GRAND ALLIANCE
HDTV SYSTEM SPECIFICATION

Version 2.0

December 7, 1994

Table of Contents

SCOPE OF DOCUMENT

This document details the System Specification of the Grand Alliance HDTV System, and is intended to form the basis for the documentation of the proposed standard. It is comprised of the documents that have evolved from the work of the Grand Alliance Specialist Groups with the guidance and cooperation of the Advisory Committee on Advanced Television Services (ACATS) Experts Groups. This system specification document also details the prototype hardware, currently under construction. This prototype implementation will be delivered to the Advanced Television Test Center (ATTC) for verification of the performance of the proposed Grand Alliance HDTV system standard.

Executive Summary

Layered System Architecture with Header/Descriptors

The Grand Alliance HDTV System is a layered digital system architecture that uses headers/descriptors to provide flexible operating characteristics. The layers of the GA HDTV System, and some of their most important capabilities, are:

- the Picture layer provides multiple picture formats and frame rates
- the Compression layer uses MPEG-2 video compression and Dolby AC-3 audio compression
- the Transport layer is a packet format based on MPEG-2 transport, that provides the flexibility to deliver a wide variety of picture, sound and data services
- the Transmission layer is a Vestigial Sideband signal that delivers a net data rate of over 19 Mbps in the 6 MHz simulcast channel

While all of the GA HDTV system's layers operate in unison as an effective simulcast system, each layer has also been designed to have outstanding interoperability. The GA HDTV system's layered digital system approach with header/descriptors will create interoperability among a wide variety of consumer electronics, telecommunications, and computing equipment.

Picture Layer

The picture layer consists of raw pixel data, organized as pixels, scan lines and frames. The GA HDTV system provides for multiple formats and frame rates, all of which will be decoded by any GA HDTV receiver. This approach allows program producers and application developers to make their own tradeoffs among resolution, frame rate, compression artifacts and interlace artifacts, and to choose the format that is best for their particular use. The formats are:

Spatial Format (X x Y active pixels)	Temporal rate
1280 x 720 (square pixels)	23.976/24 Hz progressive scan
	29.97/30 Hz progressive scan
	59.94/60 Hz progressive scan
1920 x 1080 (square pixels)	23.976/24 Hz progressive scan
	29.97/30 Hz progressive scan
	59.94/60 Hz interlaced scan

Compression Layer

The compression layer transforms the raw video and audio samples into a coded bit stream -- essentially a set of computer instructions and data that are executed by the receiver to recreate the pictures and sound. The compression layer of the Grand Alliance HDTV system:

- uses video compression syntax that conforms to the ISO-MPEG (International Standards Organization — Moving Picture Experts Group) MPEG-2 video data compression draft standard, at a nominal data rate of approximately 18.9 Mbps.

- uses Dolby AC-3 audio compression to provide 5.1 channel surround-sound at a nominal data rate of 384 kbps.

Transport Layer

The transport layer separately packetizes video, audio and auxiliary data and allows their mix to vary dynamically, providing the flexibility needed to innovate new services and new kinds of programming. The transport layer of the Grand Alliance HDTV system:

- uses a packet format that is a subset of the MPEG-2 transport protocol.

- provides basic service capabilities that include video, five-channel surround-sound audio and an auxiliary data capacity.

- offers great flexibility in the mix of video, audio and data services that can be provided to appropriately-featured receivers. It separately packages each type of data (e.g., video, audio, etc.) in its own set of transmission packets. Each packet has a Packet ID header that identifies the content of the data stream. This capability enables the creation of new services, ranging from many stereo channels of audio, to broadcast distribution of computer software, to the transmission of very high resolution still images to computers.

- allows the mix of services to be dynamically allocated. This capability will allow rapid burst-mode addressing of receivers. It will also enable broadcasters to send multiple “streams” of video, audio and data programming to their audience, all complementing or enhancing the basic program content. This capability can fundamentally change the nature of television programming, since it enables software to be broadcast to “smart receivers” that can operate in conjunction with the HDTV picture and sound. With this capability, HDTV will likely become a more interactive medium than today’s television, enabling new forms of educational and entertainment programming and games.

- provides important extensibility, since a Grand Alliance HDTV receiver will disregard any packet with a PID header that it does not recognize or cannot process. This will eliminate future “backward-compatibility” problems in the installed base of receivers, removing a crucial constraint from the introduction of new services.

Transmission Layer

The transmission layer modulates a serial bit stream into a signal that can be transmitted over a 6 MHz analog channel. The transmission layer of the Grand Alliance HDTV system:

- uses a trellis-coded 8-VSB modulation technique to deliver approximately 19.3 Mbps in the 6 MHz terrestrial simulcast channel

- provides a pilot tone that facilitates rapid signal acquisition and increases pull-in range.

- provides a training signal that facilitates channel equalization for removing multipath distortion.

- provides a related 16-VSB modulation technique to deliver two 19.3 Mbps data streams in a 6 MHz cable television channel

Summary

The GA HDTV system has the flexible operating characteristics that allow it to provide broad interoperability needed to form the basis for new and innovative services and applications of HDTV in many industries.

Chapter 1

SYSTEM BACKGROUND

1.1. System Background

A successful HDTV standard will be used for HDTV delivery to the public by terrestrial broadcasting and a variety of alternate delivery media, and it will also form the basis for new services and new applications of HDTV across a wide range of industries. In order for the enormous potential impact of digital HDTV to be realized, a high degree of interoperability is an essential attribute of an American standard.

1.1.1 Introduction

The formation of the Grand Alliance moves the HDTV standardization process from a competitive phase into a new collaborative phase. The previous competitive phase stimulated the rapid development of technology, as well as broad industry debate over various attributes in which the proposed systems differed. In the new collaborative phase, the Grand Alliance and the Advisory Committee must rapidly establish a national consensus and move forward into laboratory and field performance verification, and the documentation of a standard for FCC approval. Failure to achieve these objectives will harm every industry that is potentially advantaged by the deployment of HDTV systems and technology.

In some cases where the Grand Alliance confronted the dilemma of conflicting goals, it was technically feasible and reasonably cost-effective to allow a multiplicity of modes within the system. In these cases, the presence of modes favored by a particular industry do not preclude the inclusion of modes that are more favorably viewed by a different industry. The result is a system that is maximally useful to all industries, rather than one that burdens one industry at the expense of another.

In other cases where the Grand Alliance confronted the dilemma of conflicting goals, a choice had to be made among the conflicting alternatives. In these cases, there are good technical reasons for the choice that was made, and careful consideration has been given to interface requirements that bridge the differences between the approaches of different industries and make the system interoperable.

1.1.2 Scope of HDTV Standards and Their Impact

HDTV technology will not only revolutionize the television industry, but it will have a profound impact across a wide range of industries. The proliferation of high speed digital communications needed to deliver HDTV to the home will make the associated industries a key part of our National Information Infrastructure. Further, the core capability of cost-effectively delivering full-motion high resolution pictures and high-quality surround-sound will become a cornerstone of image communications that may be adopted by a variety of industries and applications.

However, it is important to understand the scope of the GA HDTV System, which is a proposed transmission standard that is under consideration by the FCC through its Advisory Committee on Advanced Television Service (ACATS). The Grand Alliance HDTV system (and presumably the FCC standard) is NOT a production standard, or a display standard, or a consumer equipment interface standard. In fact, a digital compression and transmission system effectively DECOUPLES these various elements of the end-to-end imaging chain.

A digital HDTV transmission standard must specify sufficient information that allows any manufacturer to build a receiver that can receive a radio frequency (RF) transmission, and from it, produce pictures and sound. For a digital system, this means specifying the meaning of a bit stream and the signal format for transmitting those bits in a 6 MHz television channel. Thus, the picture formats, the compression syntax, the packet format and the RF signal format are examples of items that require standardization. While these standardized elements establish a minimum functionality for HDTV encoders and receivers, they do not specify encoder or receiver implementation. In the opinion of the GA, this traditional approach to transmission standards is highly desirable, because it enables natural competitive and economic forces to influence the implementation of encoders and receivers, which can evolve over time as underlying technology as cost changes take place.

1.1.3 Grand Alliance Goals

Since we are considering a proposed HDTV transmission standard that contains many complex tradeoffs, it is important to state the design goals that the GA established for itself. These goals essentially summarize the selection criteria that were established by ACATS during the competitive phase of the standards process. The GA design goals are to provide:

- High quality HDTV pictures and sound
- An effective simulcast system that:
    - Provides wide HDTV service area for terrestrial broadcast
    - Avoids unacceptable interference to existing NTSC service during its lifespan
- A cost effective solution for consumers, producers and all users at the time of introduction and over the life of the HDTV standard
- Interoperability with other media (e.g., cable, satellite, computers, etc.) and applications
- The potential for a worldwide standard

In this simple list of design goals, there are many conflicting desires. For example, there is a three-way tradeoff among HDTV picture quality, HDTV service area and interference to existing NTSC service. In order to have outstanding HDTV picture quality, a high data rate is desired. But achieving a high data rate over a large HDTV service area requires high power transmission that results in too much interference to existing NTSC service. Reducing the HDTV service area is unacceptable, because there would be insufficient audience to make the service economically viable. Fortunately, with some balancing of objectives and associated engineering tradeoffs, the combination of video compression technology and digital transmission technology is sufficient to enable the creation of an HDTV simulcast service.

As another example of conflicting goals, even when considered solely as an entertainment medium, the HDTV transmission standard must have a high degree of interoperability with its common production sources, in order to ensure economical operation from production through delivery to the home. Even in this limited context, interoperability of one kind may be in conflict with interoperability of another kind. The production sources for HDTV transmission will include: the HDTV production standard, film, computer graphics, and NTSC video. The conflicting interoperability desires are that while the HDTV production standard will likely be interlaced with square pixels at 59.94 Hz, the desire for NTSC interoperability argues for preserving non-square pixels and the 59.94 Hz temporal rate, while the desire for film interoperability argues for a 24 Hz temporal rate, while the desire for computer interoperability argues for square pixels, progressive scan and possibly a frame rate higher than 70 Hz. Clearly, these conflicting interoperability desires are mutually exclusive, yet the interoperability of an HDTV transmission standard with all of these sources is important. A solution can be found that takes into account existing and widely accepted interoperability practices, such as the conversion of 24 Hz film to 59.94 Hz through a combination of a 1000/1001 slowdown to 23.976 Hz and a 3:2 frame repeat sequence to speed up the temporal rate to 59.94 Hz. In addition, the conversion between square pixels and non-square pixels must take place at some interface boundary; the question is simply where. Understanding the technical difficulty and cost of one conversion compared to another is an important element of the complex tradeoffs that must be made in the design of an HDTV system.

