中 P-STD_buffer_size 的编码频率

在一个节目流中, `P-STD_buffer_scale` 和 `P-STD_buffer_size` 字段应出现于每个基本流的第一个 PES 分组包中, 且当值改变时应再次出现。它们也可能存在于任何其它的 PES 分组包中。

9.7 节目流中系统头编码

在节目流中, 系统头紧跟在包头之后, 可能出现在任何分组中。系统头应出现于节目流的第一个分组中。该节目流的所有系统头中的编码值均应相同。

9.8 受限系统参数节目流

如果一个节目流遵循本节的规范, 则是一个“受限系统参数流”(CSPS)。节目流不受 CSPS 的约束, 可以通过在 7.3.5 中的系统头中定义 `CSPS_flag` 字段来标识一个 CSPS。CSPS 是所有可能节目流的一个子集。

分组包速率

在 CSPS 中, 如果 `packet_rate_restriction_flag` 置'1'且 `rate_bound` 字段(见 7.3.6)中的编码值不大于 4500000bit/s, 或 `packet_rate_restriction_flag` 置'0'且 `rate_bound` 字段中的编码值不大于 2000000bit/s, 分组包进入 P-STD 的最大速率是每秒 300 分组包。对于更高的比特率, CSPS 分组包速率与 `rate_bound` 字段的码值成正比。

具体而言, 若 `packet_rate_restriction_flag` 字段(见 7.3.5)置'1', 则对节目流中所有分组 p 有

$$NP \leq (t(i') - t(i)) \times 300 \times \max[1, R_{max} / (4.5 \times 10^6)] \quad (27)$$

若 `packet_rate_restriction_flag` 字段置 0, 则有

$$NP \leq (t(i') - t(i)) \times 300 \times \max[1, R_{max} / (2.0 \times 10^6)] \quad (28)$$

式中:

$$R_{max} = 8 \times 50 \times rate_bound \text{ bit/s} \quad (29)$$

NP ——相邻 `pack_start_codes` 或 `last_start_code` 与 `AVS_program_end_code`(见表 32 及 7.3.2 的语法)之间的 `packet_start_code_prefixes` 和 `system_header_start_codes` 的数目;

$t(i)$ ——分组 p 的 SCR 中编码的以秒为单位的时间；
 $t(i')$ ——紧跟在分组 p 后的分组 $p+1$ 的 SCR 中编码的以秒为单位的时间。如果是节流中的最后一个分组，则该项表示包含 AVS_program_end_code 末位的字节的到达时间。

解码缓冲区大小

在 CSPS 情形下，系统目标解码器中每一输入缓冲区的大小是有限的。音频基本流和视频基本流有不同的缓冲区大小。

对 CSPS 中的视频基本流，以下规则适用：

BS_n 等于 GB/T 17975.2-2000 或 GB/T 20090.2-2006 规定的视频缓冲区校验器(bbv)大小和一个附加缓冲区 BS_{add} 的总和。 BS_{add} 规定如下：

$BS_{add} \leq \text{MAX} [6 \times 1024, R_{vmax} \times 0.001]$ 字节

式中：

R_{vmax} ——视频基本流中的最大视频比特率。

对 CSPS 中的音频基本流，以下规则适用： $BS_n \leq 4096$ 字节

9.9 传输流中采样率的锁定

在传输流中，音频采样率和系统目标解码器的系统时钟频率之间应有固定的比例关系。同样，视频帧速率和系统时钟频率之间也应有一种特定的比例关系。system_clock_frequency 的定义见 6.2 。视频帧速率在 GB/T 17975.2-2000、GB/T 20090.2-2006 或 GB/T 17191.2-1997 中规定。音频采样速率在 GB/T 17975.3-2002、GB/T 20090.3 或 GB/T 17191.3-1997 中规定。对传输流中所有音频基本流的所有展现单元而言，system_clock_frequency 对实际音频采样率的比例 SCASR 是一常量，且等于下表中音频流在标准采样率下的值。

$SCASR = (\text{system_clock_frequency}) / (\text{audio_sample_rate_in_the_T-STD})$ (30)

符号 X/Y 表示实数除法。

标准音频采样频率 (kHz)	22.05	44.1	24	48	96
SCASR	27 000 000	27 000 000	27 000 000	27 000 000	27 000 000

	22 050	44 100	24 000	48 000	96 000

对传输流中所有视频基本流的所有展现单元而言，system_clock_frequency 对实际视频帧速率的比例 SCFR 是一常量，且等于下表中视频流在标准帧速率下的值。

$SCFR = (\text{system_clock_frequency}) / (\text{frame_rate_in_the_T-STD})$ (31)

标准帧速率 Hz	23.976	24	25	29.97	30	50	59.94	60
SCFR	1 126 125	1 125 000	1 080 000	900 900	900 000	540 000	450 450	450 000

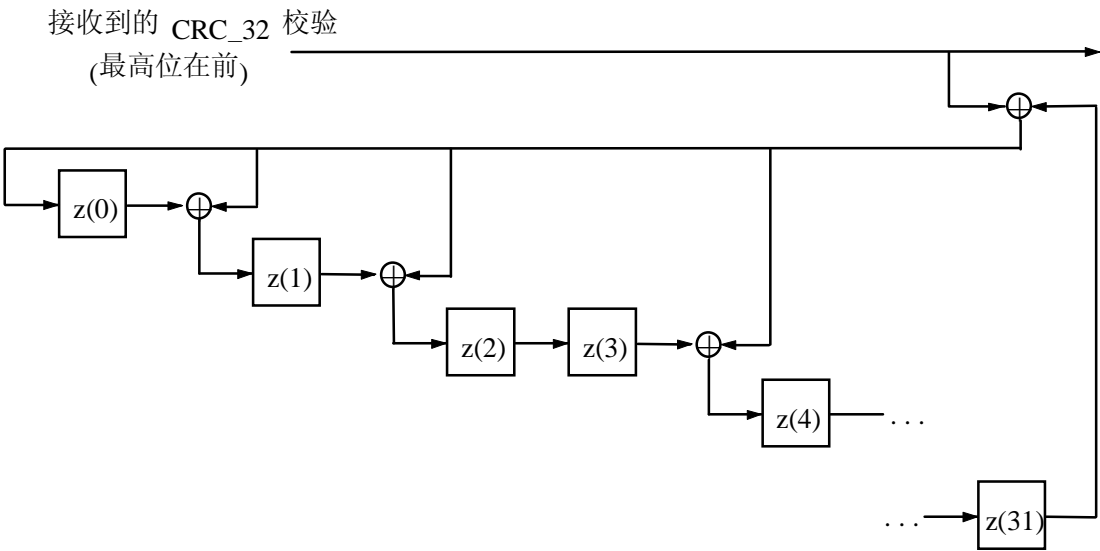
SCFR 的值是精确的。对于 23.976，29.97 或 59.94 帧/秒的标准速率，实际的帧速率与标准速率略有不同。

10 与 GB/T 17975 的兼容性

本部分中所定义的节目流和传输流与 GB/T 17975.1-2000 相兼容。本部分中所定义的节目流解码器与传输流解码器应同时支持 GB/T 17975.1-2000 的解码。

附 录 A
(规范性附录)
CRC 解码器模型

32 位 CRC 解码器模型的规定见图 A.1。



图A.1 32位CRC解码模型

32 位 CRC 解码器在位级操作，含有 14 个加法器和 32 个延迟单元 $z(i)$ 。CRC 解码器的输入与 $z(31)$ 的输出相加，结果作为 $z(0)$ 的输入及其余每个加法器的一个输入。每个剩余加法器的其余输入则来自 $z(i)$ 的输出，而每个剩余加法器的输出连接到 $z(i+1)$ 的输入。 i 的取值为 $i=0, 1, 3, 4, 6, 7, 9, 10, 11, 15, 21, 22$ 和 25 。见图 A.1。

通过以下多项式计算出 CRC 循环冗余码：

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

数据字节在 CRC 解码器的输入端被接收。每一字节都按一次一位的最左位优先的顺序移入 CRC 解码器。例如，对字节 0x01，七个'0'首先进入 CRC 解码器，然后是末位'1'。在对一个段的数据作 CRC 处理之前，每一延迟单元 $z(i)$ 都被设为初始值'1'。初始化之后，包括 4 个 CRC_32 字节在内的所有字节作为 CRC 解码器的输入。当最后一个 CRC_32 字节的末位在和 $z(31)$ 输出相加后移入解码器，即移进 $z(0)$ 后，所有延迟单元 $z(i)$ 的输出均被读出。在无差错情形下，所有 $z(i)$ 的输出都应是 0。在 CRC 编码器中，CRC_32 字段的编码值对此作出了保证。

附 录 B

(资料性附录)

数字存储介质命令与控制(DSM-CC)

B.1 引言

DSM-CC 协议是一个特定的应用协议，其目的在于提供专用于在数字存储介质上管理 GB/T 20090.1 比特流的基本控制功能和操作。本 DSM-CC 是一个网络/操作系统层之上、应用层之下的低层协议。

DSM-CC 在以下意义上是透明的：

它独立于所用的 DSM；

它与 DSM 是位于本地还是远程无关；

它独立作用于 DSM-CC 相交互的网络协议；

它独立于 DSM 所工作的各类操作系统。

B.1.1 目标

许多 GB/T 20090.1 DSM 控制命令的应用都需要对存储在本地或远程的各种数字存储介质上的 GB/T 20090.1 比特流进行存取。不同的 DSM 有其自身特定的控制命令，从而用户要了解特定 DSM 控制命令的不同集合以存取来源于不同 DSM 的 GB/T 20090.1 比特流。这给本部分或 GB/T 17191.1-1997 应用系统的接口设计带来了许多困难。为克服这些困难，本附录提出一个独立于所用特定 DSM 的公用 DSM 控制命令集。本附录仅供参考。ISO/IEC 13818.6 定义了一个更广范围内的 DSM-CC 扩展。

B.1.2 未来的应用

除当前 DSM 控制命令支持的目前应用以外，基于 DSM 命令控制扩展的未来的应用如下：

视频点播

视频节目按用户需求通过各种通信信道提供。用户可以从视频服务器提供的节目表中任选一个视频节目。这类应用可用于旅馆、有线电视、教育机构和医院等。

交互视频服务

在这类应用中，用户提供经常性的反馈信息来控制存储视频和音频的操作。这类服务包括基于视频的游戏、用户控制视频旅游和电子购物等。

视频网络

不同的应用也许需要通过某些类型的计算机网络来互换存储的音/视频信息，用户可以通过视频网络来将 AV 信息送到他们的终端。电子出版和多媒体应用就是这类应用的例子。

B.1.3 优点

制定了独立于 DSM 的 DSM 控制命令，用户就可以在对所用特定 DSM 的细节操作知之不详的情况下，完成 GB/T 20090.1 的解码。

DSM 控制命令码保证了用户能够使用独立于 DSM 和用户界面的相同的语义来展现和存储 GB/T 20090.1 比特流。它们构成了控制 DSM 操作的基本命令。

B.1.4 基本功能

B.1.4.1 流选择

DSM-CC 提供了一个途径来选择一个 GB/T 20090.1 比特流以进行后续的操作，这些操作包括新比特流的创建。该功能包括以下参数：

GB/T 20090.1 比特流的索引(该索引和对一个应用是有意义的名称之间的映射关系超出了当前 DSM-CC 的范畴)；
方式(获取/存储)。

B.1.4.2 获取

DSM-CC 提供途径用于：

展现一指定的 GB/T 20090.1 比特流；

从一给定展现时间开始播放；

设置回放速度(常速或快速)；

设置回放持续时间(即直到一个指定的展现时间，正向播放时到达比特流尾或反向播放时到达比特流始端或发出停止命令)；

设置展现方向(正向/反向)；

暂停；

继续；

改变比特流中的存取点；

停止。

B.1.4.3 存储

DSM-CC 提供途径用于：

在指定时间内存储有效比特流；

停止存储。

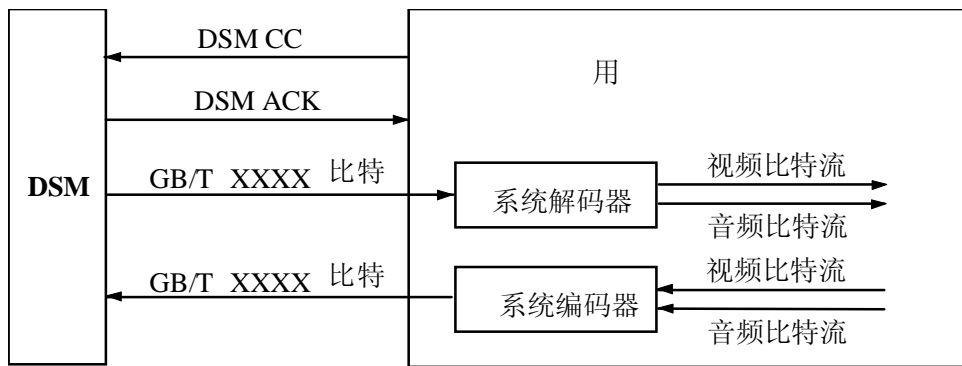
B.2 通用元素

B.2.1 范围

本工作的范围包括开发一个标准以制定一个有用的命令集用于控制存储 GB/T 20090.1 比特流的数字存储介质。这些命令可以独立于特定 DSM，用一种通用方法对数字存储介质进行远程控制，且适用于任何存储在 DSM 上的 GB/T 20090.1 比特流。

B.2.2 DSM-CC应用概述

当前 DSM-CC 语法和语义覆盖了 DSM 应用的单用户，用户系统能获取也可以生成(可选地)GB/T 20090.1 比特流。用于传送 DSM 命令的控制信道在图 B.1 中作为一带外信道描述。如果无法获得一个带外信道，则 DSM-CC 的命令和响应也可被插入 GB/T 20090.1 比特流。

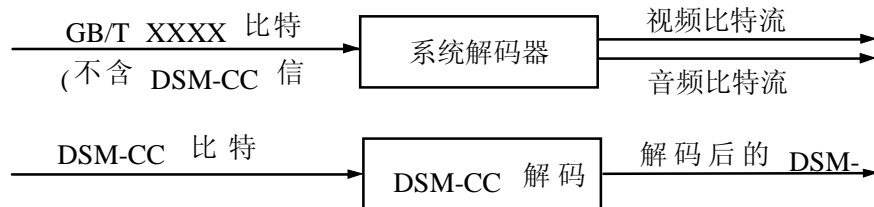


图B.1 DSM-CC应用结构

B.2.3 DSM-CC命令和响应的传输

DSM-CC 按 B.2.2 ~ B.2.9 中定义的语法和语义编码为一个 DSM-CC 比特流。DSM-CC 比特流既可作为独立的流也可在 GB/T 20090.1 系统比特流中传送。

当 DSM-CC 比特流以独立模式传输时，它和系统比特流的关系及解码过程可用图 B.2 来说明。此时，DSM-CC 比特流未被嵌入到系统比特流中。该传输方式可用于 DSM 直接连接到 GB/T 20090.1 解码器的应用，它也能用于 DSM-CC 比特流可被其它类型网络多路复用器控制和传输的情形。

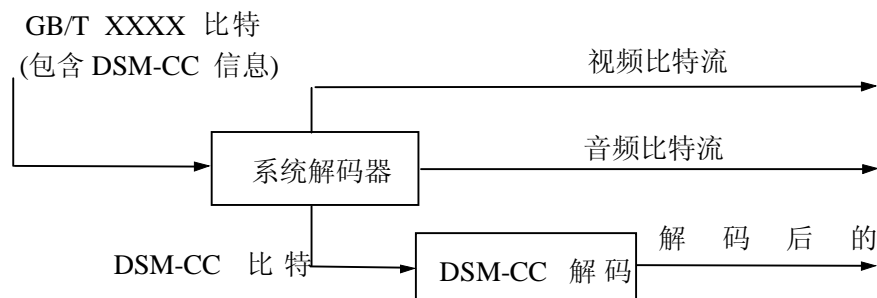


图B.2 DSM-CC比特流作为独立流解码

对有些应用，为使 GB/T 20090.1 系统的一些特征也能作用于 DSM-CC 比特流，要求在 GB/T 20090.1 系统流中传输 DSM-CC。在这种场合，DSM-CC 比特流被系统多路复用器嵌入到系统流中。

系统编码器对 DSM-CC 比特流按以下过程编码。首先，按 6.3.6 中描述的语法将 DSM-CC 比特流分组，成为分组基本流(PES)。接下来，按照传输媒体的要求，PES 分组包复用成为节目流(PS)或传输流(TS)。解码过程与编码过程正好相反，参见图 B.3。

在图 B.3 中系统解码器的输出是视频比特流、音频比特流或 DSM-CC 比特流，DSM-CC 比特流由值为'1111 0010'的 stream_id 字段标识，参见表 17 中 stream_id 的定义。当 DSM-CC 比特流被识别出后，它将遵循 T-STD 或 P-STD 的规则。



图B.3 DSM-CC比特流作为系统比特流的一部分解码

B.3 技术原理

B.3.1 术语和定义

下列术语和定义适用于本附录。

B.3.1.1 数字存储介质命令与控制命令 DSM-CC

由本部分规定，用于控制包含有 GB/T 20090.1 比特流的在本地或远端的数字存储介质。

B.3.1.2 DSM响应 DSM ACK

从 DSM-CC 命令接收器到命令发送器的响应。

B.3.1.3 MPEG比特流 MPEG bitstream

一个 GB/T 17191.1 系统流，GB/T 20090.1 传输流或节目流。

B.3.1.4 DSM-CC服务器 DSM-CC Server

一个本地或远端的用于存储及(或)获取 GB/T 20090.1 比特流的系统。

B.3.1.5 随机存取点 point of random access

GB/T 20090.1 比特流中的一个点，它具有以下特性：对于该比特流中的至少一个基本流，完全包含在比特流中的下一个访问单元'N'，不需要参考前面的访问单元就能被解码，且对于比特流中每一基本流，所有具有相同或稍后展现时间的访问单元都完全包含在后面的比特流中，且不需要参考该点之前的信息就能被 STD 完全解码。存储在 DSM 上的比特流中可存在好几个随机存取点；DSM 的输出可能包括 DSM 自身对存储材料操作而产生的附加的随机存取点(例如，存储量化矩阵以在需要时能生成一个序列头)。一个随机存取点有一个相关 PTS，即访问单元'N'的实际的或隐含的 PTS。

B.3.1.6 当前操作PTS值 current operational PTS value

与当前选定的 GB/T 20090.1 比特流中 DSM 的末访问单元前最后一个随机存取点相关联的实际的或隐含的 PTS。如果该比特流未提供访问单元，DSM 就无法提供对当前比特流的随机存取。这样，当前操作 PTS 值就是 GB/T 20090.1 比特流中的第一个随机存取点。

B.3.1.7 DSM-CC比特流 DSM-CC bitstream

满足 B2.2 语法的一个比特序列。

B.3.2 DSM-CC语法规范

每个 DSM 控制命令以表 B.1 中指定的 start_code 开始。

每个 DSM 控制命令有一个 packet_length 字段以指出该 DSM-CC 分组的字节数。

当 DSM-CC 比特流作为 6.3.6 中定义的 PES 分组包传输时,直到 packet_length 字段为止的字段应和 6.3.6 中规定的相一致。也就是说, DSM-CC 分组被封装为 PES 分组包。PES 分组包起始码是该分组开始处的唯一起始码。

实际的控制命令或响应应跟在 packet_length 字段的末字节之后。

DSM 控制比特流接收器在要求的操作开始或完成后(这取决于收到的命令)给出一个响应流。

任何时候, DSM 均负责提供标准的 GB/T 20090.1 流。可能包括 6.3.6 定义的特技方式操作。

表 B.1 DSM-CC 语法

语 法	位数	助记符
DSM-CC(){ packet_start_code_prefix stream_id packet_length command_id if (command_id == '01'){ control() }	 24 8 16 8	 bslbf uimsbf uimsbf uimsbf

表 B.1 （续）

语 法	位数	助记符
<pre> } else if (command_id == '02') { ack() } }</pre>		

B.3.3 DSM-CC语法规范中各字段的语义

分组包起始码前缀字段 packet_start_code_prefix
24 位码，与之后的 stream_id 共同组成 DSM-CC 分组起始码以标识 DSM-CC 分组比特流的起始。该字段是值为 ‘0000 0000 0000 0000 0000 0001’ (0x000001)的位串。

流标识字段 stream_id
8 位字段，规定比特流类型，在 DSM-CC 比特流中值为 ‘1111 0010’。参见表 18。

分组包长度字段 packet_length
16 位字段，规定紧跟在该字段末字节后的 DSM-CC 分组字节数。

命令标识字段 command_id
8 位无符号整数，规定该比特流为控制流还是响应流，见表 B.2。

表 B.2 command_id 指定值

值	command_id
0x00	禁止
0x01	控制
0x02	响应
0x03~0xFF	保留

B.3.4 控制层

DSM-CC 控制中标志设置的限制

每个 DSM 控制命令中，选择、回放、存储标志中至多一个被设为'1'；如果都没有置位，则该命令被忽略。

每个获取命令中，暂停、继续、停止、播放、跳过标志至多一个被置位；如果都没有置位，则该命令被忽略。

每个存储命令中，记录标志、停止模式至多一个被选择；如果都没有置位，则该命令被忽略。

DSM-CC 控制见表 B.3。

表 B.3 DSM-CC 控制

语 法	位数	助记符
<pre>Control() { select_flag</pre>	1	bslbf

retrieval_flag	1	bslbf
-----------------------	----------	--------------

表 B.3 （续）

语 法	位数	助记符
storage_flag	1	bslbf
Reserved	12	bslbf
marker_bit	1	bslbf
if (select_flag == '1') {		
bitstream_id[31..17]	15	bslbf
marker_bit	1	bslbf
bitstream_id[16..2]	15	bslbf
marker_bit	1	bslbf
bitstream_id[1..0]	2	bslbf
select_mode	5	bslbf
marker_bit	1	bslbf
}		
if (retrieve_flag == '1') {		
jump_flag	1	bslbf
play_flag	1	bslbf
pause_mode	1	bslbf
resume_mode	1	bslbf
stop_mode	1	bslbf
Reserved	10	bslbf
marker_bit	1	bslbf
if (jump_flag == '1'){		
Reserved	7	bslbf
direction_indicator	1	bslbf
time_code()		
}		
if (play_flag == '1'){		
speed_mode	1	bslbf
direction_indicator	1	bslbf
Reserved	6	bslbf
time_code()		
}		
}		
if (storage_flag == '1'){		

表 B.3 （续）

语 法	位数	助记符
Reserved	6	bslbf
record_flag	1	bslbf
stop_mode	1	bslbf
if (record_flag == '1'){ time_code() }		
}		

B.3.5 控制层中各字段的语义

标记位字段 marker_bit

1 位标志，总是置'1'以防止起始码冲突。

保留位字段 reserved_bits

12 位保留字段，供本部分将来用于 DSM 控制命令。除非标准另作规定，否则值总为 ‘0000 0000 0000’。

选择标志字段 select_flag

1 位标志，置'1'时规定一个比特流选择操作；置'0'时不允许存在比特流选择操作。

获取标志字段 retrieval_flag

1 位标志，置'1'时表示将出现一个特定的获取(回放)动作。该操作始于当前操作 PTS 值。

存储标志字段 storage_flag

1 位标志，置'1'时表示将执行一个存储操作。

比特流 ID 字段 bitstream_ID

32 位字段，分三部分编码。各部分组合起来形成一个无符号整数以指定选择哪个 GB/T 20090.1 比特流。DSM 服务器负责将存储在 DSM 上的 GB/T 20090.1 比特流的名字唯一地映射为能用 bitstream_id 表示的一系列数字。

选择方式字段 select_mode

5 位无符号数，指定所需要的比特流操作方式。已定义的方式见表 B.4。

表 B.4 选择方式指定值

代码	方式
0x00	禁止
0x01	存储
0x02	检索
0x03~0x1F	保留

跳跃标志字段 jump_flag

1 位标志。置'1'时表示回放指针跳至一个新的访问单元。新的 PTS 由相对的 time_code 参照当前操作

PTS 值来指定。该功能仅当当前 GB/T 20090.1 比特流为“停止”方式时有效。

播放标志字段 play_flag

1 位标志，置'1'时表示播放一段特定的时间。速度、方向及持续时间是比特流中的附加参数。播放开始于当前操作 PTS 值。

暂停方式字段 pause_mode

1 位代码，表示暂停回放操作，将回放指针保持为当前操作 PTS 值。

恢复方式字段 resume_mode

1 位代码，规定从当前操作 PTS 值开始继续回放操作，仅当当前比特流处于“暂停”状态时有效，比特流将被设为以常速正向播放。

停止方式字段 stop_mode

1 位代码，规定停止比特流传输。

方向指示符字段 direction_indicator

1 位代码，指示回放方向。置'1'时表示正向播放，否则表示反向播放。

速度方式字段 speed_mode

1 位代码，规定速度比率。置'1'时表示常速播放，否则表示快速播放(即快进或快倒)。

录制标志字段 record_flag

1 位代码，规定从终端用户到 DSM 的在一段时间内的或直到收到停止命令(取决于谁先出现)时的比特流录制要求。

B.3.6 响应层

DSM-CC 控制中设置标志的限制

对于每个 DSM 响应比特流，下面指定的响应位中只有 1 位可以置'1'。DSM-CC 响应见表 B.5。

表 B.5 DSM-CC 响应

语 法	位数	助记符
ack() { select_ack retrieval_ack storage_ack() error_ack() Reserved marker_bit cmd_status if (cmd_status == '1' && (retrieval_ack == '1' storage_ack == '1')){ time_code() } }	 1 1 1 1 10 1 1	 bslbf bslbf bslbf bslbf bslbf

B.3.7 响应层中各字段的语义

选择 ACK 字段 select_ack

1 位字段，置'1'时表示 ack()命令用于响应一个选择命令。

获取 ACK 字段 retrieval_ack
1 位字段，置'1'时表示 ack()命令用于响应一个获取命令。

存储 ACK 字段 storage_ack
1 位字段，置'1'时表示 ack()命令用于响应一个存储命令。

错误 ACK 字段 error_ack
1 位字段，置'1'时指示一个 DSM 错误。已定义的错误有正在被检索的流中的 EOF 错(正向播放时遇到文件结束或反向播放时遇到文件开始)及正在被存储流中的磁盘满错。若该位为'1'，则 cmd_status 未定义，当前流仍然被选中。

命令状态字段 cmd_status
1 位标志，置'1'时表示命令被接受，置'0'时表示命令被拒绝。根据收到的命令类型，该字段语义有以下不同：

若 select_ack 被置位且 cmd_status 置'1'，则该字段表明 GB/T 20090.1 比特流已被选中且服务器已准备提供选定方式的操作。当前操作 PTS 值被设为新选中的 GB/T 20090.1 流的第一个随机存取点。若 cmd_status 值为'0'，表示操作失败，没有流被选中。

若 retrieval_ack 被置位且 cmd_status 值设为'1'，这表明对所有检索命令检索操作已被初始化。当前操作 PTS 指针位置由后继的 time_code 指示。

若在 infinite_time_flag 不为'1'的情况下接到 play_flag 命令，就需要发送第二个响应。这将表明到达由 play_flag 命令定义的时间段，播放操作已结束。

若在检索响应中 cmd_status 置'0'，则操作失败。失败原因可能包括无效的 bitstream_ID，跳转超过了文件尾，或一个不被支持的功能，如常速倒放。

若 storage_ack 被置位，则表明存储操作已由 record_flag 开始或已由 stop_mode 命令结束。接下来的 time_code 字段指示了存储的最后一个完整访问单元的 PTS 值。

若由 storage_flag 命令定义的录制操作已经结束，将发送另一个响应，且当前操作 PTS 值在录制后将被报告。

若 cmd_status 在存储响应中置'0'，则操作失败。原因可能是非法 bitstream_ID 或 DSM 不能存储数据。

B.3.8 时间编码

时间编码见表 B.6。

时间编码的限制：

在测到一个实际的或隐含的 PTS，且该 PTS 与操作开始时的当前操作 PTS 值的差值对 2^{33} 取模后值超过这一时间段，则由 time_code 给出的一特定时间段内的正向操作终止。

在测到一个实际的或隐含的 PTS 后，且操作开始时的当前操作 PTS 值与该 PTS 的差值对 2^{33} 取模后值超过这一时间段，则由 time_code 给出的一特定时间段内的反向操作终止。

对控制层()中所有命令，time_code 被指定为相对于当前操作 PTS 值的一段持续时间。

对 ack()层中所有命令，time_code 由当前操作 PTS 决定。

表 B.6 时间编码

语 法	位数	助记符
time_code() { Reserved	7	bslbf

表 B.6 （续）

语 法	位数	助记符
infinite_time_flag	1	bslbf
if (infinite_time_flag == '0') {		
Reserved	4	bslbf
PTS[32..30]	3	bslbf
Marker	1	bslbf
PTS[29..15]	15	bslbf
marker_bit	1	bslbf
PTS[14..0]	15	bslbf
marker_bit	1	bslbf
}		
}		

B.3.9 时间编码中各字段的语义

有限时间标志字段 infinite_time_flag

1 位标志，置'1'时指示一个无限定的时间段，该标志在特定操作持续时间不能预知的应用中被置'1'。

PTS[32..0]字段 PTS[32..0]

比特流中访问单元的展现时间戳。该字段可以为时间的绝对值或以 90 kHz 系统时钟周期为单位的相对值，这取决于相应的功能操作。

附录 C

(资料性附录)

节目特定信息

C.1 传输流中节目特定信息的说明

6.5 包含了有关节目特定信息 PSI 的标准语法，语义和文本。在所有场合下，均要求与 6.5 的限制一致，本附录给出如何使用 PSI 函数的说明信息，同时讨论了如何在实际中运用的例子。

C.2 引言

本部分给出了一种为解复用和展现节目而描述传输流包内容的方法。编码规范通过 PSI 来提供这一功能。本附录讨论 PSI 的应用。

可将 PSI 看作属于下列 4 个表：

- a) 节目关联表(PAT)；
- b) 传输流节目映射表(PMT)；
- c) 网络信息表(NIT)；
- d) 条件访问表(CAT)。

PAT、PMT 和 CAT 的内容在本部分已有描述。NIT 为一专用表，但携带 NIT 的传输流包的 PEID 值在 PAT 中说明。然而，它还必须遵循本部分的段结构。

C.3 功能机制

以上各表都是概念化的，因为它们不必以一种指定形式在解码器中被重新生成。这些结构可能被看作简单表，它们可能在被送到传输流包之前被分割。语法允许表被分为几段，并提供了一个到传输流包有效负载数据的标准映射，从而支持分割操作。同时它还给出了一种按类似格式携带私有数据的方法。这一方法的优越性在于解码器对 PSI 数据和私有数据的处理过程基本相同，从而有助于降低成本。有关在传输流中的优化放置 PSI 的建议可参见附录 D。

每个段均通过以下元素的组合而唯一标识。

- a) 表标识 `table_id`

8 位字段以标识该段属于哪个表。

`table_id` 为 0x00 的段属于节目关联表。

`table_id` 为 0x01 的段属于条件访问表。

`table_id` 为 0x02 的段属于传输流节目映射表。

`table_id` 的其余值可被用户按私有目的分配。

还可以设置滤波器来查看 `table_id` 字段以确认一新段是否属于一感兴趣的表。

- b) 表标识扩展 `table_id_extention`

这个 16 位字段存在于一个段的长版本中。在 PAT 中，它被用于标识流的 `transport_stream_id`（一个有效的用户自定义的标志），以使传输在一个网络内或网络间的传输流互相区别开来。在 CAT 中，该字段目前尚无意义，因此被标记为“保留”以表示它必须被编码为 0xFFFF，但在本部分的后续版本中可能会对它给出一个定义。在一个传输流节目映射段中，该字段含有 `program_number` 从而标识了该段中的数据涉及哪一个节目。在有些场合下，`table_id_extention` 字段也可被用作一个滤波点。

- c) 段号 `section_number`

section_number 字段允许解码器将一个特定表的各段按原先顺序重新组装。本部分并不强制各段按数字顺序传输，但推荐这么做，除非是表的某些段的传送和其余段相比特别频繁，例如出于随机访问的考虑。

d) 版本号 **version_number**

一旦 **PSI** 中描述的传输流特性变化了(例如：额外节目的加入，给定节目的基本流的组合发生变化)，就必须传送带有更新信息的新 **PSI** 数据，因为标志为“当前”的段的最近传输版本必须始终是有效的。解码器需要能够识别最新收到的段是否和已处理 / 存储过的段一致(若一致则该段可被丢弃)。在不一样的情况下，可能表示一个配置变化。这是通过发送一个包含相关数据，且具有和前一个段相同的 **table_id**、**table_id_extension** 和 **section_number** 字段，但 **version_number** 字段取下一个值的段来实现的。

e) 当前下一个指示符 **current_next_indicator**

知道 **PSI** 在比特流的哪一点生效是很重要的。这样，每个段均可被标上数字以表示“现在”(当前)有效或在不久的将来(下一个)有效。这允许在变化发生之前传输一个将来的配置改变信息，使解码器可以为变化作好准备。然而，并不强制提前传输一个段的下一个版本，但如果传输了，它将是该段的下一个正确版本。

C.4 段到传输流包的映射

段直接映射到传输流包，即不先映射到 **PES** 分组包。段不一定在传输流包的起始处开始(虽然有可能会这样)，因为传输流包有效负载数据中的第一个段的开始是由 **pointer_field** 指出的。在 **PSI** 分组包中 **payload_unit_start_indicator** 置'1'，就标志着 **pointer_field** 的存在(在非 **PSI** 的分组中，该指示符标志 **PES** 分组包在传输流包中开始)。**pointer_field** 指向传输流包中第一个段的起始位置，因为语法规则规定传输流包中段与段之间没有空隙，其余段的起始位置均可从第一个段及其后各段的长度中计算出来，所以一个传输流包中决不会有多于一个的 **pointer_field**。

有一点很重要，即任一 **PEID** 值的传输流包中，一个段必须在下一个段允许开始之前结束，否则就无法识别数据属于哪一个段头。若一个段在传输流包的末尾前结束了，但又不便打开另一个段，则提供一种填充机制来填满剩余空间。该机制对分组包中剩下的每个字节均填充为 **0xFF**。这样，**table_id** 就不允许取值为 **0xFF**，以免与填充相混淆。一旦一个段的末尾出现了字节 **0xFF**，该传输流包的剩余字节必然都被填充为 **0xFF**，从而允许解码器丢弃传输流包的剩余部分。填充也可用一般的 **adaptation_field** 机制实现。

C.5 重复率和随机访问

在考虑随机访问的系统中，即使配置没有发生变化，本部分也推荐多次重复传输 **PSI** 段。因为在通常情况下，解码器需要 **PSI** 数据来识别传输流的内容，以能够开始解码。本部分对 **PSI** 段的重复和出现率没有要求。但是，显然重复传输对随机访问应用是有帮助的，虽然导致了 **PSI** 数据所用比特率的增加。如果节目映射是静态或准静态的，它们可能被存于解码器中以加快数据存取，而不是等待它重新传输。所需存储量和所期望信道的获取时间之间的折中由解码器制造商决定。

C.6 节目是什么

节目的概念在本部分中有精确定义(参见 3.42 节目参考时钟[系统])。对一个传输流，时间基准由 **transport_rate** 定义，这就在传输流中有效地创建了一个虚拟信道。

注意这和广播中广泛使用的定义不同。广播中一个“节目”是指不仅有共同时间基准而且有共同起止时间的基本流集合。一个“广播节目”序列(在本附录中称为事件)通过使用同一个 **program_number** 来创建一个“广播协定”的 **TV** 频道(有时称为一个服务)，可在传输流中被顺序地传送。

事件描述可在 **private_section()** 中传输。

一个节目可由 **program_number** 标识，它只在传输流中有意义。**program_number** 是一个 16 位无符号整数，从而允许 65535 个单独节目存在于一个传输流中(**program_number** 0 被保留用作标识 **NIT**)。当解

码器接到多个传输流时(例如在有线网络中), 为了成功地分离出一个节目, 解码器必须同时收到 `transport_stream_id`(以寻找正确的复用流)和 `program_number`(以寻找复用流中的正确节目)的通知。传输流映射可以由可选网络信息表完成。注意 NIT 可能存储于解码器非易失存储器中, 以减少信道获取时间。在此场合下, 它的传输次数只需保证及时地支持解码器初始化设置操作。NIT 的内容是专用的, 但至少应具有最小段的结构。

C.7 `program_number`的分配

在所有场合都把共享参考时钟的所有节目编为一组可能会带来不便。可以设想一个仅具有唯一共用 `transport_rate` 集的多服务传输流。然而, 通常情况下, 一个广播也许宁可在逻辑上把传输流分为若干个节目, 其中, `clock_recovery_PEID` 总是相同的。这种把节目成份分为伪独立节目的方式有许多用途, 以下举出两例:

a) 各个市场的多语种传输

一个视频流可能伴有不同语种的多个音频流。建议包含一个与每个音频流相关的 `ISO_639_language_descriptor` 实例以能够选择正确节目和音频。可以有带有不同 `program_number` 的节目定义。其中, 所有节目参照同一视频流和 `clock_recovery_PEID`, 但有不同的音频 `PEID`。当然, 也可以把视频流和所有音频流列为一个节目, 但不能超出 1024 字节的段大小限制。

b) 极大节目定义

段长度的最大限制 1024 字节(包括段头和 `CRC_32`)。这意味着, 单个节目定义不能超过这个长度。在大多数情况下, 即使每个节目元素有几个描述子, 这一长度是足够了。然而, 有时会遇到超过这一限制的高比特率系统。故常常需要一种对流参考进行分割的方式, 使得它们不必列在一起。一些节目成份可以在多个节目中引用, 另一些则仅存在于单个节目而不在其它节目中出现。

C.8 典型系统中PSI的使用

一个通信系统, 尤其是广播应用中, 可能包含一些单独的传输流。4 个 PSI 数据结构中的任何一个都可能出现系统中的任一传输流中。通常, 必须有一个节目关联表的完整版本以列出 TS 中的所有节目, 以及一个包含传输流中所有节目完整定义的完整的传输流节目映射表。如果有一些流被加扰了, 则也必须有一个条件访问表以列出相关 EMM 流(授权管理信息)。NIT 表的有无完全是可选的。