To further complicate our decision space, we observe that HDTV is on the vanguard of an explosion in digital technology. HDTV products will exist in a world where digital delivery of television by cable, satellite and pre-recorded media are proliferating; where the availability of low cost computational power and digital data networks are enabling the creation of a National Information Infrastructure; and where the development of new interactive services and applications will drive the creation of information appliances and products that may be hybrids of today's products, or entirely new products based on technologies that rapidly cross the computing, communications and consumer electronics industries.

1.2. System Overview

The Grand Alliance HDTV system employs two fundamental system principles that make it a highly interoperable system. First, it is designed with a layered digital system architecture, that results in the ability to interface with other systems at any layer. This means that many different applications can make use of various layers of the HDTV architecture. In addition, the GA has designed each individual layer of the system to be highly interoperable with other systems at corresponding layers. Secondly, the Grand Alliance HDTV system takes full advantage of the potential flexibility offered by a digital approach by using header/descriptor design principles that allow maximum flexibility to be achieved. As a truly flexible digital system, the Grand Alliance HDTV system provides the basis for an HDTV standard that will have a revolutionary impact on many industries and applications that are well beyond the scope of television today.

1.2.1 Layered Digital System Architecture (Modified 12-7-94)

The GA HDTV system is a layered digital system that consists of four primary layers: the Picture layer, the Compression (video and audio) layer, the Transport (packetization) layer and the Transmission layer. These layers are conceptually illustrated in Figure 2.1.

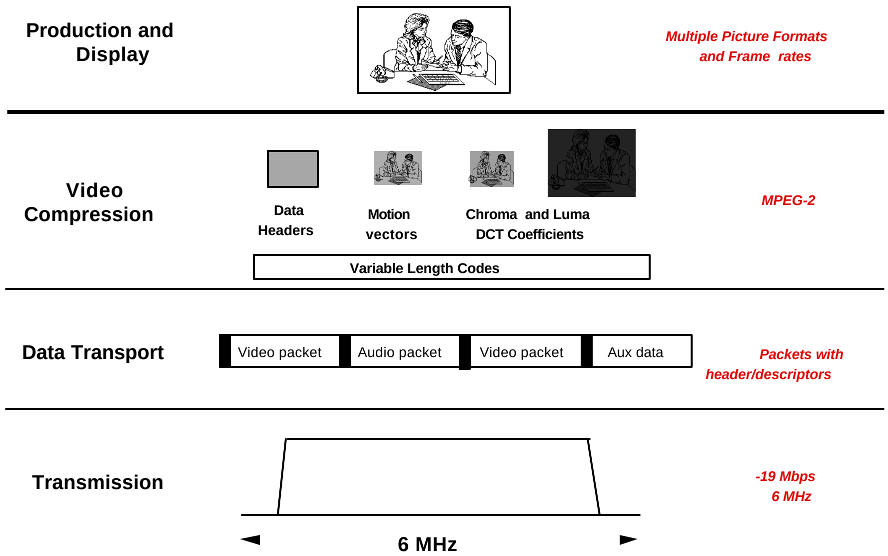


Figure 1.2.1 - The GA HDTV system's layered architecture and the PS-WP4 reference model.

While Picture layer Production and Display standards are not within the scope of a transmission standard, header/descriptors in the compression layer of the GA system support the transmission of multiple picture formats and frame rates that are related to both high-definition video and film production, and that are also appropriate for computer-based applications. At the Video Compression Layer, the GA HDTV system is based on MPEG-2 compression. At the transport layer, the GA HDTV system is also based on MPEG-2, which uses a flexible ATM-like packet protocol with headers/descriptors. At the Transmission Layer, the VSB modulation system will provide a net user data rate of over 19 Mbps delivered in the 6 MHz terrestrial simulcast channel.

The GA HDTV system has been designed to provide interoperability at every layer of its architecture, and the PS-WP4 reference model provides an excellent framework to explain its interoperability features. Figure 1.2.2 shows the protocol stack that consists of the layers of the GA HDTV system (shown on the left) expanded in a form that corresponds to the reference model established by PS-WP4 (shown in the middle) as a basis for evaluating interoperability. Also shown is the ISO Open Systems Interconnect (OSI) reference model for data communications (on the right).

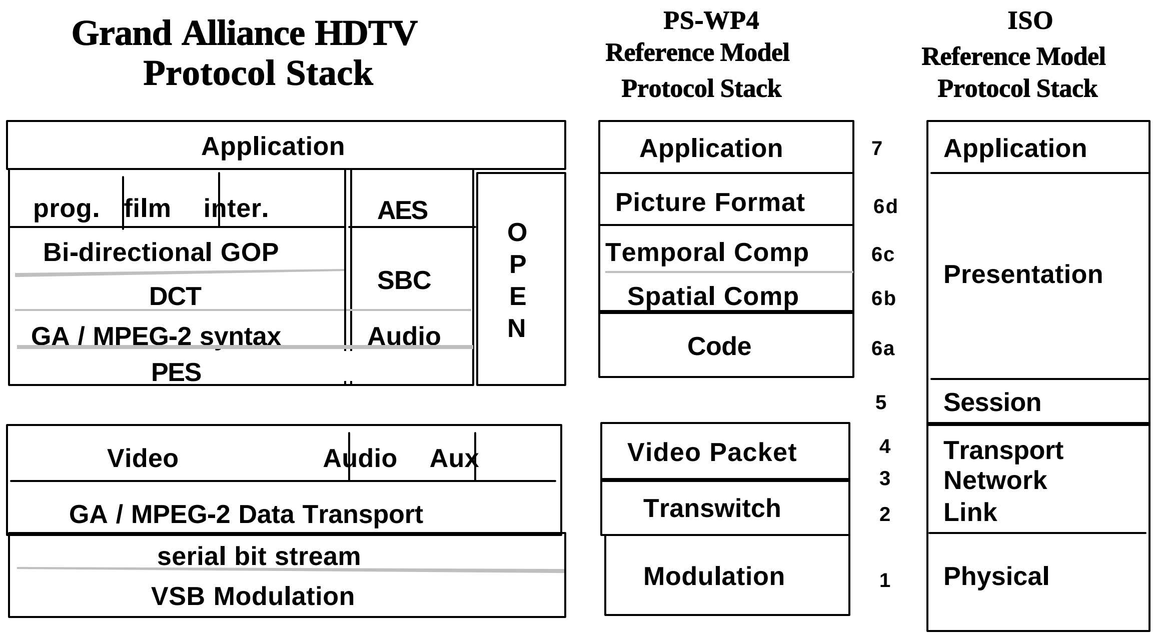


Figure 1.2.2 - The GA HDTV system's layered architecture and the PS-WP4 reference model

At the picture layer, progressive scan formats are provided at both video and film frame rates, in addition to an interlaced format. The compression layer conceptually consists of temporal compression, spatial compression and code (syntax) sublayers. For temporal compression, the MPEG Group of Pictures (GOP) approach is used to allow bi-directional motion-compensation across a block of frames. Spatial compression is achieved through the Discrete Cosine Transform (DCT). For the compressed code representation, the GA compression syntax is based on MPEG-2. The Program Elementary Stream (PES) layer is the fundamental data stream that comprises a compressed video bit stream. At the video packet/transswitch layer, the GA system uses the MPEG-2 packet format, which provides the capability to synchronize the delivery of video, audio and auxiliary data packets in an ATM-like multiplexing environment. Finally, at the transmission layer, the serial bit stream is converted to a 6 MHz signal by Vestigal Sideband Modulation. Subsequent chapters of this document will describe each layer of the architecture.

Chapter 2

VIDEO PICTURE FORMAT

2.1. Introduction

This document provides more detailed specifications of the scanning formats proposed on June 30, 1993 by the Grand Alliance, and includes modifications made since June 30 to the initial proposals. The document reflects extensive and valuable interactions between the Grand Alliance Specialist Group on Formats and the ACATS Technical Subgroup's Expert Group on Scanning Formats and Compression.

The parameters needed for specification of the HDTV systems are in two categories: those related to the interface to the encoder (on the transmission side), and those related to the encoding within the encoder in preparing the compressed data for transmission.

For the interface specification, the total line and pixel counts and the pixel clock rate are important. We have chosen to restrict the interface variations for the prototype to two formats, with 720 active lines and 1080 active lines. This limitation does not preclude other natural extensions for frame rates or for progressive scanning when such extensions become useful and equipment for new format variations is available.

For specifying the transmission format, which deals with the compressed representation of the active picture information, the total line and pixel counts are not relevant, but the frame rate variations can be distinguished and coded according to their frame rates are significant.

Parameters to be specified

The parameters that need to be specified for the interface and transmission formats include the following:

A. Number of frames per second.

B. Number of fields per frame (i.e., whether interlaced or progressive scanning is specified).

C. Number of active lines per frame.

D. Total number of lines per frame.

E. Number of active video samples per line.

F. Total number of video clock cycles per line.

G. Sampling frequency for video samples (derivative from the above specifications).

H. Allowable combinations of parameters.

Note that the HDTV system proposed by the Grand Alliance intrinsically includes more than one format, so that for some parameters there is more than one allowable value. Also note that 59.94 Hz and 60.0Hz frame/field rates are considered implementation variations, not defining separate formats.

2.2. Format Specifications

The Grand Alliance proposal includes two main format variations, with different numbers of lines per frame. These formats have 720 active lines and 1080 active lines per frame.

For interfacing to the encoder, the 720-line format uses 1280 active samples per line, while the 1080-line format uses 1920 active samples per line. These choices yield square pixels for all formats. Also for interfacing to the encoder, the 720-line format is progressively scanned, with 60 Hz (nominal) frame rate, while the 1080-line format is interlaced 2:1 with a 60 Hz nominal field rate. The prototype encoder input accepts 787.5 total lines per frame and 1125 total lines per frame for the two interface formats.

For compression and transmission, the frame rate for the 720-line format and for the 1080-line format can be 60.0 Hz, 24 Hz, or 30 Hz. The same formats are also supported with the NTSC-friendly numbers, i.e., 59.94 Hz, 23.98 Hz, and 29.97 Hz.