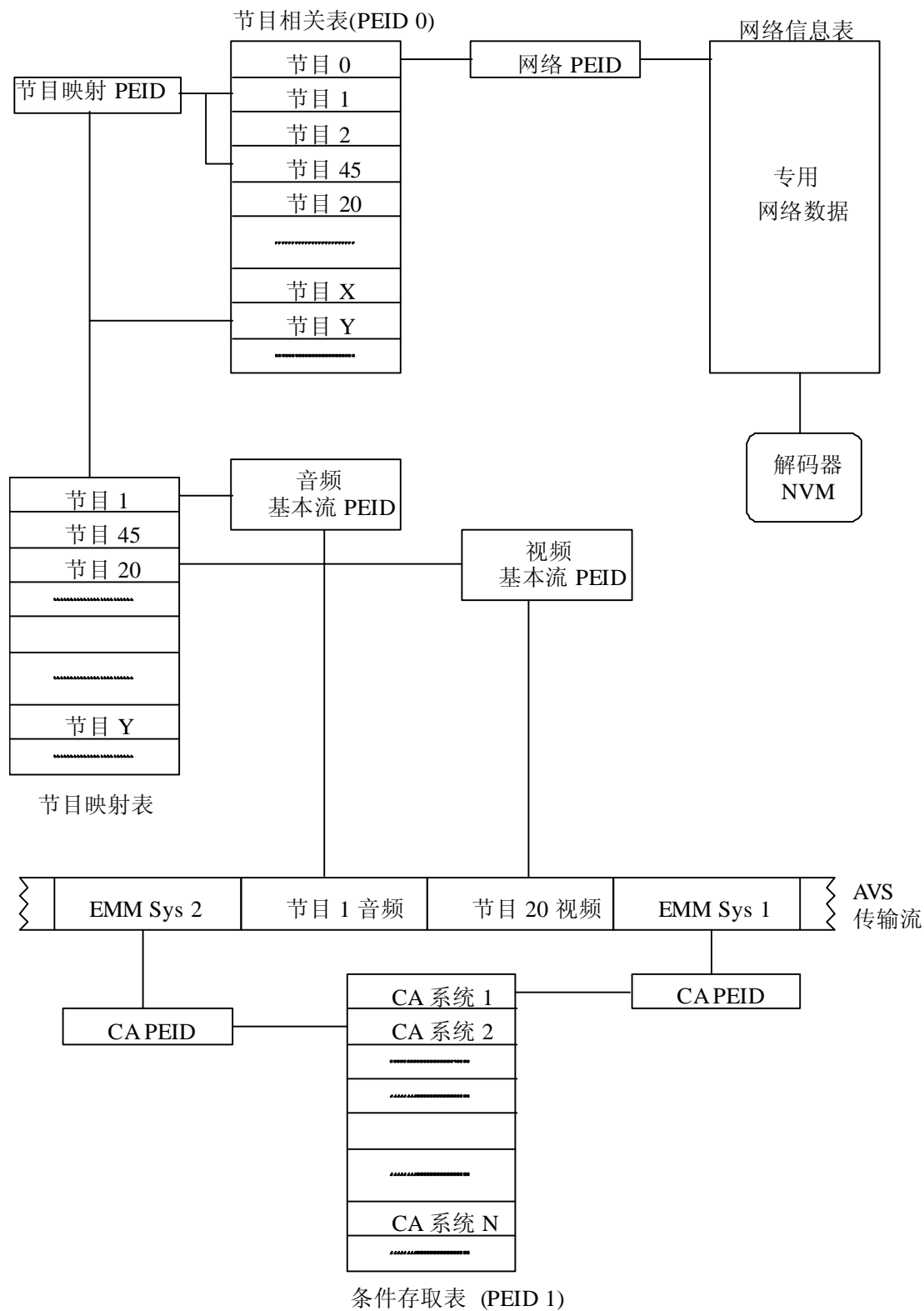
PSI 表经由上述段结构映射到传输流包。每个段在头部分有一个 `table_id` 字段, 允许来自 PSI 表的段和 `private_sections` 中的私有数据混合在有相同 `PEID` 值的传输流包中或甚至在同一个传输流包中。但须注意到, 在有相同 `PEID` 的传输流包中, 在下一段开始之前, 前一段必须完成传输。然而, 只有被标记为含有 TS PMT 段的传输流包或 NIT 传输流包, 这才是可能的。因为私有段不能映射到 PAT 或 CAT 传输流包。

所有 PAT 段都要求被映射到 `PEID` 值为 0x0000 的传输流包且所有 CA 段都要求被映射到 `PEID` 值为 0x0001 的传输流包。PMT 段可能被映射到具有用户选择 `PEID` 的分组, 对 PAT 中的每个节目均作为 `PMT_PEID` 列出。类似的, 带有 NIT 的传输流包的 `PEID` 也是用户可选的, 若存在 NIT, 则必须由 PAT 中的入口 '`program_number == 0x00`' 指出。

任何 CA 参数流内容是完全私有的, 但 EMM 和 ECM 也必须被传送到传输流包以和本部分一致。私有数据表可用 `private_section()`语法来传送。例如, 该表可用在广播环境中以描述一个服务, 一个将要发生事件、广播时刻表和相关信息。

C.9 PSI结构的关系

图 C.1 显示了四个 PSI 结构和传输流关系的示例。还可能有其它例子, 但该图中显示了最基本的关联。



图C.1 节目与网络映射关系

以下几条中，对每个 PSI 表都作了说明。

C.9.1 节目关联表

每一传输流必须含有一个完整有效的节目关联表。节目关联表给出了 program_number 与携带节目定义的传输流包的 PEID(PMT_PEID)的对应关系。PAT 在映射到传输流包之前可能被分为至多 255 个段。

每段携带整个 PAT 的一部分。划分有助于减少误差条件下的数据丢失。也就是说，包丢失或位错误可以限制在 PAT 的较少段中，而使其余段仍可被接收和正确解码。若所有 PAT 信息被放到一个段中，则一个小错误，例如，导致一个位发生改变，就可能导致整个 PAT 的丢失。但在段长度不超过 1024 字节的最大长度限制时这还是允许的。

节目 0 被保留并用以指定网络 PEID。这是一个指向携带网络信息表的传输流包的指针。

节目关联表的传输是不加密的。

C.9.2 节目映射表

节目映射表给出了节目号与包含该节目号的各节目元素之间的映射。该表存在于含一个或多个专门选择的 PEID 的传输流包中。这些传输流包可能包含其它由 table_id 字段定义的专用结构。使引用不同节目的各个 TS PMT 段携带于有相同 PEID 值的传输流包中是可行的。

本部分至少需要以下节目标识：节目号、clock_recovery PEID、流类型及节目元素 PEID。用于节目或基本流的附加信息可通过使用描述子构造来传送。参见 C.9.6。

私有数据也可以在标记为含传输流节目映射表的各段的传输流包中传送。这是由使用 private_section() 来实现。在 private_section() 中，version_number 和 current_next_indicator 是表示单个段中的值还是适用于作为更大的私有表的组成部分的各个段的值，这由应用程序决定。

注 1: 含有 PMT 的传输流包的传输是不加密的。

注 2: 可以包含在 TS_program_map_section() 中的私有描述子中传送事件信息。

C.9.3 条件访问表

条件访问(CA)表给出了一个或多个 CA 系统及其 EMM 流和与之相关的任何特定参数间的联系。

注： 包含 EMM 和 CA 参数的传输流包的(私有)内容，如果有的话，一般是加密(加扰)的。

C.9.4 网络信息表

NIT 的内容是专用的且本部分未对其作出规定。通常，它包含用户选择服务与 transport_stream_id、信道频率、卫星转发器频点、调制特性等的映射。

C.9.5 私有段 private_section()

private_section() 可以以两种基本形式出现，短版本(仅有直到且包括 section_length 的字段)或长版本(直到且包括 last_section_number 的所有字段均出现，在私有数据字节后面有 CRC_32 字段)。

private_section() 可以出现在被标记为 PMT_PEID 的传输流包中或具有其它 PEID 值（包括分配给 NIT 的 PEID）的传输流包中，仅含有 private_section() 传输流包中。若携带有 private_section() 的传输流包的 PEID 被标志为携带有 private_section 的 PEID (stream_type 值为 0x05)，则 private_section 将仅会出现在具有该 PEID 值的传输流包中。各段可以是长类型或短类型。

C.9.6 描述子

本部分定义了一些标准描述子，还可以定义更多。所有描述子均有一种共同的格式：{ 标签，长度，数据 }。任意定义的私有描述子必须遵循该格式。这些私有描述子的数据部分是用户定义的。

在 TS PMT 的段中，有一个描述子(CA_descriptor()) 用于指示与节目元素相关的 ECM 数据的位置(传输流包的 PEID 值)。在 CA 段中它指向 EMM。

为扩展可用的私有描述子的数量，可采用以下机制：专门定义一个私有 descriptor_tag 以用来构造一个合成描述子，这使得需要进一步专门定义一个子描述子作为私有描述子的私有数据字节的首字段。描述结构见表 C.1 和 C.2。

表 C.1 Composite_descriptor

语 法	位数	助记符
composite_descriptor() { Descriptor_tag (privately defined)	3	uimsbf

表 C.1 （续）

语 法	位数	助记符
Descriptor_length <pre> for(i=0;i<N;i++) { sub_descriptor() } </pre>	3	uimsbf

表 C.2 Sub_descriptor

语 法	位数	助记符
<pre> Sub_descriptor(){ sub_descriptor_tag sub_descriptor_length for(i=0;i<N;i++) { Private_data_byte } } </pre>	8 8 8	uimsbf uimsbf uimsbf

C.10 带宽利用和信号采集时间

所有 GB/T 20090.1 比特流的实现都应对 PSI 信息作合理带宽要求，且需要考虑随机访问的应用中，还应提供快速信号采集。本章分析该问题并给出一些广播应用实例。

传输流的基于包的特性使得 PSI 信息以较小的粒度分布在多路复用的数据中。这为 PSI 的构造和传输提供了极大的灵活性。

一个实际解码器的信号采集时间取决于许多因素，包括：FDM 调谐解调时间、解复用时间、序列头和 I 帧出现频率、以及密钥的获取和处理速度。

本章检查了 6.4.4 和 6.4.9 PSI 语法对比特率和信号采集时间的影响。假定不必在每个节目变化时动态接收 CAT。该假定对 EMM 流也成立。这是因为这些流不包含用于节目元素(加扰)加密的快速变化的 ECM 成份。

在以下讨论中，采集和处理 ECM 的时间也被忽略。

表 C.3 和表 C.4 给出了一系列传输流条件下带宽的使用值。该表的一个轴表示包含于单个传输流中的节目数。另一个轴表示 PSI 信息在传输流中的传输频率。

这一频率将是 PSI 结构信号采集时间的关键因素。

这两个带宽利用表均假定只提供最小节目映射信息。这意味着所提供的 PEID 值和流类型是不带附加描述子的。示例中所有节目均包含两个基本流。节目关联的长度为 2 个字节，而最小节目映射的长度为 26 字节。还有些有关版本号，段长度等信息的额外开销。这在适中长度(几百字节到 1024 字节)的分段的总 PSI 比特率中占 1%~3%，因此这里将其忽略。

以上假设允许 46 个节目关联映射到一个 PAT 传输流包(如果不出现适应字段)。类似的，7 个 TS_program_map_section 封配到一个传输流包中。可以注意到，为使“丢弃/添加”简单化，可能在每个 PMT_PEID 传输流包中，只有 1 个 TS_program_map_section。但这可能引起不希望出现的 PSI 比特

率增加。

表 C.3 和表 C.4 使用 25Hz 的频率，产生一个约 80ms 的最坏信号采集时间。这只在所要求的 PAT 数据“刚刚错过”，且一旦 PAT 被采集并解码，所需的 PMT 数据也“刚刚错过”。这种双重最坏情况是 PAT 结构所引入的额外的迂回层次的一个缺点。该方式提供对“丢弃/添加”再复用操作的优越性是对该缺点的一种补偿。同时这种缺点的不良影响可以通过协调传输相关 PAT 和 PMT 传输流包来减少。

表 C.3 节目关联表占用带宽 (单位: bit/s)

PAT表频率信息 s ⁻¹	每个传输流的节目数					
		1	5	10	32	128
	1	1504	1504	1504	1504	4512
	10	15040	15040	15040	15040	45120
	25	37600	37600	37600	37600	112800
	50	75200	75200	75200	75200	225600
	100	150400	150400	150400	150400	451200
注： 由于46个program_association_section被装配为一个传输分组， 该表中的数目直到最后一列才变化。						

表 C.4 节目映射表带宽利用(单位: bit/s)

PMT表频率信息 s ⁻¹	每个传输流的节目数					
		1	5	10	32	128
	1	1504	1504	3008	7520	28576
	10	15040	15040	30080	75200	285760
	25	37600	37600	75200	188000	714400
	50	75200	75200	150400	376000	1428800
	100	150400	150400	300800	601600	2857600

以 25Hz 的 PSI 频率可以建立以下例子(所有实例均为各种数据连接、FEC、CA 和路由开销留下了足够的余地):

6MHz CATV 信道

5 个 5.2 Mbit/s 的节目: 26.5 Mbit/s(包括传输开销)

完整 PSI 带宽: 75.2 Kbit/s

CA 带宽 : 500 Kbit/s

总 GB/T 20090.1 带宽: 27.1 Mbit/s

PSI 开销: 0.28%

OC-3 光纤信道(155Mbit/s)

32 个 3.9 Mbit/s 的节目: 127.5 Mbit/s (包括传输开销)

完整 PSI 带宽: 225.6 Kbit/s

CA 带宽: 500 Kbit/s

总 GB/T 20090.1 带宽: 128.2 Mbit/s

PSI 开销 : 0.18%

C-波段卫星发送器

128 个 256 kbit/s 的节目: 33.5 Mbit/s (包括传输开销)

完整 PSI 带宽: 826.4 Kbit/s

CA 带宽: 500 Kbit/s

总 GB/T 20090.1 带宽: 34.7 Mbit/s

PSI 开销: 2.4%(实际上, 若每个节目仅使用一个 PEID, 则会更低。)

正如所期望的, PSI 开销百分比在低速率服务中相对较大, 因为每个传输流中可能有更多服务。但在所有场合中其值均不太大。可以用更高的传输率(高于 25Hz)来传输 PSI 数据以减小对信道采集时间, 同时仅适度地增加比特率需求。

附 录 D

(资料性附录)

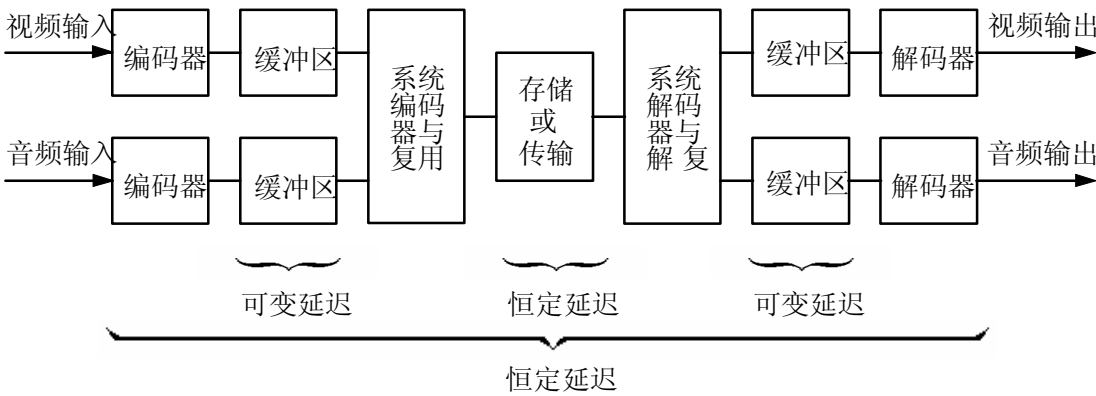
本部分的系统时间信息时钟恢复模型及应用指南

引言

GB/T 20090.1 系统规范包括一个对数字音 / 视频信号的组合进行采样、编码、编码器缓冲、传输、接收、解码器缓冲、解码及展现的特定时钟恢复模型。该模型直接嵌入符合 GB/T 20090.1 数据流的语法和语义需求规范中。若一个解码系统接收到一个按时钟恢复模型正确传递的规范比特流，则实现一个输出正确同步的高质量音 / 视频信息的解码器将是很简单的，但在这方面并无标准的要求。解码器可以用某种途径实现，只要能提供高质量的展现输出即可。在数据未以正确的时刻被传递给解码器的应用中，有可能产生预期的展现输出，但这一功能常常不被保证。本附录详细描述了 GB/T 20090.1 的系统时钟恢复模型，并给出了一些参考建议以实现适用于某些典型应用的解码系统。

D.1 时钟恢复模型

GB/T 20090.1 系统中包括一个时钟恢复模型。在该模型中，所有进入解码器的数字图像和音频样本均在一个恒定的端到端延迟后在解码器输出端顺序播放。这样，样本速率，即视频帧速率和音频采样率，在解码器中和编码器中严格相等。该时钟恢复模型见图 D.1。



图D.1 恒定延迟模型

从图 D.1 看出，在该模型中，从编码器输入到解码器输出的延迟是恒定的¹⁾，但在每个编码器和解码器的缓冲区中的延迟则是可变的。这种延迟可变性不仅存在于同一个基本流路径中的不同缓冲区，也存在于音 / 视频路径的各自缓冲区中。所以，代表音 / 视频的编码位在复用流中的相对位置并不指示同步信息。它们仅受 STD 模型的限制，因此解码缓冲区必须正确地工作；因而，代表那些要求同时展现的声音和图像的编码的音频和视频，在编码比特流中可能有长达 1s 的时间差，该值是 STD 模型所允许的解码器缓冲区延迟的上限。

1) 整个系统的恒定延迟对于正确地同步是必须的，然而也可能有一些偏差。网络延迟被看作是恒定的。轻微的偏差是允许的，且网络调节可以允许较大的网络延迟。这两者均在后面讨论。
编码器的音 / 视频采样率彼此相差很大，也未必存在确切的固定的关系，这些都取决于复用流是节目

流还是传输流，以及节目流中的 `system_audio_locked` 和 `system_video_locked` 标志是否置位。一个音频样本块(一个音频展现单元)的持续时间往往和视频图像的持续时间不同。

在编码器中有一个单一的公用系统时钟，该时钟用于创建指出音频和视频的正确的展现时间的时间戳，也用于创建指出在采样间隔时系统传输率的采样值。指出音频和视频展现时间的时间戳称为相对显示时间戳(`relative_display_time`)，而指出系统传输率的采样值称为传输率(`transport_rate`)。正是编码器中的公用系统时钟、由它所创建的时间戳和传输率，以及解码器中重新生成的时钟、时间戳和传输率的正确使用，才使本地解码器能够同步操作。

编码器的实现不一定严格遵循该模型，但实际的编码器、存储系统、网络和一个或多个复用器产生的数据流必须严格遵守该模型。(在具体应用中，数据传输可能有一定的偏差)。因此，在本附录中术语“编码器系统时钟”指的是本模型描述的实际公用时钟或与之等价的功能，而不管它是如何实现的。因为整个系统的端到端延迟是恒定的，所以音 / 视频展现严格同步。系统比特流构造受到一定的限制，以使得当他们被遵循该模型且具有适当大小的缓冲区的解码器解码时，解码器缓冲区保证不发生上溢和下溢，除非那些允许下溢的特定情况。

为了使解码器系统产生精确的延迟以保证整个系统的端到端延迟恒定，解码器就必须有一个在工作频率和绝对瞬时值上和编码器精确匹配的时钟。传送编码器系统时钟的信息编码在 `transport_rate`、`packet_count` 和 `relative_display_time` 中；下面对此功能加以详细说明。

按此模型实现的，可以以恒定速率一次性展现音频采样和视频图像（特定编码除外），且保证解码缓冲区按照本模型正确工作的解码器，在本附录中，被称为精确时钟恢复解码器或能产生精确时钟恢复输出的解码器。本部分并不要求解码器的实现按此模型展现音频或视频。有可能构造出没有恒定延迟或不是对每一个图像或音频样本都展现一次的解码器。但在这些实现中，音 / 视频信息的同步可能是不精确的，解码器缓冲区的动作也可能不遵守参考解码器模型。避免解码缓冲区上溢是很重要的，因为它可能导致对后续解码过程产生严重后果的数据丢失。本附录主要包括这种精确时钟恢复解码器的操作及实现这些解码器的一些可行的选择。

D.2 音频和视频展现同步

在本部分的系统数据编码中包含了涉及视频图像和音频样本块的展现和解码的时间戳，这些图像和块被称为“展现单元”，简称 PU。代表 PU 且包含在 GB/T 20090.1 比特流中的编码比特的集合，被称为访问单元，简称 AU。视频及音频访问单元分别简称 VAU 和 AAU。在 GB/T 20090.3 中，术语“音频帧”根据其上下文与 AAU 或 APU 的含义相同。VPU 指一帧图像，VAU 指一帧编码图像。

有一些，但并不要求全部，AAU 和 VAU 与它们的由 `relative_display_time` 导出的 `display_time` 有关。一个 `display_time` 指出了解码与该 `display_time` 相关的 AU 所得的 PU 展现给用户的时间。音频 `relative_display_time` 和视频 `relative_display_time` 都由自称为系统时钟或 STC 的共同时钟的采样值计算得到的。通过由数据流中包含的音/视频 `relative_display_time` 导出的 `display_time` 的正确值，以及根据共同 STC 得到的正确的 `display_time` 所指示的时刻展现音/视频 PU，就可以在解码系统中实现音频和视频展现的正确同步。尽管 STC 并非该标准的一个标准组成部分，且等价的信息在本部分中通过 `system_clock_frequency` 等术语来传递，但 STC 对于解释时钟恢复模型是一个重要和方便的概念，且实现一个以某种形式包含 STC 的编码器和解码器通常是可行的。

`display_time` 对于传达在音 / 视频之间精确的相对时间信息很有必要，因为音频和视频 PU 的持续时间常有很大差别且根本无关，例如，具有 1152 个 44.1 kHz 采样率的音频 PU 持续时间约 26.12ms，而帧速率为 29.97kHz 的视频 PU 持续时间为 33.76ms。一般来说，APU 和 VPU 当时的边界很难吻合。分离的音频和视频 `display_time` 在对二者持续时间及间隔的特定的关系不作要求的情况下，提供了指示音频和视频 PU 之间精确关系的信息。

`relative_display_time` 字段的值是根据 STD 定义的，它为所有系统比特流提供了一个标准限制。STD 是一个理想解码器的数学模型，它精确规定了进入和流出解码器缓冲区的所有位的动作，对比特流的基本语义约束是保证 STD 中的缓冲区不发生上溢和下溢，除非在特定情况下的下溢。在 STD 模型中，

虚拟解码器总是和数据源严格同步,音/视频解码和展现也严格同步。在保持精确和一致的前提下,STD 相对于解码器的物理实现而言,作了某些简化,以清楚地阐述其规范及方便各种解码器的实现。特别地,在 STD 模型中,所有对解码器中的比特流的操作都是瞬时的,只有位在解码缓冲区中的时间是一个明显的例外。在一个实际解码系统中,音视频解码器的单独操作不是瞬时完成的,在设计实现中必须考虑其延迟。例如,若视频图像严格按一个图像展现时间间隔 $1/P$ 来解码(P 为帧频率),而压缩数据以速率 R 到达解码器,则把与每个图像相关的位完全移出将 `display_time` 字段指出的时间延迟 $1/P$,视频解码缓冲区必须比 STD 模型规定的大出 R/P 。视频展现同样也比 STD 延迟,PTS 应该被适当地处理。由于视频的延迟,相应造成音频解码和展现的相同量的延迟以满足正确的同步要求。解码器中,音频和视频解码和展现的延迟可通过诸如在解码器中使用 `display_time` 时给 `display_time` 加一个常量来实现。

STD 和实际解码器实现的另一个不同点在于 STD 模型中明确地假定最终的视频和音频输出被瞬时展现给用户而没有进一步的延迟。但实际应用中未必如此,尤其是 CRT 显示器,且这一延迟也应在设计时加以考虑。编码器需要对音频和视频编码以使数据在 STD 中解码时可实现正确同步。输入和音/视频采样中的延迟,诸如视频摄像机光学充电集成等,都必须加以考虑。

在 STD 模型中,假设正确同步,时间戳和缓冲区行为都被检测以在这个假设下作为判断比特流有效性的条件。当然,在实际解码器中并不能自动地同步,尤其是在开始处和有时钟抖动的场合。解码器设计的目标之一是精确的解码器时钟恢复。解码器时钟恢复不准将影响到缓冲区的动作。本附录的以下各章将对此主题作进一步论述。

STD 包括 `relative_display_time` 字段。在 STD 模型中,当显示模块可以接收一帧解码数据并且 EB 中接收到完整的一个 AU 数据时,AU 从缓冲区中被抽取并解码。对于非 `low_delay` 序列中的 I 和 P 图像,在编码比特流中对视频图像作了重排。在重排序场合下,视频解码器中使用一个暂时延迟缓冲区来存储恰当的 I 图像或 P 图像直到它们被展现。

因为不是每个 AAU 和 VAU 都需要 `relative_display_time` 值,故解码器应在适当的地方插入未编码的值。在每个基本音/视频流中每隔不超过 700ms 的间隙就需要 `relative_display_time` 值。这些间隙以展现时间计量,也就是说,是在具有同一数据流的 `relative_display_time` 字段值中,而不是按照字段被传输和接收的时间计量。注意到在起始时间以及其它任何时间,`relative_display_time` 字段自身并不指示解码缓冲区的正确占用度,它们同样也不指示在解码开始前接收数据流初始位的延迟总和。通过结合 `relative_display_time` 的功能以及正确地恢复时钟(在下面讨论)来获得这一信息。在 STD 模型中,也就是在解码器中,解码缓冲区的动作由 `transport_rate`,`relative_display_time` 以及显示模块的状态决定,假定数据传送方式与时钟恢复模型吻合。这一信息指出了编码数据在解码缓冲区中的时间。编码数据缓冲区中的数据总量并未明确规定,这一信息也是不必要的,因为时序已完全规定。还应注意数据缓冲区的占用度可能随时间变化有很大不同,解码器无法对此作出预测,除非适当地利用时间戳。

为使音频和视频 `display_time` 正确指向公用 STC,解码系统中必须有一个正确恢复的公用时钟。下一章详述这一问题。

D.3 解码器中的系统时钟恢复

在 GB/T 20090.1 系统数据流中,除 `relative_display_time` 字段外还有 `transport_rate` 字段。这些参考基于系统时钟取样,既适用于编码器,也适用于解码器。它们的精度为比特/秒,在传输流中每隔至多 100ms 的时间间隔出现一次。从而在解码器中能利用它们来实现对任何应用均为足够精度的时钟控制环的重建。

传输流中的 `transport_rate`,提供了一个节目的参考传输率,一个传输流中,可能含多个节目,每一个都可能有独立的时间基准和分离的 `transport_rate` 集。

`transport_rate` 字段指出了当 `transport_rate` 字段被解码器接收时系统传输率的正确值。因为 `transport_rate` 占据了多于一个字节且系统数据流被定义为字节流。故 `transport_rate` 被定义为包含上一次 `transport_rate` 之后第一位的字节到达系统目标解码器输入端到包含当前 `transport_rate` 最后一位的字节到达系统目标

解码器输入端之间的传输率。在数据源可被解码器控制的应用中，例如一个接在本地的 DSM，可以让解码器有一个自主的 STC 频率，从而不必恢复 STC。下文着重讲述 STC 时钟必须服从于接收到的 transport_rate 的场合。

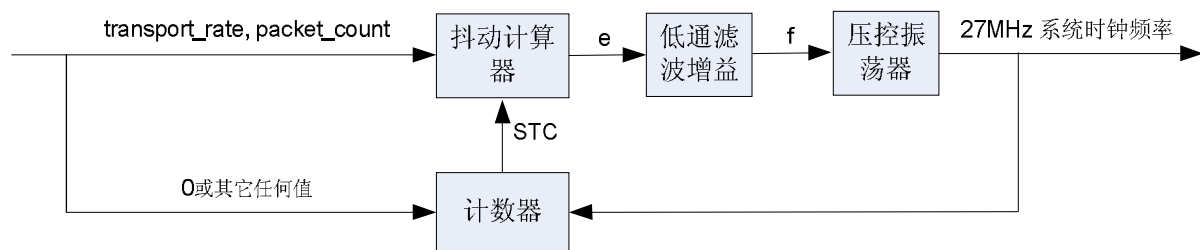
在一个正确构造和传输的 GB/T 20090.1 数据流中，每个 transport_rate 都以其值指示的传输率精确地到达解码器。在此处，“时刻”意味着 STC 的正确值。在概念上，STC 值与 transport_rate 被存储或传送时编码器的 STC 值是一致的。但编码未必实时进行，数据流在初始产生后也可能已被修改，且通常编码器或数据源可能以多种方式实现以保证编码器的 STC 是理论值。

如果解码器时钟频率与编码器精确匹配，则音 / 视频信号的解码和展现速率将与编码器一致，端到端延迟将是恒定的。此后解码器的 STC 将与编码器匹配，不需要作进一步调整。除非存在时间不连续现象，如节目流的末端或传输流中出现不连续指示符。这里匹配指相差一个恒定的常量

实际上，解码器的自由运行系统时钟频率并不与编码器匹配，编码器的时钟频率被采样并由 transport_rate 值指出。通过使用接收到的 transport_rate，解码器的 STC 能使它的时钟服从于编码器。使解码器的时钟服从于所接收的数据流的原型的实现方法是通过锁相环(PLL)。根据具体的应用需求，可采用多种形式的锁相环或其它方法。

此处画出并描述了一个在解码器中恢复 STC 的简单的 PLL。

图 D.2 图示了一个典型 PLL，除了参考和反馈术语是数字的(STC 和 transport_rate, packet_count)值，而不是诸如边界之类的信号事件。



图D.2 使用PLL恢复STC

根据一个新时间基准，即一个新节目的要求，STC 被初始化为 0 或其它任何值，其后是 PLL 作闭环操作。

PLL 的闭环可以这样实现。在第 i 个 transport_rate 到达解码器的时刻，当前 STC 值被采样并和 packet_count 一起保存在抖动计算器中的缓冲区中。在第 i+1 个 transport_rate 到达解码器的时刻，将 (packet_count(i+1)-packet_count(i))×1504×system_clock_frequency/transport_rate(i+1)和 STC(i+1)-STC(i) 作比较，差值为一个数值，一个部分以 90 kHz 为单位，另一个部分以该频率的 300 倍，即 27 MHz，为单位。该差值被线性化到单个的数值空间，通常以 27MHz 为单位，称为“e”，该循环中的误差项。e 值序列输入到按照应用需求设计的低通滤波器及增益。该阶段的输出是一个控制信号“f”，用于控制电压控制晶振(VCO)的瞬时频率。VCO 的输出是一个标称值为 27 MHz 的振荡信号，在解码器中用作系统时钟，该时钟输入到计数器中以生成当前 STC 值。STC 值包括一个由被 300 所除而产生的 27 MHz 的扩展和由对 90 kHz 计数结果的 33 位计数器得到的 90 kHz 的基准值。这 33 位，即 STC 输出中的 90 kHz 部分在需要时被用来与 display_time 相比较。整个 STC 也作为减法器的反馈输入。

连续 transport_rate 间限定的最大时间间隔允许设计和构造稳定的 PLL。PLL 带宽的上限受这一时间间隔的限制。如下文所述，在许多应用中所需 PLL 的带宽极窄，故这一限制对解码器设计和性能影响不大。

若 VCO 的自运行或初始频率与正确值即编码器系统时钟频率足够接近，则一旦 STC 被正确初始化，在 PLL 到达定义的锁定状态前，解码器就能令人满意地操作。对一个给定的解码器 STC 频率，该频率与 transport_rate 中的频率相差一个限定值且在解码器应用所需的绝对频率范围之内，若没有 PLL，

则编码器和解码器 STC 频率的失配将造成解码器缓冲区占用度渐进地不可避免地递增或递减，最终导致缓冲区上溢或下溢。因而解码器 STC 锁定到编码器前所允许的时间长度取决于附加解码缓冲区和延迟的大小。

如果解码器收到的 `transport_rate` 值和恢复时钟反映了编码器中恒定频率 STC 的正确采样，则误差 e 将在循环到达锁定状态后收敛为一个常量。在数据从编码器到解码器的存储和传输延迟恒定或延迟不恒定，但 `transport_rate` 值已被修正以反映延迟变动从而存储或传输与恒定延迟相等效时，`transport_rate` 值是正确的。随着 e 值收敛为一个常量，VCO 瞬时频率的变化在循环锁定后趋于 0，VCO 会有很小的抖动或频率回转。在循环锁定的过程中，VCO 频率的变化率，回转率可通过低通滤波和增益的设计加以严格控制。通常，VCO 变化率在满足缓冲区尺寸和延迟限制的前提下，可按应用需求设计。

在易发生错误的环境中，为了在包丢失情况下仍能正确的恢复时钟，传输包中可以包含包计数值 `packet_count`。此时 `packet_count` 的传送频率和 `transport_rate` 相同。

D.4 `transport_rate` 抖动

如果一个网络或传输流再复用器使从编码器或存储系统到解码器的数据传送延迟不同，则将导致 `transport_rate` 值和实际接收值之间的差异，这称为 `transport_rate` 抖动，例如，若 `transport_rate` 的传输延迟大于同一节目中其余类似字段，则 `transport_rate` 被延迟了。同样地，若延迟小于同一节目中其它参考时钟字段，则 `transport_rate` 提前了。

解码器输入中的时钟抖动在 `transport_rate` 值与其被接收时间组合时反映出来。假定一个时钟恢复结构如图 D2，所有这样的传输率抖动都反映在误差值 e 上， e 值非零使得 f 值变化，并最终导致 27MHz 系统时钟频率的波动。在解码系统中可能允许也可能不允许恢复时钟频率波动，取决于具体应用要求。例如，在产生合成视频输出的精确时钟恢复解码器中，恢复后的时钟频率常被用于生成合成视频样本时钟和色度副载波。副载波频率稳定性的适用规范可能只允许极慢的系统时钟频率调整。在解码器输入端有严重 `transport_rate` 抖动及对 STC 的频率回转率有严格限制的应用中，对合理的解码缓冲区大小和延迟的约束不能保证正常工作。

例如，`transport_rate` 抖动可能是由包括包或单元复用以及包在网络中延迟可变的网络传输造成的，或媒体共享系统中队列延迟或可变的网络存取时间导致的。

传输流或节目流的复用与再复用改变了数据包、`transport_rate` 的顺序或相对位置。`transport_rate` 位置的改变使原先正确的 `transport_rate` 值变为不正确。因为，通常它们的值并没有反映它们经过一定的延迟后被传送的时间。类似地，具有正确 `transport_rate` 值的节目流或传输流可能在一个网络中传输，该网络会对数据流产生可变延迟的影响而未对 `transport_rate` 加以修正。上述结果也造成 `transport_rate` 抖动，同时对解码器设计和性能产生附加影响。网络对 `transport_rate` 所造成抖动的最坏情形往往取决于一些在本部分范围之外的因素，例如各个网络交换中所实现的队列深度以及网络交换总数或对数据流进行的级连再复用操作。

对于传输流而言，再复用操作中需要对 `transport_rate`、`relative_display_time` 以及 `packet_count` 进行纠正以从一个或多个传输流中创建一个新的传输流。这一纠正通过加入一个纠正项来实现。该项计算如下：

$$\text{transport_rate}'(i+1) = \text{transport_rate}(i+1) \times (\text{packet_count}'(i+1) - \text{packet_count}'(i)) / ((\text{packet_count}(i+1) - \text{packet_count}(i)) \times (1 + \text{delay}_{\text{act}}(i+1) - \text{delay}_{\text{act}}(i))),$$

$$\text{relative_display_time}'(i) = \text{relative_display_time}(i) - (\text{delay}_{\text{act}}(i) - \text{delay}_{\text{const}})$$

式中 `packet_count(i)` 和 `packet_count'(i)` 分别是复用前后当前节目的第 i 个包计数， $\text{delay}_{\text{act}}(i)$ 是 `transport_rate(i)` 的实际延迟， $\text{delay}_{\text{const}}$ 是一个用于该节目中所有 `transport_rate` 的常量，其值取决于初始的编码器 / 复用器所使用的策略。例如，该策略可能是尽可能早地安排包，以允许传输链较晚地延迟它们。表 D.1 给出了三种不同的策略以及相应合适的 $\text{delay}_{\text{const}}$ 值。

表 D.1 再复用策略

策略	$\text{delay}_{\text{const}}$
早	$\text{delay}_{\text{min}}$
晚	$\text{delay}_{\text{max}}$
适中	$\text{delay}_{\text{avg}}$

在设计一个系统时，对于编码器 / 复用器应该采用何种策略可能需要专用的协定，因为这将影响到执行任何附加的再复用操作的能力。

在本部分中并未对复用抖动所允许的界限作出规定。然而，在一个好的系统中，5%被认为是一个合适的上限值。

在包括特殊考虑的再复用器的系统中可能需要确保传输流中信息的一致性。这对 PSI 和不连续点尤为重要。PSI 表中的改变可能需要以某种方式被插入到传输流中，从而后续的再复用器不再移动它们直到该信息不再正确。例如，在有些情况下，变动所影响的数据的 PMT 段的新版本在 5ms 中不应该被传送。

类似地，编码器 / 复用器应避免在不连续点周围 $\pm 5\text{ms}$ 窗口内插入 `relative_display_time`。

D.5 网络抖动情形下的时钟恢复

对于接收到的参考时钟时间戳存在明显抖动的应用，解码器的设计有多种选择；解码器如何实现从很大程度上取决于解码输出信号特征和输入信号及抖动特征的需求。

不同应用中的解码器对恢复的系统时钟的精确性和稳定性的要求也不同，而且所要求的稳定性和精确性被认为会沿单个轴下降。该轴的一个端点可以被认为是那些直接利用重建后的时钟来合成用于混合视频中的色度副载波的应用。该需求通常存在于具有精确时钟恢复类型的展示视频场合，如上所述。这样，每个编码图像仅被展现一次，且输出是符合可适用需求的复用视频。在该情况下，色度副载波，像素时钟及帧速率均有精确规定的比率，且都和系统时钟有已定义的关系。合成视频副载波应至少有足够的精确性和稳定性以使任何普通电视接收机的色度副载波 PLL 能够锁定到该副载波，且使用恢复后的副载波来解调的色度信号不显示出明显的人为相位失真。某些应用中需要使用系统时钟来生成与 NTSC、PAL 或 SECAM 制式完全兼容的副载波。这种需求往往比对典型的电视接收机的需求更为迫切。例如，NTSC 的 SMPTE 规范对副载波的精度要求为 3×10^{-6} ，每个水平行时间中的最大短期抖动为 1ns 且每秒最大长期漂移为 0.1Hz。

在不利用恢复后的系统时钟来生成色度副载波的应用中，仍可利用它来生成视频的像素时钟或音频的采样时钟。根据对接收监视器和可接受的音频频率漂移容限的假设，这些时钟有自身的稳定性要求。对于视频图像和音频采样被展现不恰好是一次的应用，即允许图像和音频采样“滑动”，系统时钟可能具有较松的精确性和稳定性的要求。这种解码器可能没有精确的音频和视频展现同步，视频展现也不具有与精确时钟恢复的解码器一样的质量。