The 60.0 Hz and 59.94 Hz variations for the 1080-line format are encoded as interlaced scanned images for the initial HDTV systems. Other formats are encoded as progressively scanned.

2.2.1. Number of frames per second

The frame rates for the Grand Alliance HDTV system will include both 59.94 Hz and 60.0 Hz, since the 59.94 Hz rate may simplify interworking with NTSC material during the simulcast period, and because the 60.0 Hz rate may have interoperability benefits in the future.

The proposal includes 24 Hz and 30 Hz (including the corresponding adjustments for the NTSC-friendly frequencies) as film modes. This means that the encoder will be designed to encode the reduced pixel rate from image sequences that originated at 24 Hz or 30 Hz. It also means that for both the 720-line and 1080-line formats, the encoder will be designed to identify and take advantage of the lesser pixel rate in the film modes if the film mode material is presented to the encoder at 59.94 Hz or 60.0 Hz rate, using a pull-down technique.

Initial prototypes of the HDTV equipment will not contain interfaces for direct connection to 24 Hz or 30 Hz material, although the encoder architecture can support such interfaces when they become useful.

2.2.2. Number of fields per frame

The 1080-line format with 59.94 Hz or 60.0 Hz rate will be interlaced 2:1, yielding a frame rate of 29.97 Hz or 30.0 Hz. All the other format variations will use progressive scanning.

2.2.3. Number of active lines per frame

The number of active lines per frame will be either 720 active lines or 1080 active lines.

2.2.4. Total number of lines per frame

The total line count in the prototype for the 720-line format will be 787.5 lines per frame.

The total line count in the prototype for the 1080-line format will be 1125 lines per frame.

2.2.5 Number of active video samples per line

The number of active samples per line for the 720-line format will be 1280 samples of pixels.

For the 1080-line format, the progressive-scan variations with 24 Hz or 30 Hz frame rates will include 1920 active video samples per line. For the 59.94 Hz and 60.0 Hz interlaced variations of the 1080-line format, the number of active samples per line will be 1920 for the input interface.

The prototype encoder will be designed to support both 1440 and 1920 active samples per line for encoding and transmission, for the interlaced format only. The encoded data stream will include a signal that indicates the number of samples per line.

2.2.6 Total number of video samples per line

The total number of video samples per line for the 720-line format used in the prototype will be 1600, for all variations.

For the 1800-line format in the prototype, each line will represent 2200 cycles of the pixel rate clock, or 2200 sample times.

The Grand Alliance prototype also incorporates a variation of the 1080-line format with 1440 active samples per line.

2.2.7 Sampl ing frequency for video samples

For the 720-line format, the pixel rate clock used in the prototype will be 75,600,000 Hz for the 60.0 Hz frame rate, and approximately 75,524,476 Hz for the 59.94 Hz frame rate (which is 1000/1001 times the 60.0 Hz rate).

The clock rates used in the prototype for the 1080-line format will be 74,250,000 Hz for the 60.0 Hz interlaced variation, and approximately 74,175,824 Hz for the 59.94 Hz interlaced variation, at the input interface to the encoder.

Note that the ratio of clock rates for the specified 60 Hz inputs is 56:55 (75.6 MHz: 74.25 MHz).

2.2.8 Allowable combinations of parameters

Internally, the encoder for the 720-line format uses three frame rates, to allow the encoder to take advantage of film sequence redundancies. For the input to the encoder, a single format at either 59.94 Hz or 60.0 Hz will be interfaced to the encoder for the initial system.

The 1080-line format will have an input interface with 1920 active samples per line. As previously indicated, the Grand Alliance may reduce the number of samples per line for compression, for the interlaced format only, if necessary to preserve image quality. In that case, the number of samples encoded would be 1440 samples per line, for the interlaced variation at 59.94 or 60.0 Hz field rates. The progressive scan 24 Hz and 30 Hz variations with 1080 active lines will use 1920 active samples per line.

2.3 Summary Table for formats

The table attached shows the allowable variations. The table indicates entries for 59.94 Hz and 60.0 Hz field/frame rates, although these should not be considered distinct formats, merely frame rate implementation options.

GRAND ALLIANCE VIDEO SCANNING FORMATS

Encoder Format for Transmission

Active Lines	Active Pixels/Lines	Rate (Hz)	Pro/Int
720	1280	59.94/60.0	1:1
	1280	23.98/24.0	1:1
	1280	29.97/30.0	1:1
1080	1920/1440	59.94/60.0	2:1
	1920	23.98/24.0	1:1
	1920	29.97/30.0	1:1

Chapter 3

VIDEO COMPRESSION SYSTEM

3.1 Overview

The Grand Alliance (GA) HDTV system is an all-digital system designed to transmit high quality video and audio over a single 6 MHz terrestrial channel. A major challenge in the system design is video compression. This report summarizes the current status of the video compression subsystem of the GA HDTV system. The Grand Alliance compression specialist group is in the process of further evaluating the video compression subsystem and additional improvements may be added in the future.

Modern digital transmission technologies deliver approximately 17 Mbps to about 20 Mbps to encode video data within a single 6 MHz terrestrial channel. This means that encoding the HDTV video source whose resolution is typically six times that of the NTSC resolution requires a bit rate reduction by a factor of 50 or higher. To achieve this bit rate reduction, the GA HDTV system is designed to be very efficient in utilizing available channel capacity by exploiting modern, sophisticated video compression technology. The methods utilized, for example, include source adaptive processing, motion estimation/compensation, transform representation, and statistical coding. The GA system has been designed to incorporate most key video compression technology utilized in previously tested systems. The GA system has also been designed to perform well with progressive as well as interlaced pictures. Due to efficient channel utilization, the GA video compression system is an excellent choice, not only for terrestrial broadcasting, but for cable and satellite broadcasting.

In designing the GA video compression system, we have emphasized compatibility with MPEG (Moving Picture Experts Group) syntax. There are several reasons behind the emphasis on MPEG compatibility. The MPEG standard was developed utilizing modern sophisticated video compression methods and is very efficient in video compression. In addition, the MPEG standard was established by cooperative efforts of a number of organizations throughout the world. By placing special emphasis on compatibility with MPEG syntax, we have designed a video compression system which can serve as the HDTV standard not only for North America, but also for world-wide use.

We believe that the video compression subsystem of the GA HDTV system can deliver excellent HDTV quality video within a single 6 MHz terrestrial broadcast channel in North America. Furthermore, we believe that the system is an excellent choice for world-wide HDTV delivery for cable and satellite channels.

3.2 Basics Of Video Compression

A major objective of video compression is to represent a video source with as few bits as possible while preserving the level of quality required for the given application. Video compression has two major applications. One is efficient utilization of channel bandwidth for video transmission systems. The other is reduction of storage requirements. Both of these applications apply to the HDTV system.

Video compression for HDTV application is shown in Figure 3.2-1. The video source is encoded by the video encoder. The output of the video encoder is a string of bits that represents the video source. The channel coder transforms the string of bits to a form suitable for transmission over a channel through some form of modulation. The modulated signal is then transmitted over a communication channel. The communication channel typically introduces some noise, and provision for error correction is made in the channel coder to compensate for this channel noise. At the receiver, the received signal is demodulated and transformed back into a string of bits by a channel decoder. The video decoder reconstructs the video from the string of bits for human viewing. This report focuses on video compression, namely video encoding and decoding.

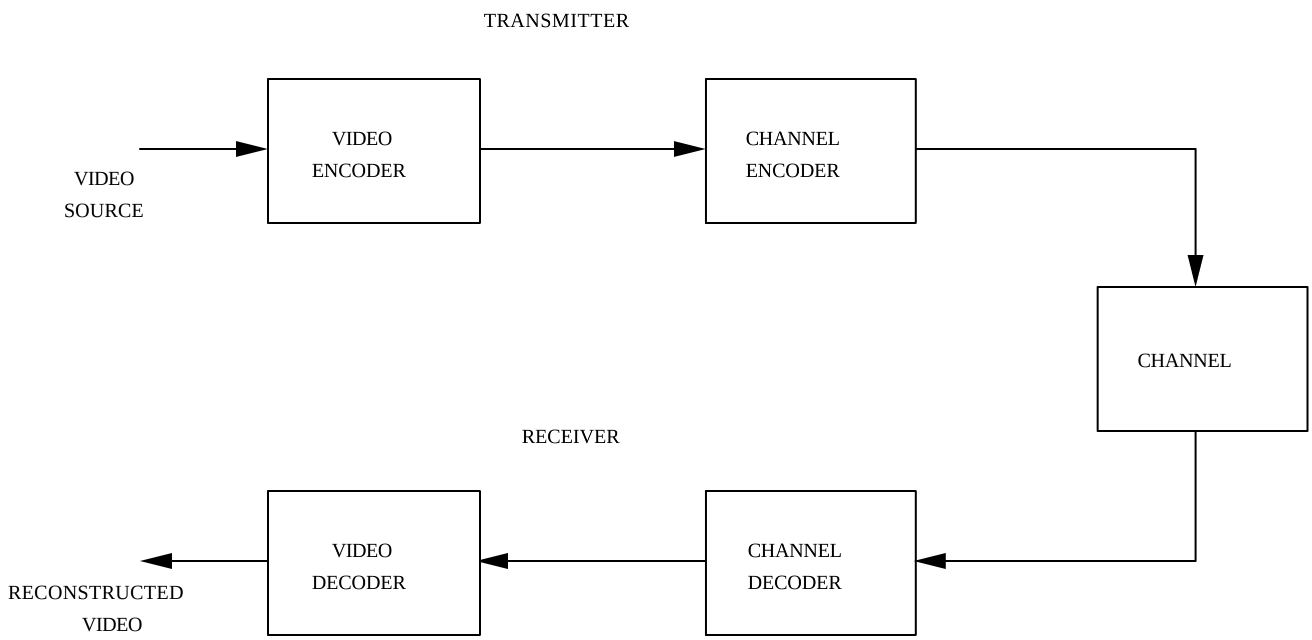


Figure 3.2-1. Video Compression for HDTV Application

The video compression system consists of three basic operations, as shown in Figure 3.2-2. In the first stage, the video signal is expressed in a more efficient representation which facilitates the process of compression. In essence, the representation determines what specifically is coded. The representation may contain more pieces of information to describe the signal than the signal itself, but most of the important information will be concentrated in only a small fraction of this description. In an efficient representation, only this small fraction of the data needs to be transmitted for an appropriate reconstruction of the video signal. The second operation, quantization, performs the discretization of the representation information. To transmit video over a digital channel, the representation information is quantized to a finite number of levels. The third operation is assignment of codewords, which are the strings of bits used to represent the quantization levels.