对恢复后的系统时钟的精确性及稳定性要求的选择取决于应用。下文着重讲述以上描述的最迫切的需求，即利用系统时钟生成色度副载波的情况。

D.6 用于色度副载波生成的系统时钟

解码器的设计要求由对生成的副载波和必须接受的网络抖动大量的需求所决定。类似地，若已知系统时钟性能要求和解码器设计的性能，就可确定所允许的最大网络抖动。虽然对这些需求的讨论已超出本部分的范畴，但为了清晰地说明问题并阐述一个有代表性的设计，仍标识了一些说明设计所必须的数值。

利用图 D.2 所示的时钟恢复 PLL 电路，所恢复的系统时钟就能满足在频率相对于标准有最大偏移(以 10^{-6} 为单位)，且频率回转率(以 10^{-6} 为单位)最差情况下的要求。点对点的未经过纠正的网络时钟抖动

值是以毫秒为单位的。在图上的 PLL 中网络时钟抖动是以误差项 e 出现的, 且由于 PLL 作为一个对以抖动为输入的低通滤波器来工作, 当输入端有一个 `transport_rate` 时钟的最大增幅函数时, 就会出现对 27MHz 输出的最坏情况。值 e 会有一个等于点到点网络抖动的最大增幅。以 e 的这个最大值作为其输入的低通滤波器(LPF)输出的最大变化率, f , 直接决定了 27 MHz 输出的最大频率回转率。对于任一给定的 e 的最大值和 f 的最大变化率, 均可决定一个 LPF。但是, 随着 LPF 增益或截止频率的减小, PLL 锁定到 `transport_rate` 所表示的频率所需的时间相应延长。利用数字的 LPF 技术或模拟滤波技术可实现长时间恒定的 PLL。在数字 LPF 实现中, 当频率项 f 输入到模拟 VCO 时, f 经 D/A 转换器量化。在计算输出频率的最大回转率时, 应考虑到该转换器的等级。

为确保 e 值收敛到 0, PLL 的开环增益必须很高, 例如在该 PLL 的低通滤波器中采用积分函数。在精度给定时, 可以构造一个初始操作频率符合该需求的 PLL。此时, 在 PLL 锁定前, 初始的 27 MHz 频率的精确性已足够满足频率需求。如果不是因为解码器的缓冲区最终会上溢或下溢的话, 这一初始频率对长时间的操作是足够的。但是从解码器开始接收并解码数据到系统时钟被锁定到以接收的 `transport_rate` 所表示的时间和时钟频率的这段时间内, 数据到达缓冲区的速率和被抽取的不同, 或者说, 解码器在与 STD 不同的时间抽取访问单元。伴随着所恢复的系统时钟频率与解码器时钟频率的不同, 解码器的缓冲区占用度将与 STD 中的不同。解码缓冲区的占用度将随初始的 VCO 频率及编码器系统时钟频率增加或减少。假定该关系未知, 则解码器需要附加的缓冲区以考虑每种情况。解码器应被构造为能够将所有的解码操作延迟一段时间, 该时间至少等于在初始 VCO 频率大于编码器时钟频率时为避免缓冲区溢出所分配的附加缓冲区所代表的时间。如果初始 VCO 频率不够精确以满足所述的精度要求, 则 PLL 必须在解码可以开始前到达锁定状态, 且存在关于在这段时间内 PLL 性能及附加缓冲区数量和合适的静态延迟量的一个不同的集合。

在图 D.2 的 PLL 的误差项 e 中产生一个阶跃函数的输入时钟抖动中的阶跃信号将产生一个输出频率项 f 。这样, 当它乘以 VCO 增益值后, 最大变化率将小于规定的频率回转率。VCO 的增益用输出频率相对与控制输入的改变而发生的改变来表示。对 PLL 中的 LPF 有一个附加限制, 即当循环锁定时, e 的稳定值必须是有界的, 这样才能保证必须实现的附加缓冲区和静态解码延迟量的有界性。当 LPF 有很高的 DC 增益时, 该项值最小。

和图 D.2 略有差别的时钟恢复电路也是可行的。例如, 可用数控晶振(NCO)代替 VCO 来实现一个控制循环, 该 NCO 使用一个固定频率的晶振且在输出端从规则的周期性的施加事件中插入或删除时钟周期以调整解码和展现时钟。当在合成视频中使用这种方法时可能有一些困难, 因为可能会导致副载波错误的相位偏移或水平或垂直扫描时钟中的抖动。一个可能的方法是在垂直消隐开始时调整水平扫描周期, 同时维持色度副载波的相位。

总之, 能否构造一个可以重构有足够的精确性和稳定性的解码器, 同时又能维护所需的解码器缓冲区大小及增加解码延迟, 这取决于该需求所规定的值。

D.7 分量视频与音频的重构

如果在解码器输出端产生一个分量视频, 则时钟精确性和稳定性要求通常不象合成视频那么严格, 典型的频率容限应限制在显示偏转电路能接受的范围之内, 稳定性容限则由在显示器上避免出现可察觉的图象移动要求所决定。

以上原理总是适用的, 尽管特定的需求通常较容易满足。

音频采样率重建也遵循以上原理。然而, 稳定性需求由可接受的长期和短期采样率的变化量所决定。使用上一章所述的 PLL 方法可以使短期偏差很小且长期频率变化表现为感知的音调。同样, 一旦对这些变化设置了特定的界限, 就能决定设计要求。

D.8 帧滑动

在某些对解码时钟不作精确要求的应用中, 解码器系统时钟不一定调节其工作频率以匹配接收的 `transport_rate` 所代表的频率; 它可能有一个自运行的 27 MHz 的时钟。但解码器 STC 仍须服从所接收

到的数据。在此场合下，STC 值必须更新以匹配收到的 `transport_rate`，这会导致 STC 值的不连续。这种不连续的幅值取决于解码器的 27 MHz 频率与编码器的 27 MHz（即收到的 `transport_rate` 所代表的值）之间的差及收到相邻 `transport_rate` 的时间间隔。因为解码器的 27 MHz 系统时钟频率没有锁定到接收数据，故在假定每个视频和音频展现单元恰好展现一次且保持解码器和编码器中的图像和音频展现率一致的精确时钟时，不能用于创建视频或音频采样时钟。使用这种结构的解码和展现系统有多种可能实现方式。

在一种类型的实现中，图像和音频样本在解码器 STC 指定的时间被解码，而在稍有差别的时间被展现，这取决于本地产生的样本时钟。图像和音频样本可能偶尔展现多于一次或不被展现，这依赖于解码器采样时钟和编码器系统时钟的关系。对音频而言，这被称为“帧滑动”或“样本滑动”。该机制将会引入可察觉的人为失真，音视频常常因为在某些时间单元上重复或删除图像和音频展现单元而不严格同步。在特定实现方式中，编码数据或解码的展现数据常常需要附加缓冲区。解码与展现可能紧密衔接，而不遵循解码器 STC；解码的展现单元也可能由于延迟和重复展现而被存储。若解码与展现时间相同，则需要一种机制以支持删除图像和音频采样的展现而不会使预测编码数据的解码出现问题。

D.9 网络抖动的平滑处理

在某些应用中，可以在网络和解码器之间引入某种机制以减弱网络引起的抖动程度。这种机制是否可行，决定于接收流的种类及抖动幅值和类型的期望值。

传输流和节目流在它们的语法中均指出了该流期望输入到解码器的速率。这一速率并不精确且不能用于精确地重建数据流时钟。但它们可用作平滑机制的一部分。

例如从网络接收的一个传输流，其数据可能以突发方式传送。有可能缓冲所接收到的数据并以近似恒定速率将数据从缓冲区传输到解码器，以保证缓冲区近似半充满。

然而一个变速流不能以恒定的速率传输，且对于变速流而言，平滑缓冲区不会总是半充满的。缓冲区的恒定平均延迟要求缓冲区占用度随数据率变化。从缓冲区中抽取数据并输入到解码器的速率，可近似地使用数据流中的速率信息。在传输流中，期望的速率由各 `transport_rate` 字段值及彼此间的传输流字节数决定。在节目流中，则由 `program_max_rate` 明确指定，尽管在本部分中，该速率可能在 SCR 处下降为 0，即当数据以指定速率传送时，`transport_rate` 在预计的时间之前到达。

在变速流的场合下，平滑缓冲区的占用度随时间变化，不一定能由速率信息完全确定。在另一个可行的途径中，可以用 `transport_rate` 测出数据进入缓冲区的时间，并控制数据离开缓冲区的时间。可以设计一个控制环来提供恒定的缓冲区平均延迟。可以看到，这种设计与图 D.2 的控制环是类似的。在解码器之前插入一个平滑机制所获得的性能也可以通过多个时钟恢复 PLL 来达到。通过级联 PLL 的组合低通滤波器可以达到消除所接收到的时钟的抖动。

附 录 E

(资料性附录)
数据传输应用

E.1 一般性考虑

GB/T 20090.1 传输复用在传输音频和视频之外还用于传输其它数据。
不象广播应用中的音/视频流那样，数据基本流可能是不连续的。
尽管标识一个 PES 分组包的开始已经是可能的，但未必总能通过下一个 PES 分组包的开始来标识前一个 PES 分组包的结束。这是由于携带 PES 分组包的一个或多个传输流包可能会丢失。

E.2 建议

一个合适的解决方案是在一个相关的 PES 分组包之后紧接着传输下一个 PES 分组包。当没有更多的 PES 分组包要发送时，可以发送一个不带有有效负载数据的 PES 分组包。
表 E.1 是该 PES 分组包的一个例子。

表 E.1 PES 分组包头示例

PES分组包头字段	值
packet_start_code_prefix	0x000001
stream_id	已指定
PES_packet_length	0x0003
'10'	10
PES_scrambling_control	00
PES_priority	0
data_alignment_indicator	0
Copyright	0
original_or_copy	0
PTS_DTS_flags	00
ESCR_flag	0
ES_rate_flag	0
DSM_trick_mode_flag	0
Additional_copy_info_flag	0
PES_CRC_flag	0
PES_extension_flag	0
PES_header_data_length	0x00

附录 F

(资料性附录)

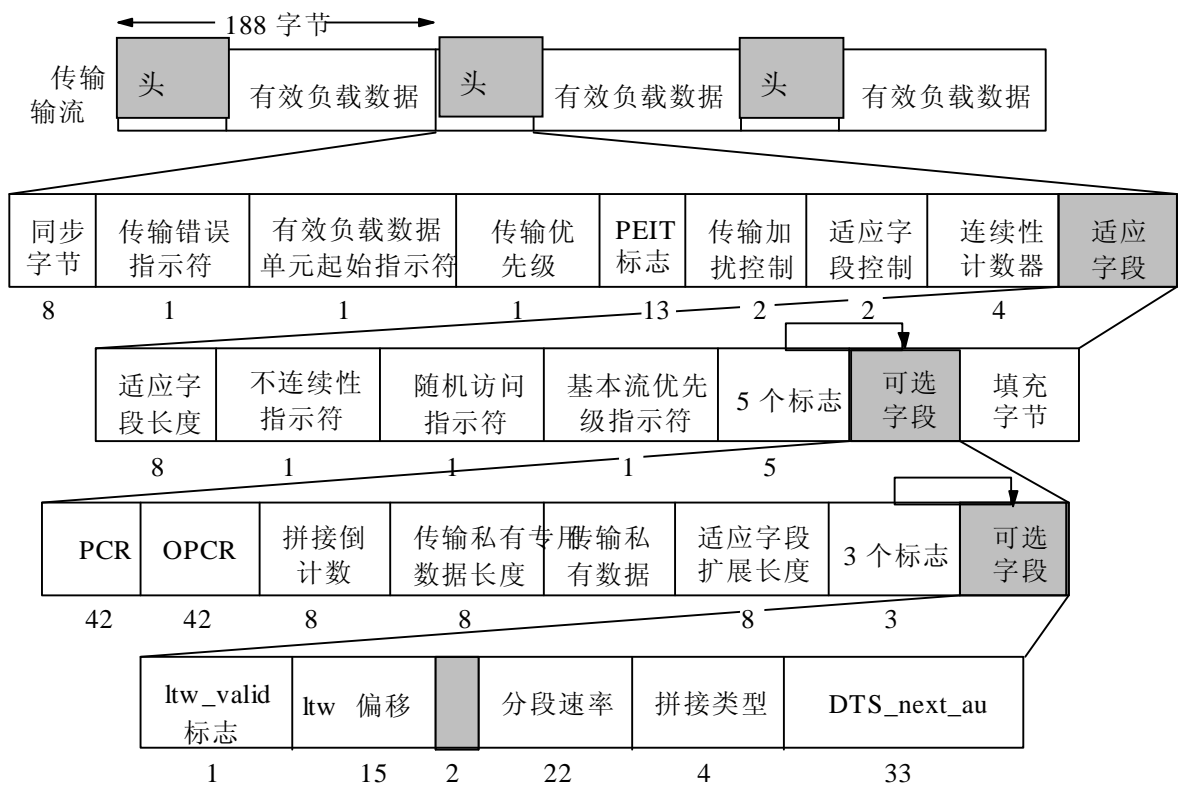
本部分语法的图形表示

引言

本附录给出了传输流和节目流语法图示信息，但它并不能取代本部分的任何章、条。
 为了图示清楚，图中未完整地描述或表示所有字段。保留字段可能被省略或仅由一个区域指出而无任何细节。字段长度以位为单位给出。

F.1 传输流语法

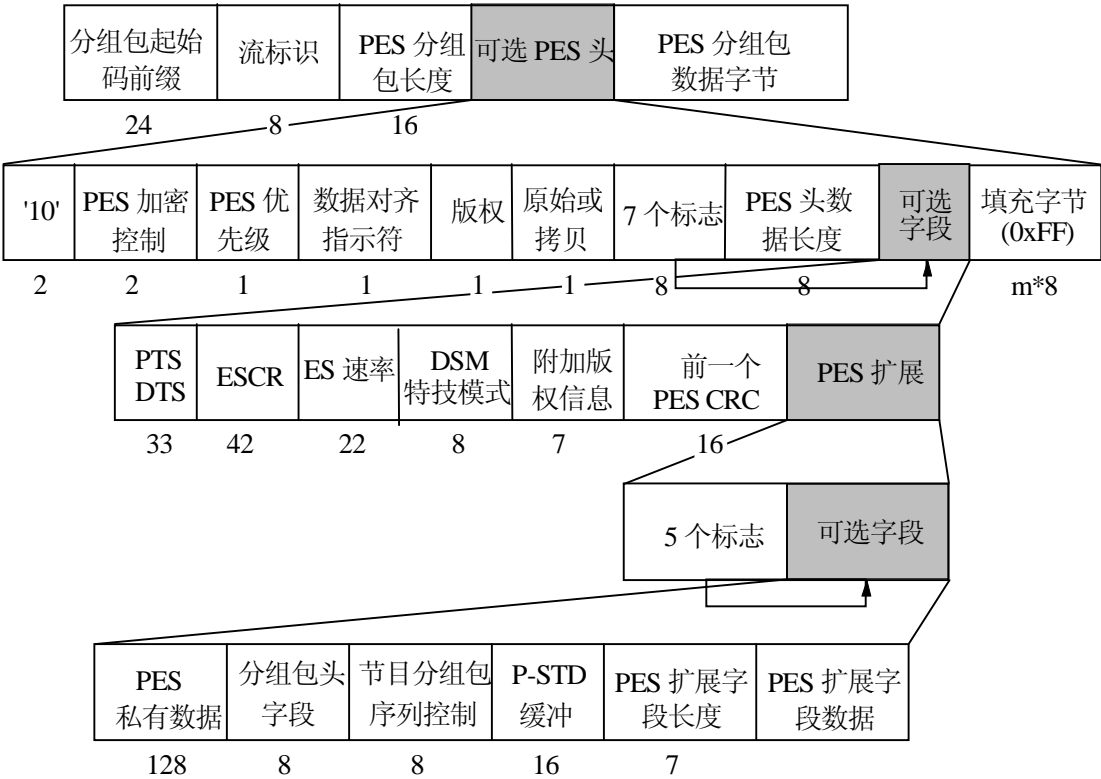
传输流语法见图 F.1。



图F.1 传输流语法图

F.2 PES分组包

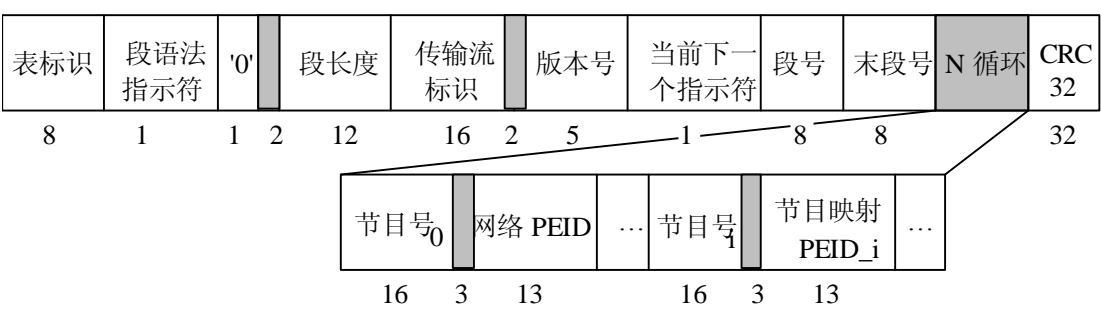
PES 分组包见图 F.2。



图F.2 PES分组包语法图

F.3 节目相关段

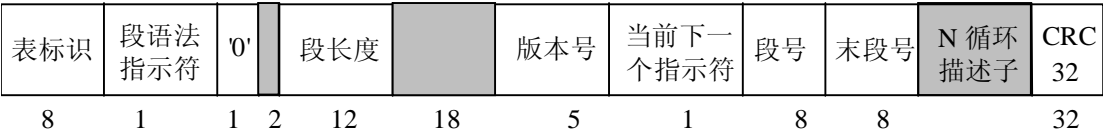
节目相关段见图 F.3。



图F.3 节目相关段图

F.4 CA段

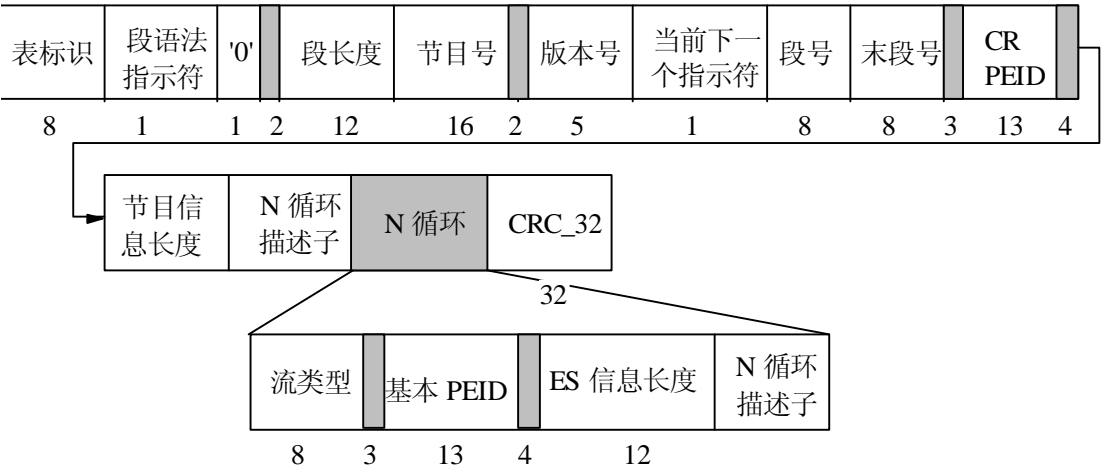
CA 段见图 F.4。



图F.4 条件访问段图

F.5 传输流节目映射段

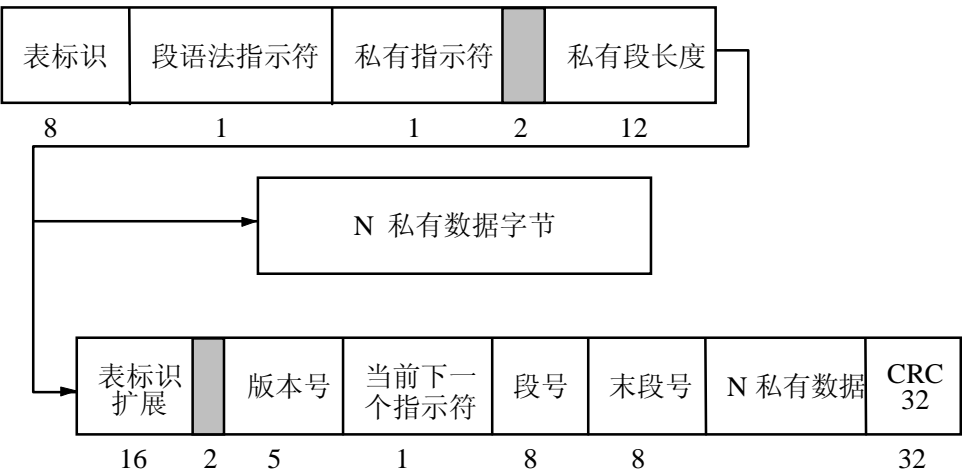
传输流节目映射段见图 F.5。



图F.5 传输流节目映射段图

F.6 私有段

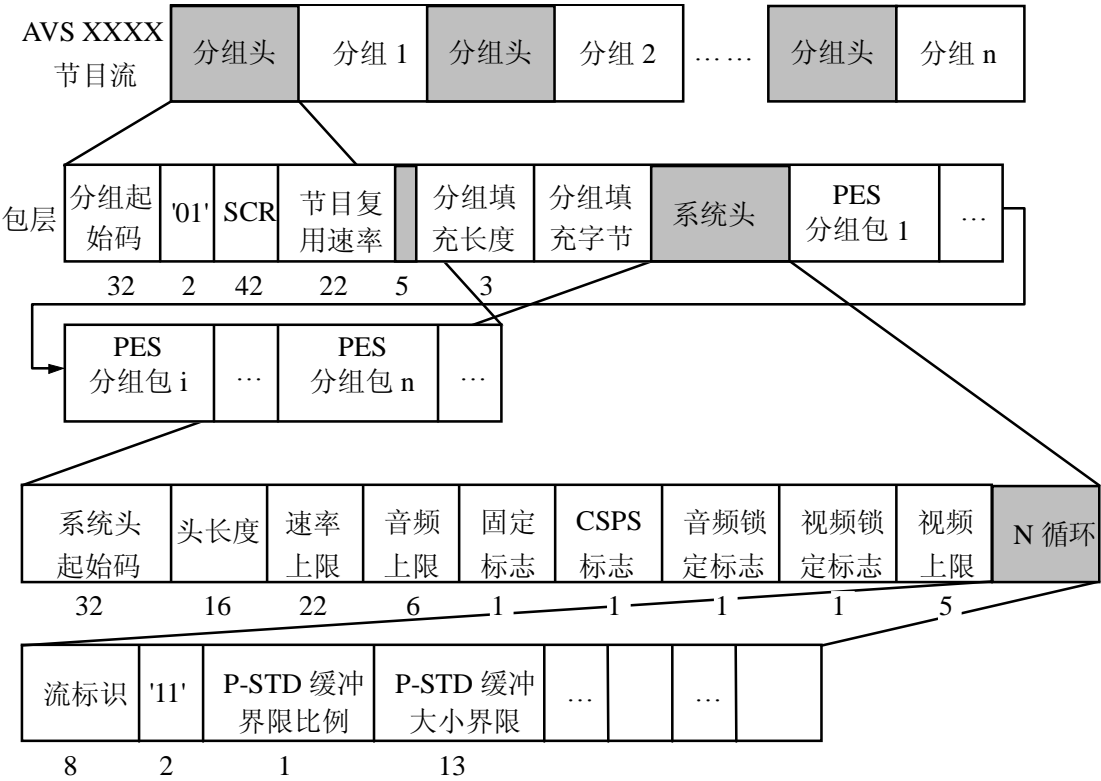
私有段见图 F.6。



图F.6 私有段图

F.7 节目流

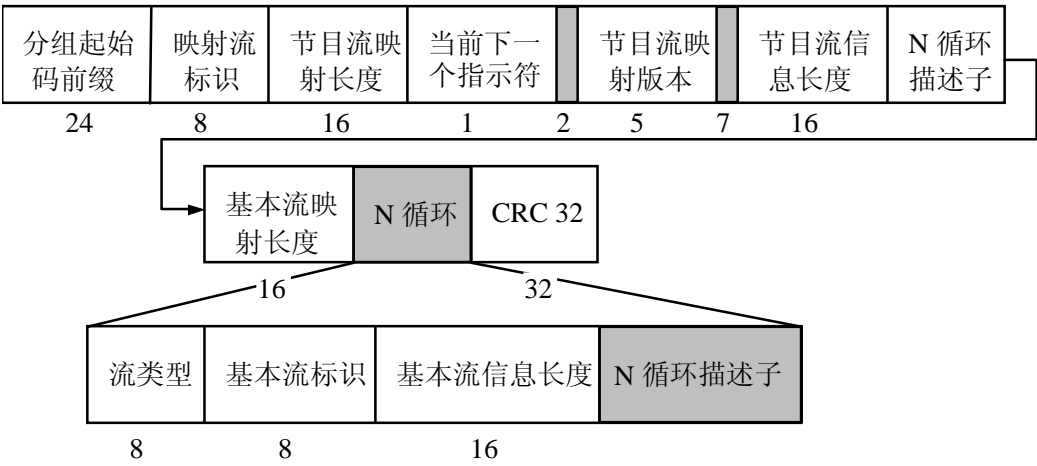
参见图 F.7。



图F.7 节目流图

F.8 节目流映射

节目流映射见图 F.8。



图F.8 节目流映射图

附录 G

(资料性附录)

通用信息

G.1 同步字节冲突

建议在 PEID 值的选择中应避免同步字节的周期性冲突。这一冲突可能会潜在地出现在 PEID 字段或 PEID 字段与相邻标志设置的组合中。建议允许在最多 4 个连续传输流包头的同一个位置上出现同步字节冲突。

G.2 跳过的图像状态和解码处理

假定正在被显示的序列中只包含 I 和 P 帧。用 `picture_next` 表示下一帧待解码的图像，而正在被播放的图像用 `picture_current` 来表示。因为视频解码器可能跳过图像这一事实，当移出那些位以用于瞬时解码和显示的时间到达时，`picture_next` 的所有比特不一定都在 STD 缓冲区 EB_n 或 B_n 中。当此情况发生时，不从缓冲区中移出任何位，并再次展现 `picture_current`。当下一个图像的显示时间到达时，若此时与 `picture_next` 相对应的剩余位在缓冲区 EB_n 或 B_n 中，`picture_next` 的所有位被移走且 `picture_next` 被显示。若此时与 `picture_next` 相对应的所有位不在缓冲区 EB_n 或 B_n 中，则重复上述再次显示 `picture_current` 的过程。该过程一直重复到 `picture_next` 能被显示为止。注意到在比特流中，若 `picture_next` 前有一个 PTS，则 PTS 可能是不正确的，相差图像显示间隔的若干倍，该差值本身取决于一些参数，且 PTS 必须被忽略。

当上述跳过的图像情况发生时，要求编码器在 `picture_next` 后解码的图像前插入一个 PTS。这使得解码器可以立即确认它已正确展现了所收到的图像序列。

G.3 PEID值的选择

应用应尽可能地使用小的 PEID 值(避免表 22 中规定的保留值)并将它们编组在一起。

G.4 PES start_code冲突

三个具有 `packet_start_code_prefix` 值(0x000001)的连续字节，在和第 4 个字节拼接在一起时，可能会在流中的一个意想不到的位置与四字节的 `PES_packet_header` 相冲突。

这种所谓的起始码冲突在视频基本流中不会出现。在音频和数据流中可能出现。这种情况也可能在 `PES_packet_header` 和 `PES_packet` 有效负载的边界处出现。即使 `PES_packet` 有效负载数据是视频。

附 录 H

(资料性附录)

私有数据

私有数据指国际标准未规定的，本部分中涉及的任何用户数据。本部分并未(今后也不会)对这些数据的内容进行规定。本部分中定义的 STD 未涉及私有数据，除了分离操作过程之外。一个专门的团体可以为各个私有流定义 STD。

在本部分语法中，私有数据可携带于下列位置。

传输流包（见表 2）

transport_packet()语法的数据字节可能包含私有数据。以该格式携带的私有数据在 stream_type 表 30 中被称为用户专用。包含私有数据的传输流包也可包括 adaptation_field()。

传输流适应字段（见表 5）

adaptation_field() 中出现的任何可选 private_data_bytes 由 transport_private_data_flag 标记。private_data_bytes 的数量受 adaptation_field_length 字段语义的限制，这里的 adaptation_field_length 的值不能超过 183 字节。

PES 分组包（见表 16）

有两种方法在 PES 分组包中携带私有数据。一种是在 PES_packet_header 中，在 PES_private_data 可选的 16 个字节中。该字段的存在由 PES_private_data_flag 标识，PES_private_data_flag 字段的存在由 PES_extension_flag 标识。如果出现这些字节时，则它们与相邻字段一起考虑时，不应和 packet_start_code_prefix 发生冲突。

第二种方法是在 PES_packet_data_byte 字段中。在 stream_type 表 30 中将其称为 PES 分组包中的私有数据。这种私有数据可被一分为二：private_stream_1 表示遵循 PES_packet()语法的 PES 分组包中的私有数据，这样，直到且包括(但并不局限于)PES_header_data_length 的所有字段均会出现。private_stream_2 表示包含私有数据的 PES_packet_data_byte 之后仅跟有前三个字段的 PES 分组包中的私有数据。

注意，PES 分组包同时存在于节目流和传输流中，因此 private_stream_1 和 private_stream_2 也同时存在于节目流和传输流中。

描述子

描述子存在于节目流和传输流中。用户可定义一系列的私有描述子。这些描述子应以 descriptor_tag 和 descriptor_length 字段开始。对于私有描述子，descriptor_tag 取值为 64~255，如表 40 所示。这些描述子可放在 program_stream_map()表 36、CA_section()表 28、TS_program_map_section()表 29 及任意私有段表 31 中。

私有段

私有段表 30 进一步以两种形式提供了携带私有数据的方法。这种类型的基本流可以在 stream_type 表 30 中标识为 PSI 段的 private_data。一种类型的 private_section()只包含前五个已定义字段，后跟私有数据。在这种结构中，section_syntax_indicator 应置'0'。在另一中类型中，section_syntax_indicator 应置'1'，直到且包括 last_section_number 的所有语法均须出现，后跟 private_data_bytes，且以 CRC_32 校验码结尾。

附 录 I

(资料性附录)

系统一致性和实时接口

GB/T 20090.1 节目流和传输流的一致性由本部分所规定。这些规范除其它要求外还包括一个系统目标解码器(T-STD 及 P-STD)以规定当流输入到理想解码器时该解码器的行为。这一模型及有关验证,除传输流和节目流所表示的系统时钟频率的精确性之外,不包括涉及流实时传输的信息。所有传输流和节目流必须符合本部分。

此外,对于传输流和节目流输入到解码器,还有一个实时接口规范。本部分允许对 MPEG 解码器和网络适配器、信道、存储介质间的接口加以标准化。信道的时间效应,以及实际适配器无法完全消除该效应,导致了实际传输和理想传输之间存在偏差。虽然不是所有 MPEG 解码器都要实现该接口,但包括该接口的实现必须遵循本部分。本部分包括了传输流和节目流到解码器的实时传输行为,以保证解码器中的编码数据缓冲区不发生上溢和下溢,以及解码器能以应用所要求的性能来恢复时钟。

MPEG 实时接口规定了理想字节传送时间表允许的最大偏差,该值由编码在流中的传输率(transport_rate)来指出。

附 录 J

(资料性附录)

引入抖动的网络与 AVS 解码器的接口

J.1 引言

在本附录中系统流这个词既用于表示 GB/T 20090.1 传输流也用于表示 GB/T 20090.1 节目流。在使用术语 STD 时,对节目流专指 P-STD(节目系统目标解码器),对传输流专指 T-STD(传输系统目标解码器)。通过对流的分析可得到系统流预期的字节传输时间表。如果一个系统流能被理想解码器的一个数学模型 STD 解码,则称它为规范的。如果一个规范的系统流在一个引起抖动的网络上传输,则实际的字节传送时间表可能和预期的时间表有很大不同。此时可能无法在理想解码器中对系统流进行解码,因为抖动会引起缓冲区上溢或下溢,且很难恢复时间基准。ATM 就是引起抖动的网络的一个重要例子。本附录的目的是对在引起抖动的网络上传输系统流的实体给出指导和考察。可以为多种类型的网络,包括 ATM,开发用于传输系统流的特定网络规范性模型。STD 加上一个实时接口定义在该模型的定义中居主导地位。开发网络规范性模型的一个框架见第 J.2 章。

第 J.3 章讨论了网络编码的三个例子,网络编码使构造平滑抖动的网络适配器成为可能。在第一个例子中,假定系统比特流速率恒定,并用 FIFO 来平滑抖动。例二中,网络适配层中包含了时间戳以方便平滑抖动。例三中,假定有一个端到端的公共网络时钟,并用它来平滑抖动。

第 J.4 章给出可以适应网络引起的抖动的解码器的两个实现例子。在例一中,在网络输出与 AVS 解码器之间插入了一个平滑抖动的网络适配器。假定 AVS 解码器符合实时 AVS 接口标准。这一接口要求 AVS 解码器比 STD 的理想解码器有更高的抖动容限。网络适配器处理输入的抖动比特流并输出字节传输时间表符合实时要求的系统流。例一在 J4.1 中讨论。对某些应用而言,采用网络适配器的代价太大,因为它需要两个阶段的处理。因此,在例二中,去除抖动和 AVS 解码的功能集成在一起,忽略了去除抖动设备的中间处理过程,故只须一个阶段的时钟恢复。在本附录中将这种把去除抖动和解码综合在一起的解码器称为特定网络集成解码器,简称集成解码器。在 J4.2 中对集成解码器加以讨论。

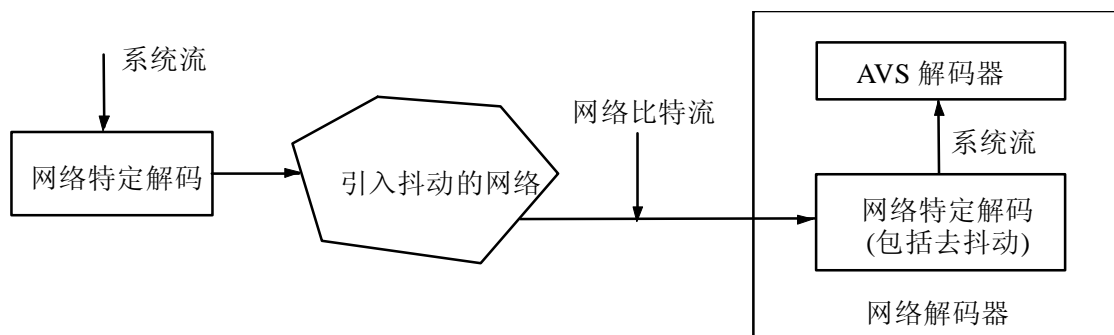
为了构造网络适配器或集成解码器,需要对点对点网络抖动的峰值作出假设。为提高互操作性,对每种相关的网络类型均须规定抖动的界限值。

J.2 网络一致模型

图 J.1 给出了在引入抖动的网络中传输系统流的一个模型。

系统流在进入特定网络编码装置后,被转变为特定网络格式的系统流。该格式中包含部分有助于在网络输出消除抖动的信息。网络解码器包含一个特定网络解码器和一个 GB/T 20090.1 解码器。假定 GB/T 20090.1 解码器符合实时接口规范,与 STD 有相同的结构且有适当扩充的缓冲区以提供更高的抖动容限。特定网络解码器移去由特定网络编码器加入的非 GB/T 20090.1 数据,并对网络输出进行去抖动。特定网络解码器的输出是符合实时规范的系统流。

一个网络目标解码器(NTD)可按以上构造定义。一个规范的网络比特流应能被 NTD 解码。一个网络解码器如果能对任何可被 NTD 解码的网络流解码,则称之为规范的。一个实际网络解码器在结构上不一定和 NTD 相同。

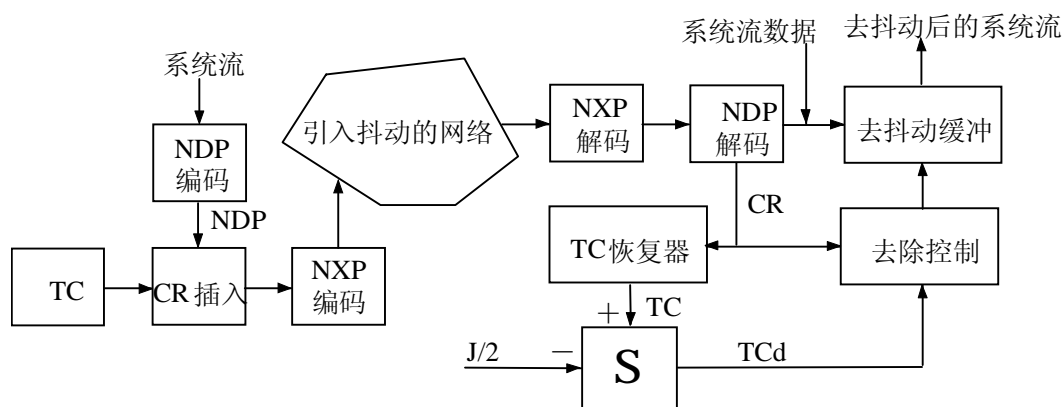


图J.1 在引起抖动的网络上传送系统流

J.3 平滑抖动的网络规范

对比特率恒定的系统流而言，平滑抖动可通过 FIFO 实现。在网络适配层中不需要附加对去抖动提供特定支持的数据。移去网络编码所加入的字节后，系统流被置入 FIFO。一个 PLL 根据缓冲区占用度来调节输出速率以保持缓冲区为近似半充满状态。本例中，平滑抖动的效果取决于 FIFO 的大小和 PLL 的性能。