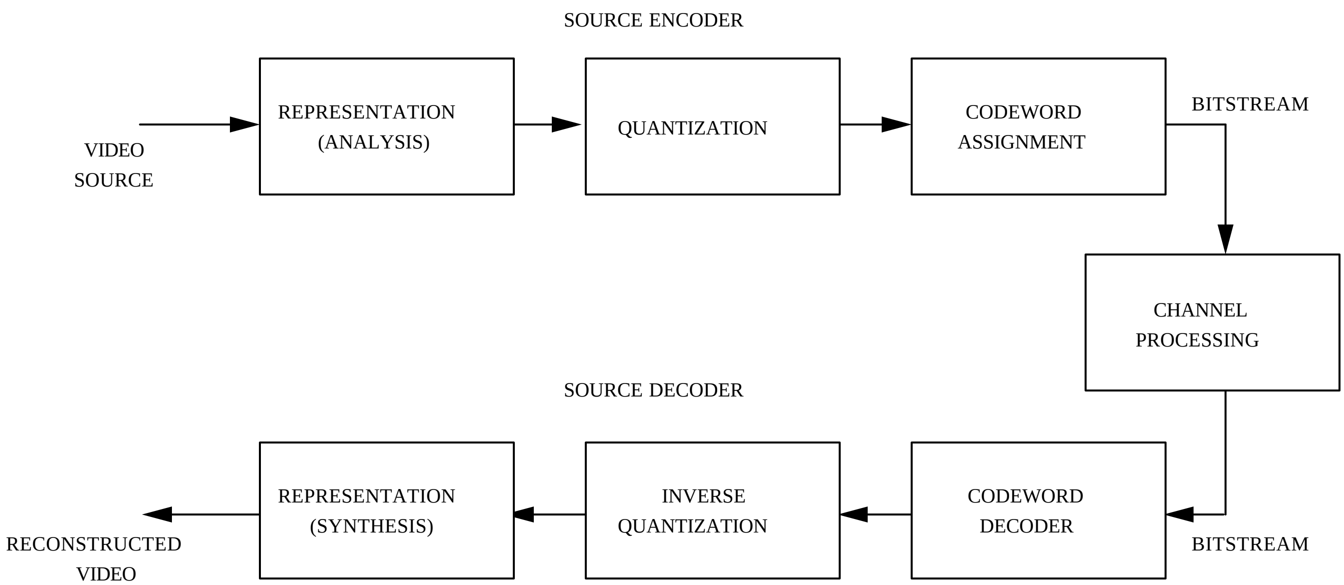


Figure 3.2-2. Overview of Video Compression System

Each of the three operations attempts to exploit the redundancy present in the video source and the limitations of the human visual system. By removing the redundancy present in the video source, the same or related information does not have to be transmitted repeatedly. By exploiting the limitations of the human visual system, the information in the video source that is not utilized by the human visual system does not need to be transmitted. In the following sections, we discuss the methods used in the GA video compression system.

3.2.1 Representation

To remove the redundancy in the signal and to eliminate the information irrelevant to the human viewer, the GA system utilizes many techniques discussed in this section.

3.2.1.1 Source-Adaptive Processing

A given television system must provide an interface to many kinds of video source formats. These include video from a video camera, the various film standards, magnetic and optical media, and synthetic imagery. In the GA system, we exploit the differences that may exist among the various sources to improve the performance of the video compression system.

3.2.1.2 Color-Space Processing

The input video source to the GA video compression system is in the form of R (red), G (green), and B (blue) components. The RGB components are highly correlated with each other. Furthermore, the human visual system responds differently to the luminance and chrominance components. To reduce the correlation and exploit this difference in the human visual system, the RGB components are converted to the YC1C2 color space through a linear transformation. Y corresponds to the luminance (intensity or black-and-white picture) while C1 and C2 correspond to the chrominance.

In the YC1C2 color space, most of the high frequency components are concentrated in the Y component. Furthermore, the human visual system is less sensitive to high frequency components of the chrominance components than of the luminance component. To exploit these characteristics, in the GA video compression system the chrominance components are low-pass filtered and sub-sampled by a factor of two along both the horizontal and vertical dimensions, producing chrominance components that are one-fourth the spatial resolution of the luminance component.

3.2.1.3 Motion Estimation/Compensation

A video sequence is a series of still images shown in rapid succession to give the impression of continuous motion. Even though each of the frames is distinct, the high frame rate necessary to achieve proper motion rendition usually results in much temporal redundancy among the adjacent frames. Motion compensation attempts to remove this temporal redundancy.

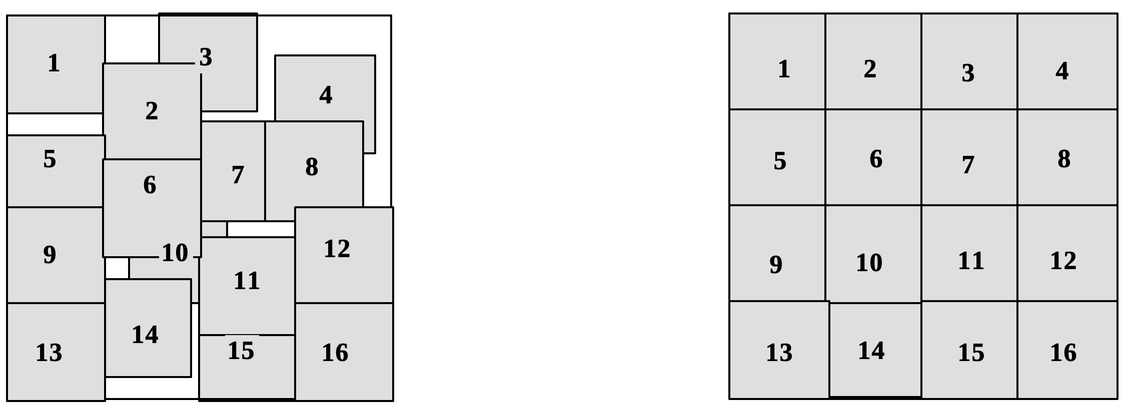
Much of the variation in intensity from one frame to the next is due to object motion. In motion-compensated coding, the current frame is predicted from the previously encoded frame by estimating the motion between the two frames and compensating for the motion. The difference between the current frame and the prediction of the current frame is called the motion- compensated residual and this residual is encoded. For a typical video sequence, the energy in the residual is much less than that in the original video due to the removal of the temporal redundancy. Encoding the residual rather than the video itself ensures that the temporally redundant information is not encoded repeatedly. In motion estimation, the same imagery is assumed to appear in consecutive video frames, although possibly at different locations. The motion may be global or local within the frame. To optimize the performance, an estimate of the motion is computed for each local region within a frame. The most common model for the local motion is simple translational motion. This model is highly restrictive, and cannot represent the large number of possible motions, such as rotations, scale changes, and other complex motions. Nevertheless, by assuming these motions only locally and by identifying and processing those regions separately where the model fails, excellent performance can be achieved.

One approach for motion estimation, which is very popular and is utilized in the GA video compression system, is based on block matching. In this approach, the current frame is partitioned into rectangular regions or blocks, and a search is performed for the displacement which produces the "best match" among possible blocks in an adjacent frame. Block matching is popular, because it achieves high performance and exhibits a simple, periodic structure which allows straightforward VLSI implementation. The process of motion-compensated prediction used in the GA system is illustrated in Figure 2-3. The frame to be encoded is partitioned into blocks and each block is predicted by displacing a block from the reference frame. The difference between the current frame and its prediction is the motion-compensated residual.

3.2.1.4 Intra-frame and Inter-frame Encoding

Motion-compensated coding is an example of inter-frame encoding. In some cases, it is useful to encode the video itself (intra-frame coding) without motion compensation. The GA video compression system utilizes both intra-frame coding and inter-frame coding.

Motion compensation is a form of predictive coding applied along the temporal dimension. For typical video frames, prediction by motion compensation is quite good. For some frames such as those at scene changes, however, a prediction is not as good. Even within a given frame, some regions may be predicted well with motion compensation while other regions may not be predicted very well using motion compensation based on translational motion. In some cases, therefore, the system may perform worse with predictive coding than by simply coding the frame itself. The GA video compression system utilizes inter-frame encoding when the motion- compensated prediction is quite good, while utilizing intra-frame encoding otherwise.



BLOCKS OF PREVIOUS FRAME USED TO PREDICT NEXT FRAME.

PREVIOUS FRAME AFTER USING MOTION VECTORS TO ADJUST BLOCK POSITIONS.

Figure 3.2-3. Motion-Compensated Prediction

The intra-frame coding mode is also utilized in the GA system for receiver initializations and channel acquisition (when the receiver is turned on or the channel is changed). It is also used when non-correctable channel errors occur. With the motion-compensated prediction, an initial frame must be available at the decoder to start the prediction loop. Therefore, a mechanism must be built into the system so that if the decoder loses synchronization for any reason, it can rapidly reacquire tracking. One approach used is periodic intra-frame encoding (refresh). In this approach, the entire frame is encoded using the intra-frame mode (I frames). Alternatively, a part of the frame is encoded using the intra-frame mode, and by encoding different parts of the frame in different frames, the entire frame is refreshed (progressive refresh). This refresh approach produces a periodic re-initialization of the temporal prediction which allows the decoder to re- acquire tracking. The GA system utilizes both progressive refresh and complete-frame refresh through the use of I-frames.

3.2.1.5 Discrete Cosine Transform

Motion compensation reduces the temporal redundancy of the video signal, but there still remains spatial redundancy in the motion-compensated residual (MC-residual). This is especially true if no MC processing is performed and the original frame is to be encoded. For simplicity, the term residual will be applied to whatever frame of data is to be spatially processed, irrespective of whether MC has been applied or not. To reduce the spatial redundancy of the residual, the Discrete Cosine Transform is used.

The Discrete Cosine Transform (DCT) compacts most of the energy of the residual into only a small fraction of the transform coefficients. The coding and transmission of only these energetic coefficients can result in the reconstruction of high quality video. The DCT was chosen in the GA system because it has good energy compaction properties, results in real coefficients, and there exist numerous fast computational algorithms for its implementation. The DCT of a two-dimensional signal may be computed by applying the one-dimensional DCT separably first to the rows and then to the columns of the signal.

The size of the DCT might be chosen as the entire frame, but much better performance can be achieved by subdividing the residual into many smaller regions each of which is individually processed. The motivation for this is very easy to see and it is one of the most important aspects of a high quality video encoder. If we compute the DCT of the entire residual frame, we treat the whole residual frame equally. For a typical image, some regions contain a large amount of detail and other regions contain very little detail. By exploiting the changing characteristics of different images and of different portions of the same image, significant improvements in performance can be realized. In order to take advantage of the varying characteristics of the residual over its spatial extent, the residual is partitioned into 8x8 blocks. The blocks are then independently transformed and adaptively processed based on their local characteristics. The partitioning of the residual into small blocks before taking the transform not only allows spatially adaptive processing, but also reduces the computational and memory requirements. Partitioning the signal into small blocks before computing the DCT is referred to as the Block DCT.

The DCT tends to reduce the spatial correlation of the residual. This makes the DCT domain an appropriate representation since the DCT coefficients tend to have less redundancy. In addition, the DCT coefficients contain information about the spatial frequency content of the residual. By exploiting the spatial frequency properties of the human visual system, the DCT coefficients can be encoded to match the human visual system so that only the perceptually important DCT coefficients are encoded and transmitted.

3.2.2 Quantization

Through the processing discussed up to this point, an elegant representation in the form of the motion field, spatial frequency coefficients, and luminance/chrominance components has been created; however, no compression has been achieved. In fact, an expansion of data has resulted, since there are currently more pieces of information used to describe the video than before. However, most of the perceptually important information has been compressed into only a fraction of these "pieces of information", and this data can be selected and encoded for transmission.

The goal of video compression in the HDTV application is to maximize the video quality at a given bit rate. This requires a wise distribution of the limited number of available bits. By exploiting the statistical redundancy and perceptual irrelevancy within the new representation, an appropriate bit allocation can yield significant improvements in performance. Quantization is performed to discretize the values, and through quantization and codeword assignment, the actual bit rate compression is achieved. The quantization process can be made the only loss step in the compression algorithm. This is very important, as it simplifies the design process and facilitates fine tuning of the system. Quantization may be applied to elements individually (scalar quantization) or to a group of elements simultaneously (vector quantization).

In scalar quantization (SQ), each element may be quantized with a uniform (linear) or nonuniform (nonlinear) quantizer. The quantizer may also include a dead zone (enlarged interval around zero) to quantize or core to zero small, noise-like perturbations of the element value. The close relationship between quantization and codeword assignment suggests that separate optimization of each may not necessarily yield the optimum performance. On the other hand, joint optimization of quantization and codeword assignment is a highly nonlinear and complex process. Experiments have shown that a linear quantizer with an appropriate step-size individually chosen for each element to be quantized, followed by proper entropy coding, may yield close to optimum performance. This is the approach taken in the GA system.

When quantizing transform coefficients, the differing perceptual importance of the various coefficients can be exploited by "allocating the bits" to shape the quantization noise into the perceptually less important areas. This can be accomplished by varying the relative step-sizes of the quantizers for the different coefficients. The perceptually important coefficients may be quantized with a finer step-size than the others. For example, low spatial frequency coefficients may be quantized finely, while the less important high frequency coefficients may be quantized more coarsely. A simple method to achieve different step-sizes is to normalize or weight each coefficient based on its visual importance. All of the normalized coefficients may then be quantized in the same manner, such as rounding to the nearest integer (uniform quantization). Normalization or weighting effectively scales the quantizer from one coefficient to another. The GA video compression system utilizes perceptual weighting, where the different DCT coefficients are weighted according to a perceptual criterion prior to uniform quantization. The perceptual weighting is determined by quantization matrices and the GA video compression system uses adaptive quantization matrices.

In video compression, most of the transform coefficients are quantized to zero. There may be a few non-zero low-frequency coefficients and a sparse scattering of non-zero high-frequency coefficients, but the great majority of coefficients will have been quantized to zero. To exploit this phenomenon in the GA system, the two-dimensional array of transform coefficients is reformatted and prioritized into a one-dimensional sequence through a zigzag scanning. This results in most of the important non-zero coefficients (in terms of energy and visual perception) being grouped together early in the sequence. They will be followed by long runs of coefficients that are quantized to zero. These zero-valued coefficients can be efficiently represented through run length encoding. In run length encoding, the number (run) of consecutive zero coefficients before a non-zero coefficient is encoded, followed by the non-zero coefficient value. The run length and the coefficient value can be entropy coded, either separately or jointly. The scanning separates most of the zero and the non-zero coefficients into groups, thereby enhancing the efficiency of the run length encoding process. Also, a special End Of Block (EOB) marker is used to signify when all of the remaining coefficients in the sequence are equal to zero. This approach is extremely efficient, yielding a significant degree of compression.

3.2.3 Codeword Assignment

Quantization creates an efficient discrete representation for the data to be transmitted. Codeword assignment takes the quantized values and produces a digital bit stream for transmission. The quantized values can be simply represented using uniform or fixed length codewords. Using this approach, every quantized value will be represented with the same number of bits. Greater efficiency, in terms of bit rate, can be achieved by employing entropy coding. Entropy coding attempts to exploit the statistical properties of the signal to be encoded. A signal, whether it is a pixel value or a transform coefficient, has a certain amount of information, or entropy, based on the probability of the different possible values or events occurring. For example, an event that occurs infrequently conveys much more new information than one that occurs often. By realizing that some events occur more frequently than others, the average bit rate may be reduced.

There are two important issues that arise when considering the application of entropy coding. First, entropy coding involves increased complexity and memory requirements over fixed length codes. Second, entropy coding coupled with the non-stationarity of the video signal results in a time-varying bit rate. (Other aspects of the source coding may also raise this issue). A buffer control mechanism is necessary when the variable bit rate source coder is to be coupled with a constant bit rate channel.

In the GA system, an entropy coder is used to reduce the statistical redundancy inherent in the elements encoded for transmission. The primary redundancy is the nonuniform probability distribution over the possible range of each element. The more the probability distribution deviates from a uniform distribution, the greater improvement can be achieved via entropy coding. Other sources of statistical redundancy which may exist include the statistical dependence among the encoded elements.

Huffman coding which is utilized in the GA system is one of the most common entropy coding schemes. In Huffman coding, a code book is generated which minimizes the entropy subject to the codeword constraints of integer lengths and unique decodability. Events which are more likely to occur will be assigned shorter length codewords while those which are less likely to occur will be assigned longer length codewords. Huffman coding results in a variable bit rate per event; more importantly, the average bit rate is reduced. The training or generation of the code book is achieved by using a representative set of data to estimate the probability of each event. Optimal performance can be achieved by designing an individual code book for each element to be encoded. However, this results in a large number of code books. Close to optimal performance is achieved by using a new code books where elements with similar statistics are grouped and encoded together. Similarly, the size of each code book can be reduced by grouping together very unlikely events into a single entry within the code book. When any event belonging to this group occurs, the codeword for this group is transmitted followed by an exact description of the event.

Whenever entropy coding is employed, the bit rate produced by the encoder is variable and is a function of the video statistics. If the application requires a constant bit rate output, a buffer is necessary to couple the two. The buffering must be carefully designed. Random spikes in the bit rate can overflow the buffer while dips in the bit rate can produce an underflow. What is needed is some form of buffer control that would allow efficient allocation of bits to encode the video while ensuring that no overflow or underflow occurs.

The buffer control typically involves a feedback mechanism to the compression algorithm whereby the amplitude resolution (quantization) and/or spatial, temporal and color resolution may be varied in accordance with the instantaneous bit rate requirements. The goal is to keep the average bit rate constant and equal to the available channel rate. If the bit rate decreases significantly, a finer quantization can be performed to increase it. The buffer control mechanism is an essential part of any high-performance constant-output bit rate HDTV system. The GA system utilizes a highly sophisticated buffer control mechanism to deliver the highest video quality within a reasonable buffer size.

In this section, some of the basic video compression elements utilized on the GA video compression system have been discussed. Further details can be found in references 1 and 2.

3.3 Video Compression Approach

Figure 3.3-1 shows the functional block diagram of the GA video encoder. The components are described in the following sections.