图 J.2 给出了实现平滑抖动的另一方法。在该例子中，假定网络适配层提供时间戳支持。使用这种技术对恒定或可变速率的系统流均可去抖动。



图J.2 用网络层时间戳来平滑抖动

假定网络适配器被设计为补偿 J 秒的峰-峰抖动。通过使用对时钟(TC)的参考时钟(CR)采样来重建预期的字节传送时间表。网络数据分组(NDP)编码把每个系统流分组转变为一个网络数据分组(NDP)。NDP 中包含一个传送 CR 值的字段，在 NDP 离开 NDP 编码器时 TC 的当前值被插入到该字段。网络传输分组(NXP)功能是把 NDP 封装成网络传输分组。通过网络传输之后，当 NDP 进入 NDP 解码器时，CR 被 NDP 解码器提取出来。CR 用于重建 TC，例如通过使用 PLL。当被延迟的 TC(TCd)等于第一个 AVS 分组的 CR 时，第一个 AVS 分组从去抖动缓冲区中移走。当后续 AVS 分组的值等于 TCd 的值时也被移走。

忽略掉诸如 TC 时钟恢复循环的速度以及 TC 的纯正性之类的实现细节，去抖动缓冲区的大小仅取决于所需平滑的最大的峰-峰抖动及系统流中的最大传输率。去抖动缓冲区的大小， B_{dj} 由下式决定：

$$B_{dj} = JR_{\max}$$

其中， R_{\max} 是系统流的最大数据率，以比特/秒为单位。在分组穿过网络遇到标准延迟的情况下，缓冲区为半满。当延迟为 $J/2$ 秒时，缓冲区是空的，当延迟(提前)为 $-J/2$ 秒时，缓冲区为满。

作为最后一个例子，在有些场合下，有一个端到端的公共网络时钟，且把系统时钟频率锁定到公共时钟是可行的。网络适配器能通过 FIFO 来去抖动。适配器用 `transport_rate` 重建原始的字节传送时间表。

J.4 解码器实现举例

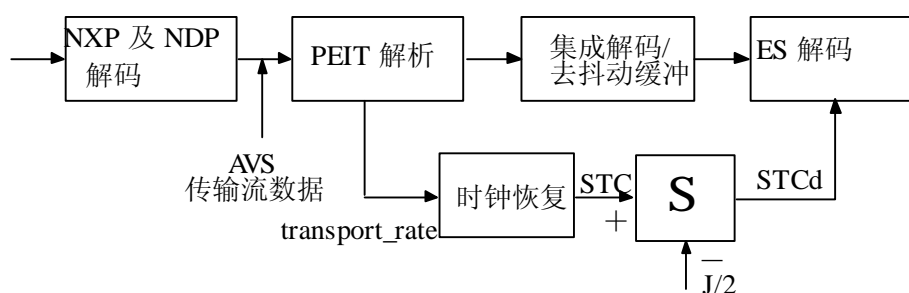
J.4.1 跟有AVS解码器的网络适配器

在实现中，一个符合规范的网络适配器被连接到一个符合实时接口规范的 AVS 解码器。

J.4.2 集成解码器

J4.1 中的例子需要两个阶段的处理过程。前一阶段用于对网络输出去抖动。后一阶段通过处理 `transport_rate` 来恢复 STC，以满足 STD 解码的需要。本条的例子是一个在单个系统中集成了去抖动和解码两个功能的解码器。通过使用去抖动后的 `transport_rate` 值直接恢复 STC 时钟。为了表述该示例，假定使用传输流。

图 J.3 描述了集成解码器的操作。假定输入到解码器的网络分组流与图 J.2 中的一样。



图J.3 集成去抖动和AVS解码

输入的网络分组被 NXP 和 NDP 解码功能重组为 AVS 传输流数据。抖动的 GB/T 20090.1 传输流包被滤波以抽取具有有效 PEID 的分组。在本例中，被解码的 PEID 也携带有 `transport_rate`。`transport_rate` 值被送到 PLL 以恢复 STC。所选 PEID 的整个分组被放入到集成缓冲区。从 STC 中减去 $J/2$ 以获得延迟的 STC、STCd。同样， J 是网络解码器能容纳的峰-峰抖动。引入延迟是为了保证访问单元需要的所有数据在该访问单元的 $td_n(j)$ 等于 STCd 的当前值时，都已到达了缓冲区。

不考虑诸如 STC 时钟恢复循环的速度以及 STC 的纯正性之类的实现细节，有：

$$\begin{aligned} B_{\text{size}} &= B_{\text{dec}} + B_{\text{mux}} + B_{\text{OH}} + 512 + B_J \\ &= B_n + 512 + B_J \end{aligned}$$

其中： $B_J = R_{\text{max}} J$ ， R_{max} 指数据输入到 PEIT 解析器中的最大速率。和传输 STD 中一样，集成的存储器可被分为两部分，这取决于具体的实现。

附录 K

(资料性附录)

拼接传输流

K.1 引言

在本附录中，术语“拼接”指的是对两个不同基本流的传输层进行的连结操作，产生的传输流完全符合 GB/T 20090.1。这两个基本流可能创建于不同的地点或时间，且在创建时也未必就打算要拼接在一起。下文中，我们称其中“旧”的流为连续基本流(音频或视频)，它在某一点开始被另一个流(“新”的流)取代，该点称为拼接点。它是“新”“旧”两个流的数据之间的边界。

拼接可以是无缝的或非无缝的：

一个无缝的拼接不会导致解码的不连续性(见 9.6)。也就是说，“新”流的第一个访问单元的解码时间与拼接前“旧”流的访问单元的解码时间一致，即若“旧”流继续的话，它等于“旧”流的下一个访问单元的解码时间。以下，这一解码时间被称为“无缝解码时间”。

一个非无缝拼接将导致解码的不连续性，也就是说，“新”流的第一个访问单元的解码时间大于无缝解码时间。

注：不允许解码时间小于无缝解码时间。

只要所产生的流是合法的，拼接可在任何传输流包边界处进行。但在一般情况下，如果不知道 PES 分组包和访问单元的起始位置，这一限制将导致传输层和 PES 层及基本流层均被分析，在有些情况下，甚至需要对传输流包的有效负载数据作一些处理。要想避免这些复杂的操作，拼接就应在传输流具有良好特性的部位进行，这些特性由拼接点的出现所指出。

拼接的存在由 `splice_flag` 和 `splice_countdown` 字段指示(这些字段的语义见 6.3.4)。在下文中，`splice_countdown` 字段值为 0 的传输流包称为“拼接包”。拼接点紧跟在拼接包的末字节之后。

K.2 拼接点的不同类型。

一个拼接点可以是普通的或无缝的。

K.2.1 普通拼接点

如果没有 `seamless_splice_flag` 字段或其值为 0，则该拼接点为普通的。一个普通拼接点仅标志基本流的对齐特性：拼接包在访问单元的末字节处结束，具有相同 PEID 的下一个传输流包的有效负载以一个 PES 分组包头开始，该 PES 分组包的有效负载数据以基本流存取点开始(对视频而言，也可以紧跟在一个基本流存取点之后的 `sequence_end_code()` 为开始)。这一特性允许对传输层方便地进行“剪贴”操作，同时遵守句法限制并确保流的一致性。但它并不提供有关时间信息和缓冲区特性的任何信息。这样，对这类拼接点，无缝拼接只能在专用安排的帮助下或通过分析传输流包的有效负载数据并跟踪缓冲区状态及时间戳的值才能实现。

K.2.2 无缝拼接点

若 `seamless_splice_flag` 字段存在并置'1'，则该拼接点给出了“旧”流的一些特性的有关信息。这一信息不是针对解码器的。它的目的在于方便无缝拼接。这一拼接点称为无缝拼接点。有用的信息包括：无缝解码时间，即 DTS，该值被编码在 `DTS_next_AU` 字段内的 DTS 值。该 DTS 值以拼接分组中有效的时间基准来表示。

对视频基本流，为便于无缝拼接，在“旧”流产生时加上一些限制。这些条件在与视频流的档次和等级相对应的表中由 `splice_type` 字段给出。

注意，通过丢弃附加信息，一个无缝拼接点就可被用作普通拼接点。如果认为这一信息有助于进行无缝拼接或除拼接以外的其它目的，则也可使用该信息。

K.3 拼接中解码器的行为

K.3.1 非无缝拼接

如上所述，一个非无缝拼接将导致解码不连续。

值得注意的是，对于这种拼接，必须满足与解码不连续性(见 9.6)相关的限制。特别地：

在“新”流的首访问单元中，应对 PTS 编码(除非在特技模式操作中或 `low_delay='1'`时)；

从该 PTS(或相关的 DTS)导出的解码时间不应早于无缝解码时间；

对视频基本流，若拼接包不以 `sequence_end_code()` 结束，则“新”流应以其后紧跟 `sequence_header()` 的 `sequence_end_code()` 为开始。

理论上，因为它们引入了解码的不连续性，这些拼接将导致展现单元展现的不连续(即两个连续图像或音频帧之间的一段可变长度的静止时间)。实际上，结果取决于解码器的实现方式，特别是视频。对某些视频解码器，冻结若干图像可能是一种较受欢迎的解决方法。参见 ISO / IEC 13818-4。

K.3.2 无缝拼接

解码连续性是为了保证播放连续性。对音频而言，它总是能得到保证。但要注意的是，对视频而言，播放的连续性在以下两种情况时理论上是不可能的：

a) “旧”流以一个低延迟序列的末尾为结束，而“新”流以一个非低延迟序列的起始为开始。

b) “新”流以一个低延迟序列的末尾为结束，“新”流以一个非低延迟序列的起始为开始。

以上情形造成的影响取决于实现方法。例如，对情形 a) 一帧图像可能持续显示两个帧周期。对情形 b) 可能要跳过一帧图像。然而，在技术上无副作用地实现对以上情形的支持是可能的。

此外，参见 GB/T 20090.2-2006 的 7.2.2，`sequence_end_code()` 应在“新”流的第一个 `sequence_header()` 之前出现，如果在两个流中至少有一个参数(即定义在序列头或序列头扩展中的参数)的值不同。唯一的例外是那些定义在量化矩阵中的参数。例如，如果“旧”流和“新”流的比特率字段值相异，则 `sequence_end_code()` 就应出现。于是，如果拼接包不以 `sequence_end_code` 结束，拼接流就应以一个跟有 `sequence_header` 的 `sequence_end_code` 为开始。

根据以上各段，在大多数拼接中，`sequence_end_code` 字段是强制的，即使在无缝拼接中。必须注意到 GB/T 20090.2-2006 规定了视频序列(即 `sequence_header` 和 `sequence_end_code` 之间的数据)解码过程，而未规定如何处理一个序列变化。因此，对于该情况下解码器的行为参见 ISO / IEC 13818-4。

K2.3 缓冲区溢出

即使两个基本流在拼接前均符合 T-STD 模型，也不能保证在这两个流的比特均位于这些缓冲区中的时间段内，STD 缓冲区不发生拼接流溢出。

对恒定速率视频，如果对“旧”流没有特殊条件，在拼接中也未采取特殊预防措施，则在“新”流视频比特率大于“旧”流时可能溢出。实际上，若数据以“旧”流速率被传给 T-STD 则 MB_n 和 EB_n 一定不会溢出。但如果在“旧”流被完全移出 T-STD 之前， TB_n 输入端的移入速率加快，STD 缓冲区的占用度将会比“旧”流继续而不进行拼接时提高，且可能导致 EB_n 或 MB_n 溢出。对可变速率比特流，若“新”流的传送速率高于创建“旧”流时所作的限制，也存在类似问题。这种情形是不允许的。然而，生成“旧”流的编码器可以在拼接点附近添加 BBV 缓冲区管理的条件以限制“新”视频流的比特率小于某个选定值。例如，对无缝拼接点，这些对 `splice_decoding_delay` 和 `max_splice_rate` 的附加条件可由在表 6 至表 15 中与之相对应的 `splice_type` 值指出。此时若“新”流比特率小于 `max_splice_rate`，则可确保在两个流的比特均位于 T-STD 缓冲区期间拼接流不会导致溢出。

在没有以上限制的场合，可通过在“旧”流和“新”流之间的比特流传输中引入一个静止时间来避免这一问题，使 T-STD 缓冲区在传送“新”流的之前足够空。设 t_{in} 为“旧”流的最后一个访问单元的末字节进入 STD 的时间， t_{out} 为它离开 STD 的时间，可以确保在 $[t_{in}, t_{out}]$ 时间间隔内进入 T-TD 的拼接流的比特数和“旧”流继续而不拼接时一样。例如，若“旧”流有恒定比特率 R_{old} ，“新”有恒定比特率 R_{new} ，则引入满足下式的静止时间 T_d 足以避免溢出：

$$T_d \geq 0 \text{ 且 } T_d \geq (t_{out} - t_{in}) \times (1 - R_{old} / R_{new})$$

附 录 L

(资料性附录) 系统解复用过程

L.1 基本模式解复用

L.1.1 节目入口搜寻

本附录考虑与 MPEG2 系统兼容的情况。PEIT 表内结构中的条目可以包含如 PAT, PMT, CAT, NIT, TSDT 等 PSI 信息,也可以包含其它各种节目成分的信息。每个条目对应唯一的 PEID 字段。所以通过搜索已知的 PEID 字段,相应的条目就可以在 PEIT 表中得到。具体的 PEID 定义在表 22 中列出。PAT, CAT, TSDT 的 PEID 字段是预先定义的。

接收机通过搜索网络缓冲中带有 PEIT_indicator 的传输包来寻找 PEIT 表。在找到 PEIT 后,接收机首先需要定位和解析 PAT。也就是说,在重建并得到 PEIT 表之后,在 PEIT 表中使用 0x0000 做为搜索标志符寻找 PAT 的传输包在 PEIT 区间的位置。一个或多个 PAT 传输包重建构成完整的 PAT 表。兼容 MPEG2 系统的 PAT 表中列出了节目号(program_number)和 _Program_map_PID(直接对应本标准所定义的 Program_map_PEID)。

为了得到所选频道的基本流节目成分列表(比如音视频等),接收机搜索相应的 PMT。具体地,使用 PAT 中所选频道对应的 Program_map_PEID 在 PEIT 表中搜索,从而得到该 PEIT 区间内相应 PMT 表的传输包。每个节目的 PMT 都包含该节目中各成分的列表和相应的 Elementary_PEID。同样地,所需要的其它节目成分类型的 PEID(比如 Network_PEID, clock_recovery_PEID, CA_PEID 等)也可以从传输流中通过本标准所定义的语义解析方式而得到。

通过以上方式,接收机就可以从各个节目号所对应的 Program_map_PEID,以及各个节目中节目成分和相应的 Elementary_PEID,以及其它所需要的节目成分类型的 PEID(比如 Network_PEID, clock_recovery_PEID, CA_PEID 等),构建完整的传输流的节目成分识别系统。

L.1.2 节目成分的识别过程

构建得到完整的传输流的节目成分识别系统后,解复用端就可以通过节目成分信息表的解析以及所得到的传输包在所在区间内的顺序和位置关系来识别传输包序列中的节目成分。在基本模式下,接收端假设传输包到达的顺序完全符合 PEIT 表中的描述。之后,接收端分析收到的 PEIT 表,按照表中所提供的排列顺序信息和计数信息来识别和提取相应的节目成分。假设接收端收到的是图 L.1 中所示的传输流。为了简化说明解复用的过程,假设在传输过程中没有 PEIT 信息和传输包的误码。另外,在下面的说明中考虑基本复用模式的情形。高级复用模式的说明在 L.2 中阐述。

如图所示,解复用器收到第一个节目成分信息包 PEIT0 后,解析节目成分信息表。从表中的结构和节目成分的顺序得知:在该 PEIT 区间内,前面两个传输包分别是 PAT 和节目 2 的 PMT 表,解复用器把它们发送到 PSI 处理器,完成如节目列表的构建或更新等。之后接收端收到了 4 个传输包。从 PEIT 表中得

知, 节目 1 的音频包计数为 1 而视频包传输计数为 2. 所以,解复用器把它们分别发送到节目 1 的音频解码缓冲和视频解码缓冲. 之后, 解复用器把 V2 发送到节目 2 的视频解码缓冲. 最后, CAT 传输包被发送到 PSI 处理器以提取条件接收信息.

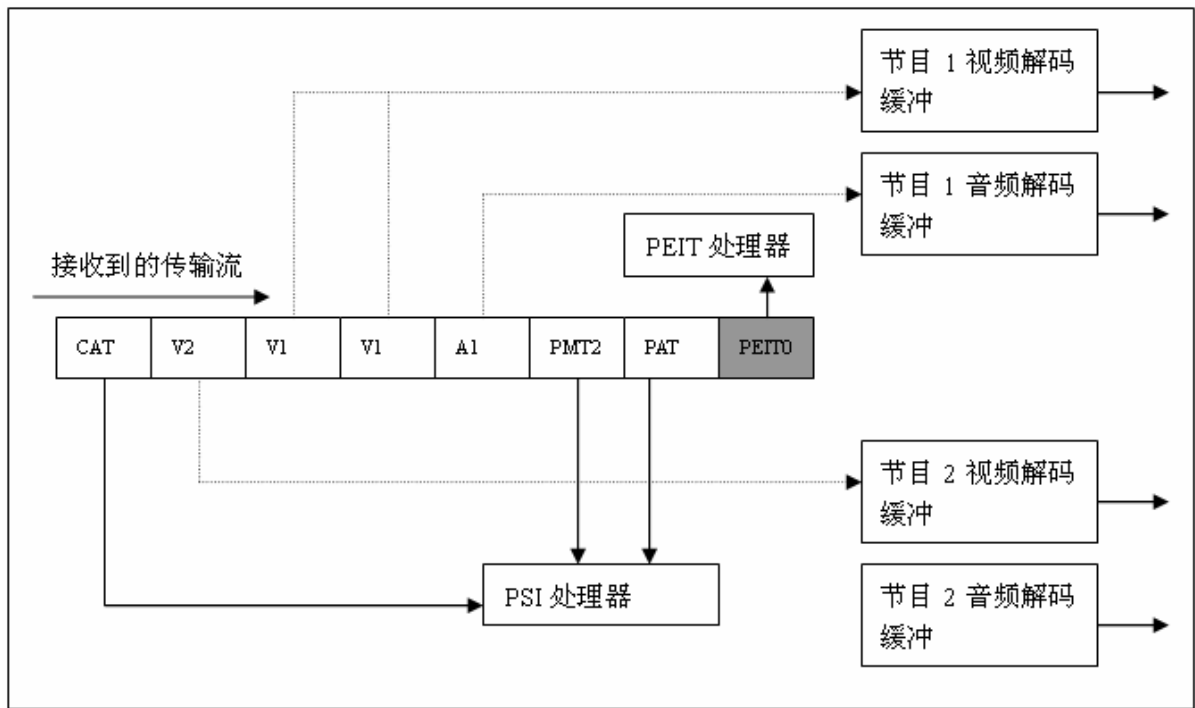


图 L.1: 解复用过程示意图

当传输包含 PEIT 段的时候, PEIT_indicator 值为 0x000F. 需要和 MPEG2 反向兼容的时候, PEIT_indicator 字段可以作为 MPEG2 的 PID 使用, 但是 MPEG2 系统中的 PID 选择应该避免使用 PID=0x000F 作为除传输 PEIT 私有段之外的其它用途.

L.2 高级模式解复用

高级模式的解复用过程在上述基本模式的解复用过程上, 增加了以下功能:

- 1) 包关联技术: 主要用于误码纠错.
- 2) 包定位技术: 用于定位传输包的位置.