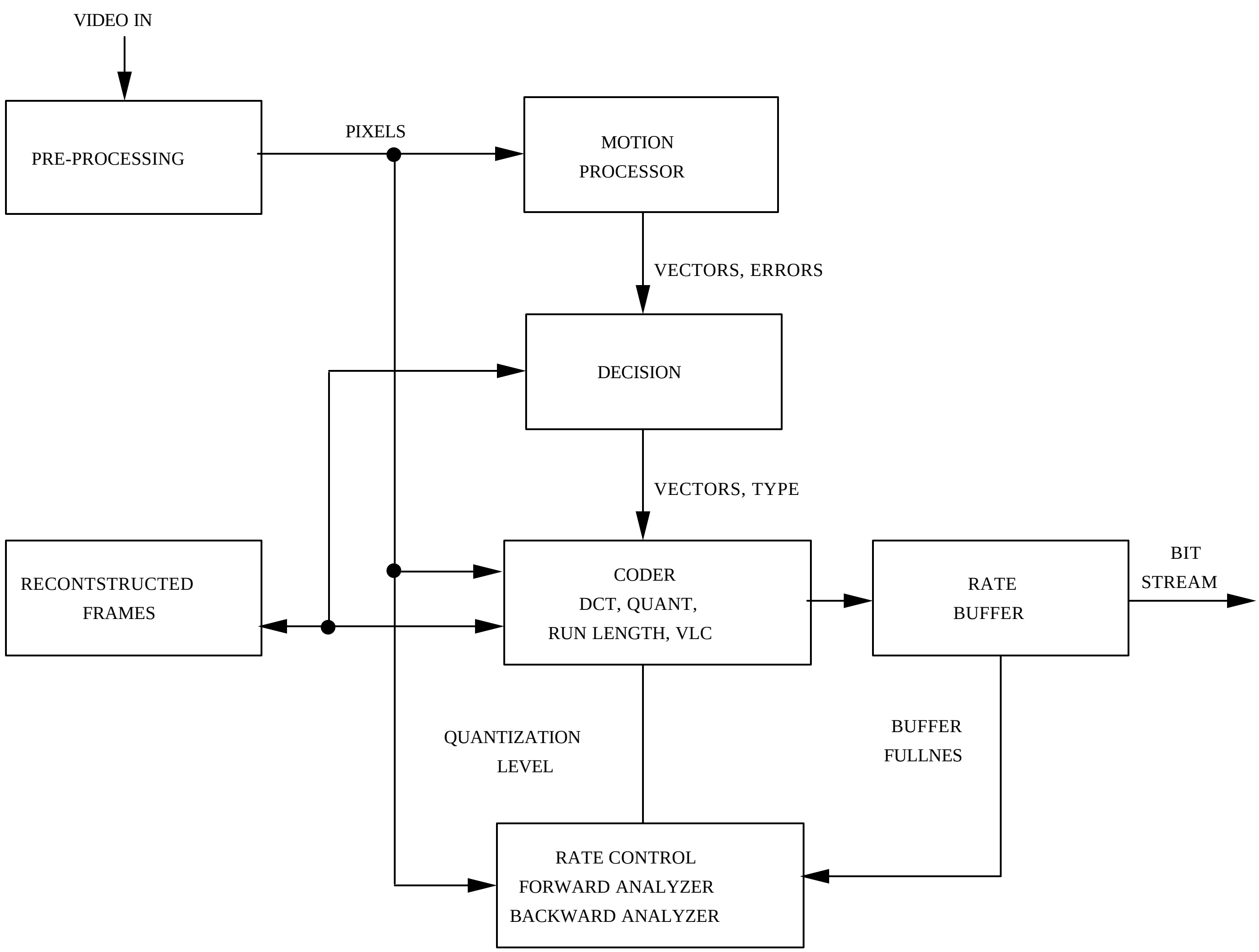


Figure 3.3-1. GA Video Encoder

3.3.1 Video Preprocessor

As shown in Figure 3.3-2, analog video is received in RGB format and digitized using 10-bit A/D converters. Gamma correction is then applied to each color component in order to compensate for the non-linear response of the camera. This helps to reduce the visibility of quantization noise, particularly in the dark regions of the image.

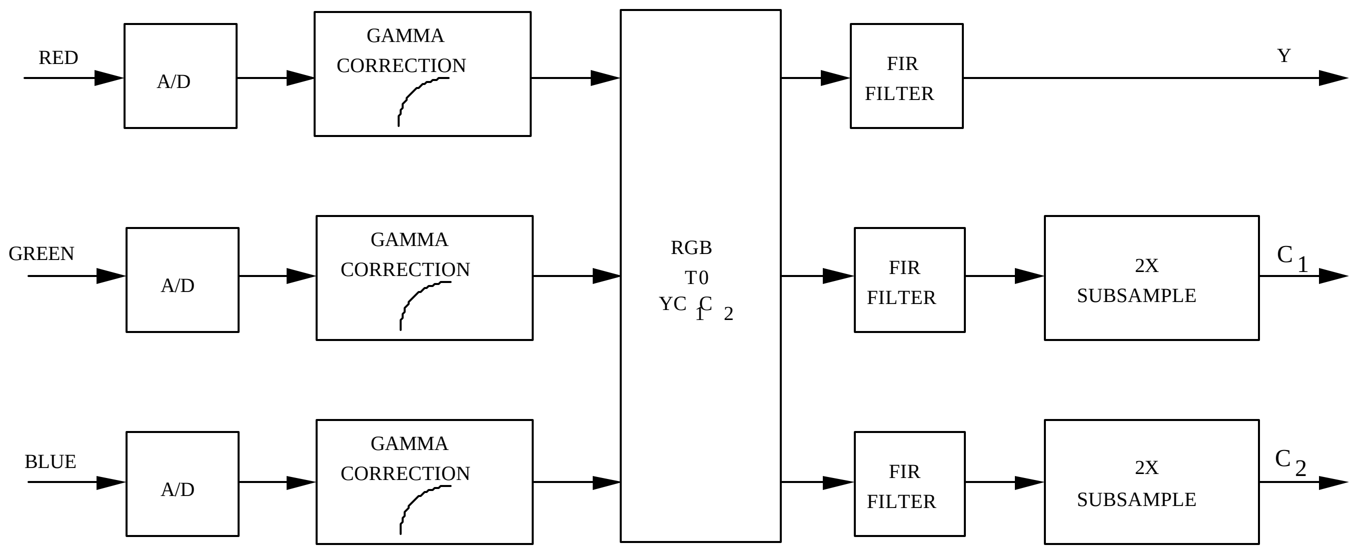


Figure 3.3-2. Video Preprocessor

The digitized and gamma-corrected RGB samples are then converted to the SMPTE 240M YC1C2 color space using a linear matrix transformation. Programmable FIR filters are used to shape the frequency response of each color component. The two chrominance components use half-band filters in order to prevent aliasing during the subsequent sub-sampling process. The sub-sampling is performed by a factor of 2, both horizontally and vertically.

3.3.2 Motion Estimation and Compensation

The GA compression algorithm utilizes multiple prediction methods to effectively provide motion compensation for progressive and interlaced pictures. The prediction methods include frame prediction, adaptive field prediction, dual prime (used for forward prediction only), and bi- directional prediction. Among these, the dual prime and the bi-directional prediction modes deserve further discussion. In the dual prime case the prediction is formed from pixels in previously displayed frames, while in the B-frames case the prediction is based on both previous frames and future frames. The dual prime and bi-directional methods are complementary since the dual prime mode is for interlaced only and is precluded when a sequence uses B-frames.

3.3.2.1 The "P-Frame" Motion Vector Prediction Mode

P-frames are frames where the prediction is in the forward direction only (i.e., predictions are formed only from pixels in previously displayed frames). These forward-predicted frames allow the exploitation of inter-frame coding techniques to improve the overall compression efficiency and picture quality.

Figure 3.3-3 illustrates a time sequence of video frames consisting of intra-coded pictures, predictive coded pictures, and bi-directionally predictive coded pictures. P-frames are predicted using the most recently encoded P-frame of I-frame in the sequence. In Figure 3-3, P-frames occur every second frame except when an I-frame is used. Each macroblock within a P-frame can be either forward-predicted or intra-frame coded. If a macroblock is forward-predicted, then either frame-based or field-based prediction may be used. The decision is made by the encoder based on the smallest prediction error using each method. Dual prime prediction may also be chosen at this stage if the prediction error using dual prime is smallest. Regardless of whether field-based, frame-based or dual prime prediction is used, the predicted macroblock is subtracted from the original macroblock, and only the "difference" values are encoded and transmitted.

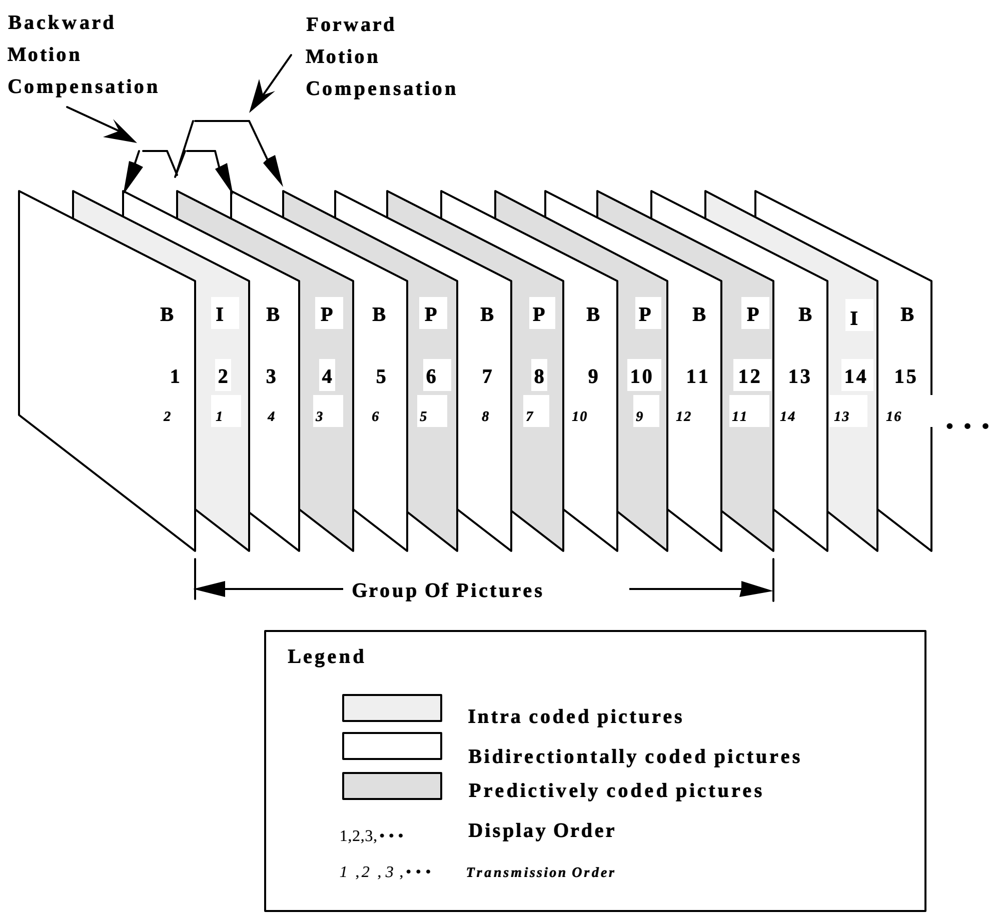


Figure 3.3-3 Example of a Coded Video Sequence Using P-Frames and B-Frames

3.3.2.2 Dual Prime

The dual prime prediction mode is an alternative "special" prediction mode which is based on field-based motion prediction but requires fewer transmitted motion vectors than conventional field-based prediction. This mode of prediction is available only for interlaced material and only when the encoder configuration does not use B-frames (M=1). For these reasons, this mode of prediction is particularly useful for improving encoder efficiency for low delay applications.

The basis of dual prime prediction is that field-based predictions of both fields in a macroblock (MB) are obtained by averaging two separate predictions which are predicted from the two nearest decoded fields (in time). Each of the MB fields is predicted separately, although the four vectors (one pair per field) used for prediction are all derived from a single transmitted field- based motion vector. In addition to the single field-based motion vector, a small "differential" vector (limited to vertical and horizontal component values of +1, 0 and -1, and represented by two 1-2 bit codes) is also transmitted for each MB. Together, these vectors are used to calculate the pairs of motion vectors for each MB field. the first prediction vector in the pair is simply the transmitted field-based motion vector. The second prediction vector is obtained by combining the differential vector with a scaled version of the first vector. Once both predictions are obtained, a single prediction for each macroblock field is calculated simply by averaging each pel from the two original predictions. The final averaged prediction is then subtracted from the macroblock field being encoded.

Figure 3.3-4 illustrates the relationship between the transmitted vectors (one field-based vector and one differential vector) and the prediction vectors for each of the fields in a macroblock. The two separate sets of vectors shown in Figure 3-4 correspond to the predictions for the two fields which make up the macroblock. The transmitted field-based motion vector and the transmitted differential vector (identical for each set) are represented by solid lines. The differential motion vector is the smaller vertical vector. The scaled vectors are represented by dotted lines. The second (calculated) field-based motion vectors are represented by the dashed lines.

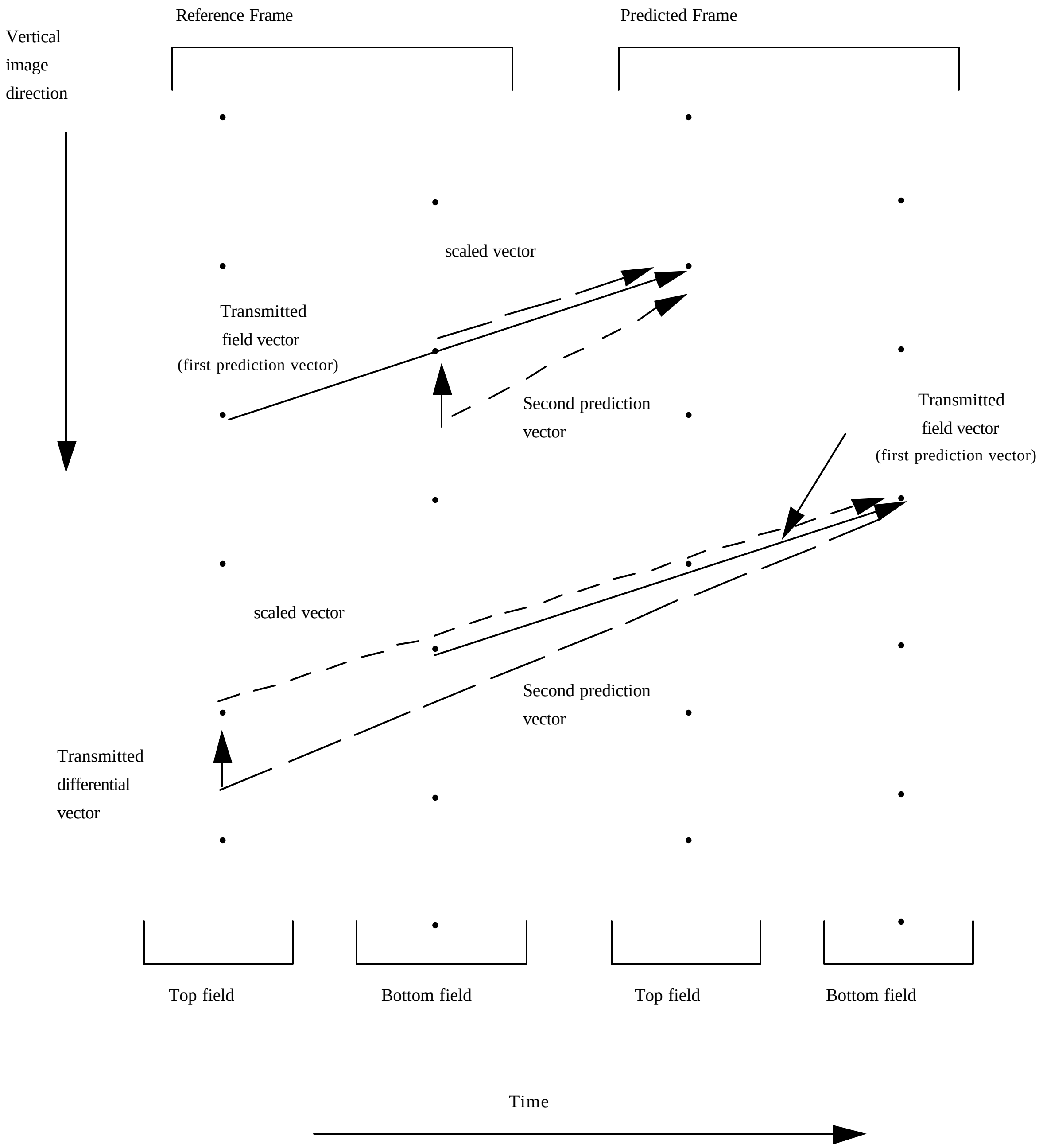


Figure 3.3-4. Dual Prime Prediction

3.3.2.3 The "B-Frame" Motion Vector Prediction Mode

The B-frame is a picture type within the coded video sequence which has some special prediction modes available to the encoder. It is useful for increasing the compression efficiency and perceived picture quality when encoding latency is not an important factor. It is also an attractive tool since it works equally well with both interlaced and progressive material. However, it has impact on the receiver cost since it requires additional memory. B-frames are included in the GA system because the increase in compression efficiency is noticeable especially with progressive scanning where techniques such as dual prime are not available. The improvement in performance was found to be sufficient to justify the cost of the additional memory required by the ATV receiver.

The basis of the B-frame prediction is that a video frame is correlated both with frames which occur in the past and the frames in the future. Consequently, if a future frame is available to the decoder, a superior prediction can be formed, thus saving bits and improving performance. Some of the consequences of using future frames in the prediction are: the B-frame cannot be used for predicting future frames, the transmission order of frames is different than the displayed order of frames, and the encoder and decoder must reorder the video frames, thereby increasing the total latency. In the example illustrated in Figure 3-3, there is one B-frame between each pair of I/P frames. Each frame is labeled with both its display order and transmission order. Although a B-frame is displayed between its adjacent I/P frames, it is transmitted out of sequence. This is so that the video decoder has the adjacent frames decoded and available for prediction. For a given macroblock within the B-frame, the encoder has four options for its prediction. They are: forward prediction, backward prediction, bi-directional prediction, and intra-frame coding. When bi-directional prediction is used, the forward and backward predictors are averaged and then subtracted from the target macroblock to form the prediction error. The prediction error is then transformed, quantized and transmitted in the usual manner.

3.3.3 Refreshing Options

A motion compensated prediction loop is not practical without some form of refreshing. This refreshing may be constant or variable and periodic. If a sequence of input images were perfectly predictable, the decoder would not receive any coefficient data and therefore could not reconstruct the picture after initialization. Decoder initialization occurs after the channel is changed, or the signal is lost and then recovered.

Consider the coding of a still image. The motion vectors and the prediction error would be zero. If the decoder were started with a sequence of zero prediction frames, then a blank or zero reconstructed frame would result. If the viewer changed to a channel transmitting a coded still image, the still image would never appear. A portion of the original picture must be spatially or temporally mixed with the prediction in the encoder to allow the decoder to synchronize to the encoder after decoder initialization.

The GA system satisfies the need for refreshing with two different mechanisms. I-frame refreshing uses periodic intra-coded frames. The advantages are that it provides clean insertion points in the compressed bit stream and nominal picture quality after acquisition (typically .5 seconds). The disadvantages are that it requires a larger bit buffer, it increases latency, and it complicates rate control.

Progressive refreshing uses periodic intra-coded (16x16) macroblocks. The advantages over I-frame refreshing are that it reduces the required buffer size, it simplifies rate control, and it reduces latency. One disadvantage is that motion vectors must be restricted in order to guarantee complete picture buildup after channel acquisition or channel errors.

3.3.4 Adaptive Field/Frame Processing

There are two options when processing interlaced video signals. The first option is to separate each frame into its two field components and then process the two fields independently (Figure 3.3-5). The second option is to process the two fields as a single frame by interleaving the lines of field 1 and field 2 (Figure 3.3-6).

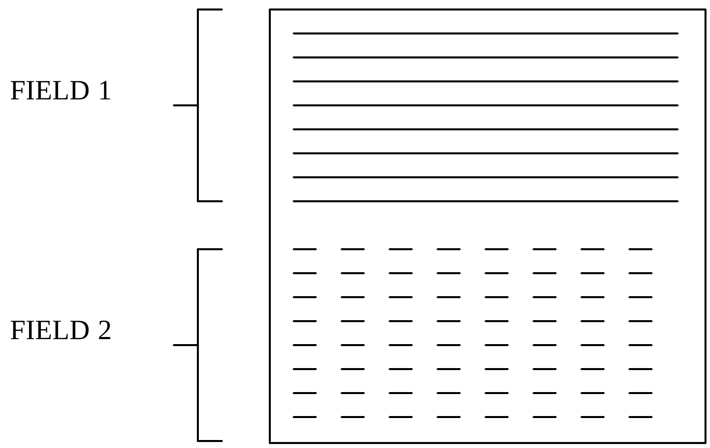


Figure 3.3-5. Field Processing

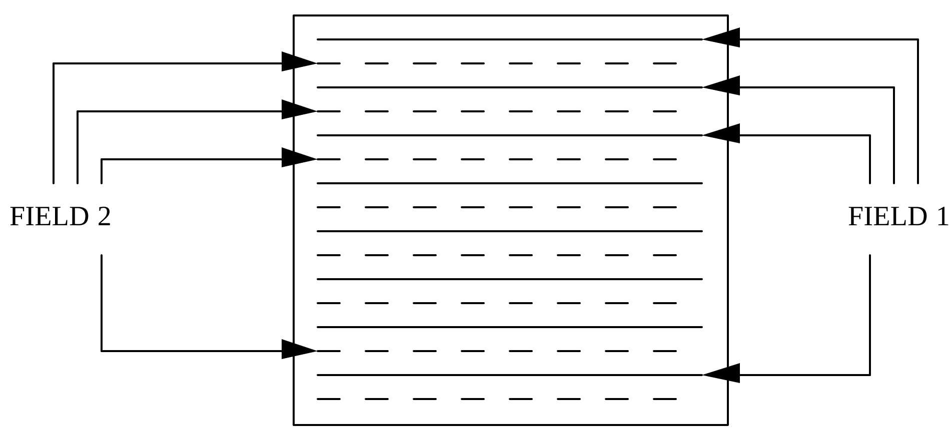


Figure 3.3-6. Frame Processing

Frame processing works better than field processing when there is little or no motion. Since each frame has twice as many lines or samples for a given picture height, there will be more correlation between samples and hence compressibility will be increased. Therefore, to achieve the same accuracy, field processing will require a higher bit rate, or alternatively, for equal bit rates, frame processing will achieve greater accuracy.