在解复用端的节目成分识别过程中, 有可能因为信道误码而导致 PEIT 信息表不可用的情况. 另外, 也希望在接收端实现低延迟的复用流随机访问和频道切换. 在这里, 提出参考传输包(AP: Anchor Packet)和包关联映射表(PLT: Packet Link Table)的概念来提高复用流的抗误码和随机访问能力. 其中 AP 把其位置信息和具有同一节目成分的被链接的传输包(LP: Linked Packet)进行关联. 复用端使用 PLT 表来动态更新 AP 的位置信息并写入 LP 的头信息中. 在 PEIT 发生误码的情况下, 解复用端通过搜索重建的 PLT 表来识别被链接的传输包的所属节目成分. 如图 L.2 的例子所示, 即使 PEIT2 表发生丢失或误码不可用的情况, 节目成分 V2 也可以根据从 PEIT0 序列中的 V2 所提供的关联信息而得到识别.

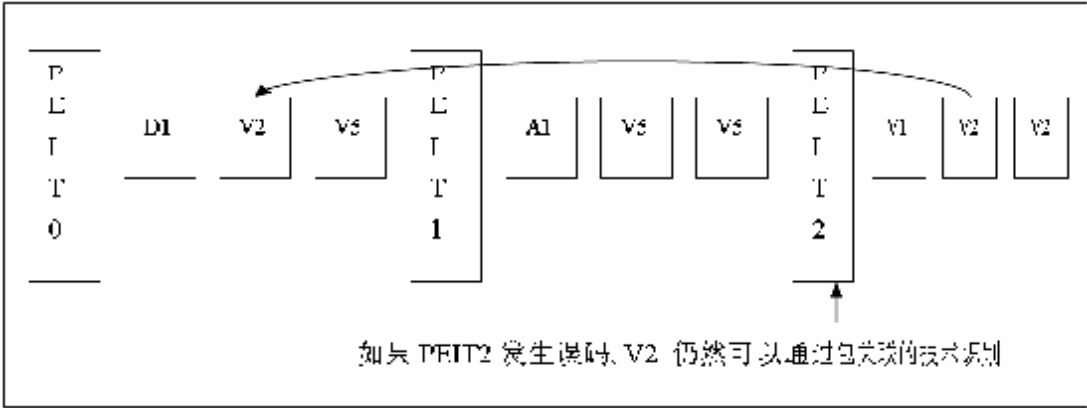


图 L.2: 使用包关联技术的误码纠错示意图

解复用端通过重建的 PLT 来进行误码纠错。解复用器检查每一个收到的传输包的头信息适应字段，如果 `avs_specific_transport_data` 字段的 `anchor_flag` 标志位中显示该传输包是 AP，那么该传输包的位置信息（包括 `PEIT_seq_num` 和 `TP_seq_num`）将被写到 PLT 表中和该传输包节目成分所对应的条目中。需要注明的是，这个写入过程是动态的，也就是说系统每一次收到同一节目成分的 AP，相应的条目将被更新。

如果某一个 PEIT 区间的 PEIT 描述信息表发生误码,那么系统仍然可以识别属于 LP 类型的传输包。具体地，系统将根据该 PEIT 区间中属于 LP 类型的传输包头信息中的 AP 位置（包括 `AP_PEIT_seq_num` 和 `AP_TP_seq_num`）搜索 PLT 表的条目,如果发现 AP 位置信息存在,那么相应的节目成分就可以通过 PLT 映射表得到。由此分析，通过 PLT 的关联来辅助识别节目成分，系统的抗误码能力有大幅度的提升。在图 L.2 中说明了一对一关联的情形。事实上,同样可以实现一对多(一个 LP 对应多个 AP)的关联。PEIT 表中的 `num_of_anchors` 字段表示当前 PEIT 区间内传输包被关联的 AP 的数目。

有关复用过程的说明参见附录 M。

附录 M

(资料性附录) 系统复用过程

M.1 复用过程

复用端和解复用端共同维系一个双方默认的节目成分排序法则,这个排序法则提供了多个节目及其节目成分在序列中的排放顺序。节目成分排序法则可以预先设定。

复用端设置一个重排缓冲,该重排缓冲从信号源端接收信息包并按照所选节目成分排序方式进行排序。信号源可以直接来自音视频等实时编码器也可以是已经复用好的传输流。重排缓冲包含一个计数器,它对进入缓冲的信息包个数进行计数,以确保存入预先指定数量的传输包,并对归属于相同节目和相同节目成分的同一类型的传输包进行分别计数,所得到的相应计数信息被记录下来。重排完成后得到的传输包序列所对应的节目成分排列顺序应该完全符合预定的排序方式。

重排过程应该遵循一些约束条件,通常包括:

- 1) 重排过程产生的传输流应该符合 STD 的约束条件,比如确保解码缓冲既不上溢(overflow),也不下溢(underflow)。
- 2) 同一种节目成分的传输包之间的相对顺序不应被改变。
- 3) 生成的复用流应当小于 5ms 的抖动上限。

图 M.1 说明了重排过程的一个例子。

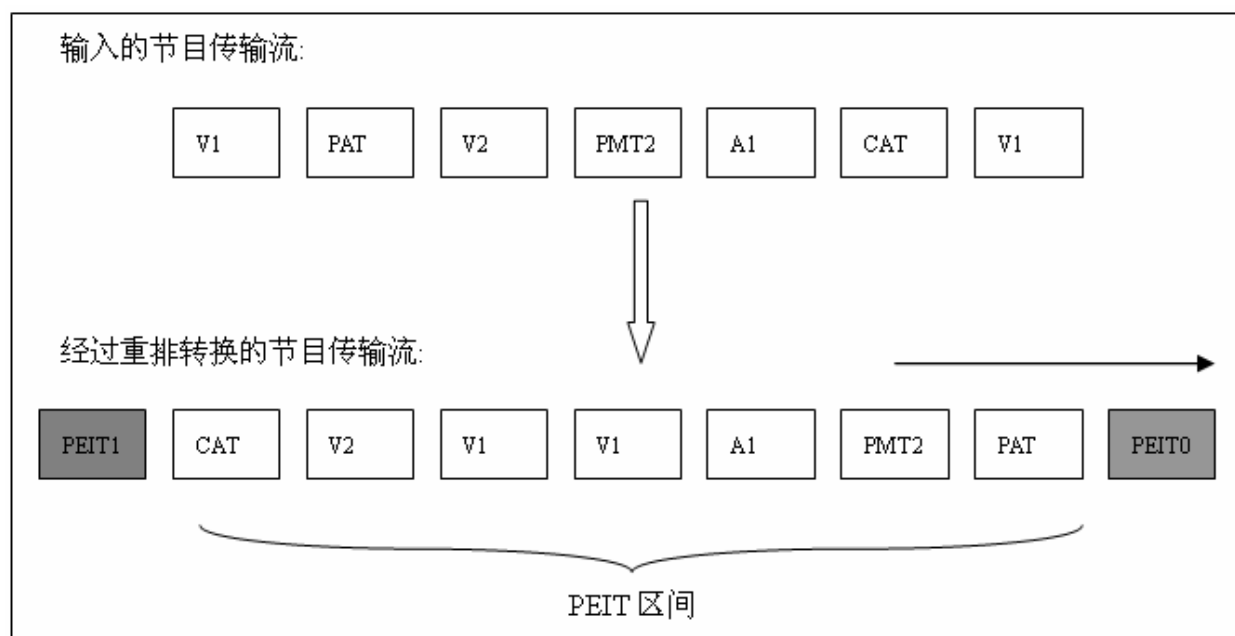


图 M.1: 节目信息包重排过程示意图

在图 M.1 中, 输入的是一个需要进行重排转换的信息包序列。其中 PAT, PMT2, CAT 是 PSI 信息包(其中 PMT2 是节目 2 的 PMT 表), V1,A1 分别表示第一个节目的视频和音频传输包,V2 表示第二个节目的视频传输包。如图所示, 经过重排转换后, 节目 1 和节目 2 按照节目的序号顺序排列, 而每个节目中的音频, 视频传输包也分别按照排序法则所指定的先后顺序排列。PSI 信息包按复用器在当前重排区间内所选择的顺序排列。需要注明的是排序法则的选择对复用器来说是灵活多样的。需要考虑的除了之前提到的约束条件和其他因素等, 还包括最大程度地压缩对重排后所得到的序列进行描述的节目成分信息表的空间(利用重排可以使得相邻的同一节目成分的传输包可以合并为节目成分信息表的一个条目)。

复用端在节目成分信息表中设定一种所选取的节目成分排序方式。在节目成分排序方式确定后, 节目成分信息表的基本结构, 比如, 节目和节目成分的在表中排列方式也是确定的。重排后的 PEIT0 的信息表结构图(对应于图 M.1 的传输流序列)如图 M.2 所示。

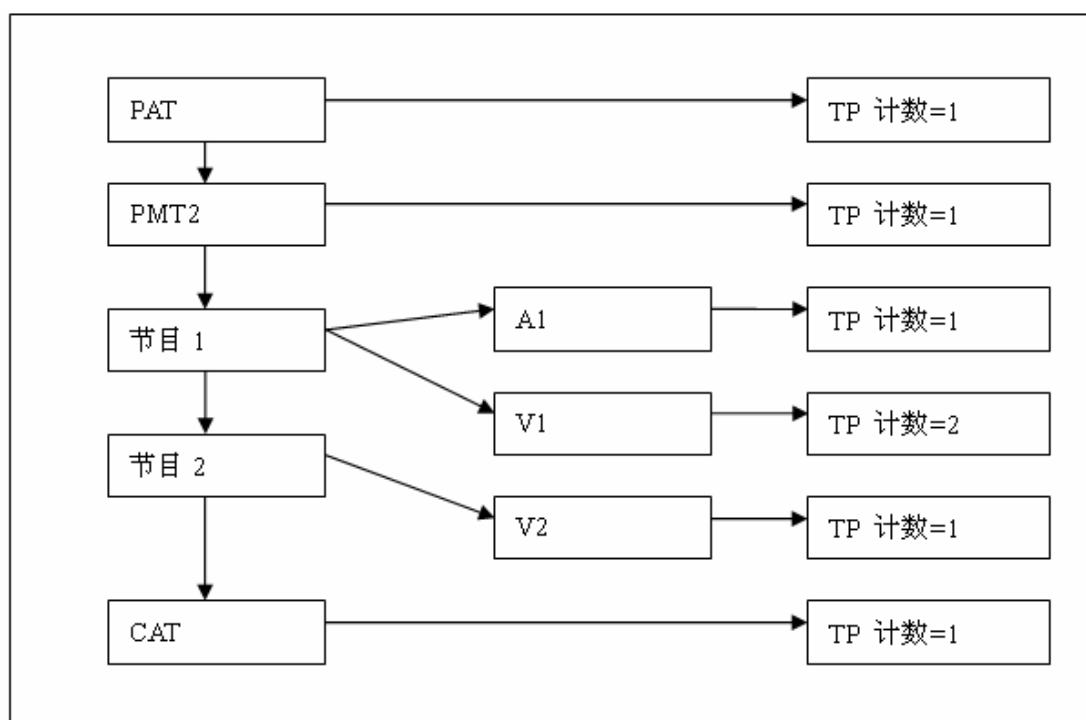


图 M.2: PEIT 节目成分信息表结构图。

在上面所述的传输包重排过程完成后, 输出的传输包序列称为 PEIT 序列(PEIT Sequence)。而单个 PEIT 表(table)或段(section)所描述的传输包序列称为 PEIT 区间(PEIT interval)。PEIT 区间的长度可以根据具体的应用场合由复用器确定, 它可以是可变的, 也可以是定长的。通常地, 建议 PEIT 区间选择为定长的。在重排的过程中, 复用器对连续出现的具有相同节目成分的传输包进行计数。如图 M.2 所示, 这个数值将被写到 PEIT 树结构中所对应的节目成分的“树叶”中。

完成后的 PEIT 表将被打包插入到其所描述的 PEIT 区间的前面, 然后被发送到传输网络中去。PEIT 表的具体位置并不需要直接毗邻所描述的 PEIT 区间。而为了便于实现, 它通常应该被提前放置。建议提前一个区间传输 PEIT 表, 这样可以给接收机足够的解复用反应时间。接收机可以通过如 6.4.2 中所定义的 PEIT_interval_pointer 得到当前 PEIT 表所对应的 PEIT 区间开始的位置。如图 M.3 所示, PEIT0 和 PEIT1 分别被封装在一个段中, 并分别打成 M0 和 M1 个 TP 包, 并且其对应的区间的长度为 N0 和 N1, 那么复用器可以在 PEIT0 的后面插入 K 个空包, 然后将该表中 PEIT_interval_pointer 字段设置为

$M0+M1+K-1$. 然后在 PEIT1 的后面插入对应于 PEIT0 的 $N0$ 个传输包, PEIT2 表的后面插入对应于 PEIT1 表的 $N1$ 个传输包,依此类推.

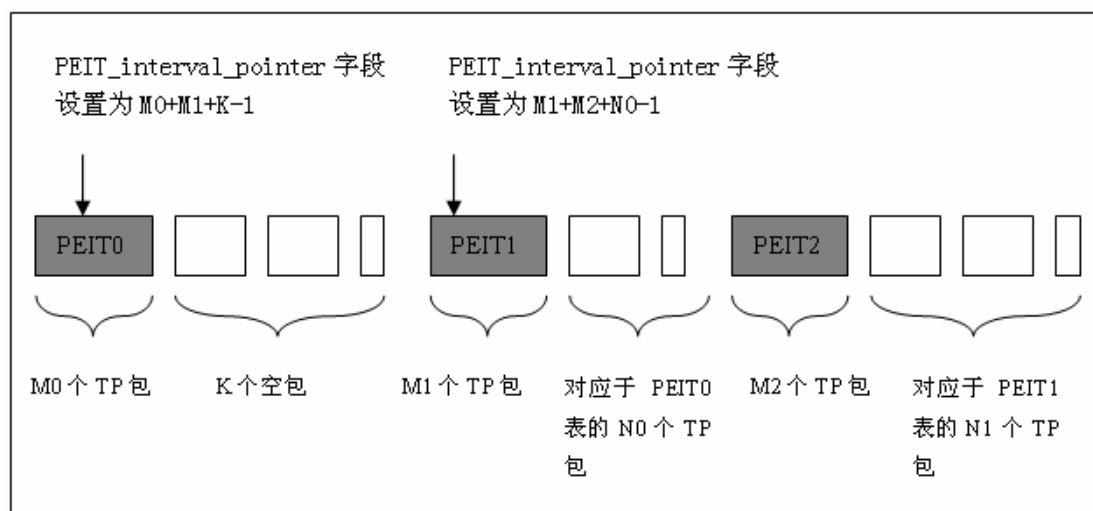


图 M.3: PEIT 表的提前放置

PEIT 的传送间隔应该不大于 PAT 的间隔时间. 在设置 PEIT 区间的长度时候, 需要权衡延迟和传输开销这两个因素. PEIT 间隔越小, 显然延迟也越小, 但是传输带宽的开销也越大. 另外, PEIT 序列的重排会在复用端引入一个延迟. 这个延迟通常是很小的. 对于 PEIT 区间为变长的复用方式, 重排缓冲延迟应该设置为 $\max\{\text{PEIT_interval_size}(n)\}$, 以避免不必要的复用抖动. 其中 $\text{PEIT_interval_size}(n)$ 为复用器所选的 PEIT 区间大小的集合. 如果复用器所选 PEIT 区间的最大长度为 255, 那么对于 20Mbps 的传输流来说, 这个延迟在 18 毫秒左右. 对于数字电视广播类型的应用, 这通常是可以接受的.

如果 PEIT 表包含的节目信息数据字节数大于单个传输包所能容纳的空间, 那么 PEIT 将分为两个或以上的连续的传输包进行传输.

在高级复用模式的传输流中, PEIT 表的定位信息单独由 PEIT_Seq_Num 来表示, 而在 PEIT 序列中的传输包由 (PEIT_Seq_Num, TP_Seq_Num) 来定位. 其中 PEIT_Seq_Num 是 PEIT 在传输流中的高级复用模式 PEIT 的序列号, TP_Seq_Num 是传输包在该 PEIT 区间中的序列号. 如果 PEIT_Seq_Num 和 TP_Seq_Num 都从 0 开始计数, 那么第二个高级复用模式 PEIT 区间的第五个传输包的位置是 (1, 4). PEIT_Seq_Num 在达到比特位数所能表示的最大值后应当重新置 0. 一个 PEIT 区间的长度最大不超过 255. 序列号定位信息可以放入适应字段的 `avs_specific_transport_data` 字段中传输.

复用器对连续输出的 PEIT 序列进行 AP 的选择以及标志位的设置. 选择的对象可以是在 PEIT 区间内首次出现的节目成分所对应的传输包(当然也可以是其它位置的传输包, 但通常只需选择一个). AP 通过设置一个标志位以表明它的存在, 该标志位放入适应字段的 `avs_specific_transport_data` 字段中. 选中的 AP 包的位置将被更新到 PLT 表中. 同时系统对该 PEIT 区间之后所有 PEIT 区间的收到的节目成分传输包进行检测. 当同一节目成分的传输包再次出现时, 之前被更新的 AP 的位置信息从 PLT 表中取出, 并写到这些传输包的头信息中. 需要注意的是, 为了提高抗误码的健壮性, PEIT 区间的所有传输包都应当被写入相应的 AP 位置信息. 另外, 由于 AP 不需要知道被关联的下一个同一节目成分的传输包在哪里出现, 所以, 复用端系统的延迟性能并不因此机制受影响.

M.2 复用模式的选择

不同网络传输情况和不同复用系统设计对传输质量有很大的影响。比如在传输信道发生误码的情况下,有可能导致 PEIT 序列中的一个或多个传输包或 PEIT 信息表传输包本身发生丢失的情形。另外,在服务质量(QoS: Quality of Service)不能保证的 IPTV 网络中,包不按顺序抵达接收机的情况也是时有发生。

基本复用模式的 PEIT 区间的传输包不需要带有任何序列号等定位信息,这样就大大地节省了传输带宽。在基本模式中,节目成分的定位是依靠被描述的传输包和 PEIT 表的相对位置。由于数字有线电视广播等信道误码率较低,另外传输包丢失的情况是很少见的,所以基本模式可以符合传输的质量要求。当然,在实际的复用器的实现中,比如为了匹配恒流模式(Constant Bit Rate)的传输信道,复用器通常会往 PEIT 序列中插入一些填充包。在这种情况下,系统就需要更新 PEIT 表,也就是说把填充包也当成 PEIT 表的一个结构单元加入 PEIT 的结构体中以描述其在 PEIT 区间内的位置。

高级复用模式实现了抗误码丢包和抗包乱序等高级功能。高级复用模式通过包关联技术来实现误码纠错。而包定位通过引入节目成分信息表和传输包的序列号的机制来实现。

复用模式的选择可以精确到某一个 PEIT 区间。也就是说,每一个 PEIT 区间都可以选择自己的复用模式。这样就增加了系统设计的灵活性。PEIT 通过一个 advanced_mode_enable 的标志位来标识被描述的 PEIT 区间的复用模式。advanced_mode_enable 标志位被设置为'1'的 PEIT 区间是高级复用模式,PEIT 表中将带有 PEIT 序列号,同时其对应的 PEIT 区间内将含有一个以上的 AP 或 LP,每个 AP 和 LP 都将带有 adaptation field。avs_specific_transport_data 字段将包含 AP 和 LP 的标志位以及定位信息。PEIT 序列号通常只需在开启高级复用模式的 PEIT 区间中计数递增。而高级复用模式 PEIT 区间中只需要对 AP 或 LP 传输包进行定位。也就是说,如果某 PEIT 区间内只含有一个 AP/LP,那么除该 AP/LP 之外的其它传输包都不需要包含定位信息。PEIT 重新置 0 代表一个新的定位区间的开始,所以 PLT 表也应该被重新置位(reset)。