Similar advantages over field processing will be realized if horizontally moving features have little horizontal detail or if vertically moving features have little vertical detail. In other regions, where there is little detail of any sort, frame processing may still work better than field processing, no matter how rapidly changes occur.

Field processing generally works better than frame processing in detailed moving areas. In such cases, the interleaving of the first and second fields would introduce spurious high vertical frequencies if frame processing were used. This would reduce the correlation between lines and therefore the effectiveness of the compression process.

The GA system combines the advantages of both frame processing and field processing by adaptively selecting one of the two modes on a block-by-block basis. The mode which is selected is the one which maximizes correlation between adjacent lines within the block. Since the decision is made on a local basis, the system can adjust to scenes containing both moving and non-moving features, and therefore accurately reproduce vertical details in non-moving areas, and deliver good motion rendition in others.

The adaptive field/frame feature is only enabled when processing interlaced source material. Non-interlaced sources are always processed using frame mode.

3.3.5 Forward Estimation for Rate Prediction

Motion compensation, adaptive quantization and variable length coding produce highly variable amounts of compressed video data as a function of time. For example, the compressed bit-rate after a scene change can be several times greater than the bit-rate of the channel. Therefore, compressed video data buffering in the encoder and decoder is required for efficient channel utilization.

Buffer size is constrained by the maximum tolerable delay through the system and by cost. The fullness of the buffer is controlled by adjusting the amount of distortion or quantization error in each image. In the encoder, the buffer will fill more quickly if the distortion is low. A feedback control system is required to regulate the distortion level which controls the buffer fullness to prevent overflow. The design of this control system is complicated by:

- Delay between a change in the distortion level and the subsequent change in buffer fullness.

- Perceptual constraints on the instantaneous and averag distortion levels.

Difficulties in modeling the bit-rate as a function of distortion level.

The visibility of distortion due to an increase in scene complexity is minimized by smoothly increasing the distortion level. This is facilitated by accurately modeling the rate versus distortion level since an accurate model allows the desired buffer fullness to be achieved for each frame. The model allows an estimate of complexity to be translated into a bit-rate for a given frame.

In the GA rate control system, the complexity estimate is the variance of the ideal (unquantized) displaced frame difference (DFD). Since the whole-pel motion vectors are available one frame ahead of the loop, the ideal displaced frame difference is calculated using whole-pel accuracy (i.e., no sub-pel estimation). The pixel values of the ideal DFD are squared and summed over the picture.

3.3.6 Adaptive Intra and Non-Intra Quantization Matrices

The MPEG-2 syntax allows the quantization matrices to be specified for every picture for improved coding efficiency. A certain probability distribution is associated with the VLC codes for quantized coefficients in the MPEG standard. Although one cannot change the VLC distribution to match the actual distribution of the data, the quantization matrices can be adjusted to help match the distribution of the data to the distribution of the VLC. Over the course of encoding the frame data, the variance of each spatial frequency band is calculated for both Intra and Non-Intra data. Applying upper and lower bounds per band to ensure reasonable operation in all cases, the desired quantization matrix can be derived.

Transmitting the quantizer matrices costs bits in the compressed data stream; if sent every picture in the 60 Hz progressive mode, the matrices consume 0.1% of the channel bandwidth. This modest amount of overhead is reduced by updating the quantization matrix at the start of each GOP, at least every 400 ms (for start-up), or when the difference between the desired quantizer matrix and the prevailing quantizer matrix becomes significant.

Sufficient compression cannot be achieved unless a large fraction of the DCT coefficients are dropped and therefore not selected for transmission. The coefficients which are not selected are assumed to have zero value in the decoder. The GA system encodes the selections and runs of zeros following a zigzag pattern through the array of frequency ordered coefficients. The DC coefficients are coded differentially to take advantage of high spatial correlation.

3.3.7 Perceptual Weighting and Coefficient Selection by Perceptual Sensitivity

The human visual system is not uniformly sensitive to coefficient quantization error. Perceptual weighting of each source of coefficient quantization error is used to increase quantization error in order to lower the bit-rate. The amount of visible distortion resulting from quantization error for a given coefficient depends on the coefficient number or frequency, the local brightness in the original image and the duration or the temporal characteristic of the error. DC coefficient error results in mean value distortion for the corresponding block of pixels which can expose block boundaries. This is more visible than higher frequency coefficient error which appears as noise or texture.

Displays and human visual systems exhibit non-uniform sensitivity to detail as a function of local average brightness. Loss of detail in dark areas of the picture is not as visible as it is in brighter areas. Another opportunity for bit savings is presented in textured areas of the picture where high frequency coefficient error is much less visible than in relatively flat areas. Brightness and texture weighting require analysis of the original image since these areas may be well-predicted in the DFD. Finally, distortion is easily masked by limiting its duration to one or two frames. This effect is most profitably used after scene changes where the first frame or two can be greatly distorted without perceptible distortion at normal speed.

Traditionally, a given coefficient is transmitted whenever its quantized level is non-zero. By selecting some non-zero coefficients for elimination, a fine level of quantization can be achieved on the remaining coefficients, potentially improving the overall picture. Perceptual selection is such a method.

Perceptual Selection is used in the GA system using the properties of the human visual system to code pictures using fewer bits within a perceptually consistent level of quality. For each transform block, the perceptual selection method determines the acceptable amount of distortion per each frequency band; if the magnitude of the coefficient in the loop is smaller than the acceptable distortion level, then the coefficient is set to zero, regardless of the quantization step size level.

3.3.8 Adapting M and N

The motion estimator range of the GA prototype system is ±32 V x ±128 H between two successive frames. The use of B-frames reduces the effective motion tracking range by a factor of M where M-1 is the number of B-frames between a given pair of I- or P-frames.

Scenes with rapid large area motion may exceed the range of the motion estimator if M is too high. This condition is detected in the forward analyzer which then signals the frame reorder section and motion estimator to use a lower M value for subsequent frames. The increased bit- rate caused by the temporary motion estimator overrun is absorbed by the rate buffer to maintain picture quality.

3.3.9 Film Mode

The GA system is able to detect source material originating from 24 fps film that has been converted to 60 fps using the 3:2 pulldown process. As shown in Figure 3.3-7, some fields (or some frames) are known to be identical due to the 3:2 pull down method. The GA system detects this redundancy through utilization of a film detection method that looks for the 3:2 pull down pattern. In the future, when the film mode is detected, the GA system removes the redundancy and converts it back to the original 24 fps source format prior to video encoding. The removal of redundancy ensures that the same information is not transmitted repeatedly, thus significantly improving the efficiency of the video compression system.

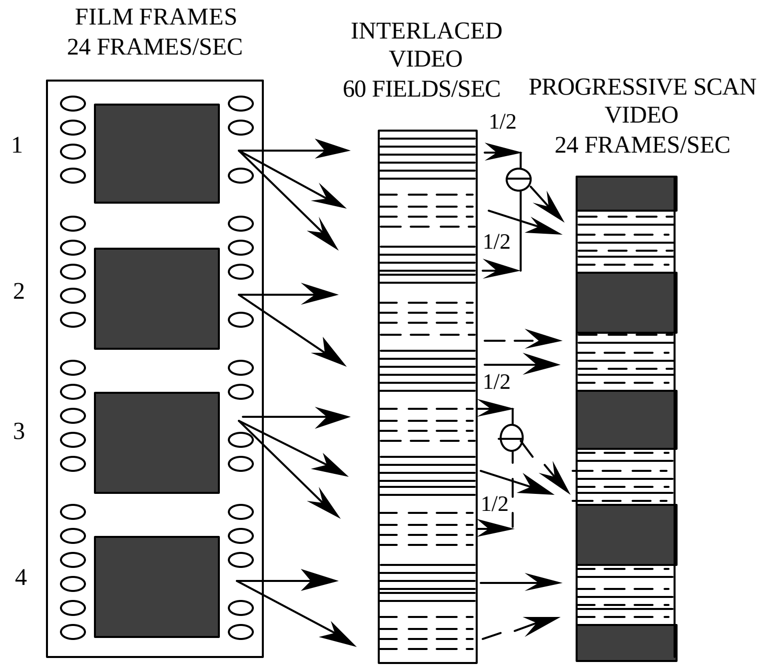


Figure 3.3-7. Conversion of Film to Interlaced Video and Restoration of 24 fps Progressive Scan

The GA system is also able to detect 30 fps material that has been converted to a 60 fps progressive scan format by simple frame repetition. When this conversion is detected, the encoder will discard the redundant frame and indicate to the decoder that each frame is to be displayed twice.

3.4. Inter-Operability

The GA compression algorithm is based on the emerging MPEG-2 compression standard. Consequently, the HDTV decoders will have resident decoder functionality to implement a device with a high degree of inter-operability. This inter-operability is a manufacturers option, and is not mandated.

3.4.1 Inter-Operability with MPEG

The opportunities for inter-working are with MPEG-2 at High and High 1440 Level, CATV, DBS, and Teleco Standard Definition MPEG-2 and MPEG-1 services, and inter-working with MPEG-1 services such as those expected to be implemented for computer multimedia terminals. In each case, the HDTV decoder has the required decode engine and frame memory to decode all of these services. The implementation of image scaling for display, or multi-scan displays is left to the manufacturer.

The inter-operability choices with MPEG-2 are shown in Figure 3.4-1. Each line illustrates a potential inter-working scenario. The lines originate at the encoding side, with the arrow pointing to the decoder. A solid line indicates that inter-operability is achieved, while a dashed line indicates that the system is not inter-operable unless the manufacturer adds functionality to the decoding device. The lines labeled A and E show a US HDTV decoder inter-working with a MPEG-2 bit stream. Line A is dashed, while line B is solid, since the GA decoder is a super set of an MPEG-2 decoder. Line C is also dashed, since there is not yet a requirement that a GA television be able to decode and display a compressed digital standard definition bit stream. This level of inter-operability does not require any change to the decoding engine, since MPEG-2 is a subset of the GA algorithm. It will require that the display interpolation functionality be resident in the GA decoder. There is no line from the GA system to the MPEG-2 main because MPEG-2 main level decoder will not be able to decode the GA bit stream due to memory and speed constraints.

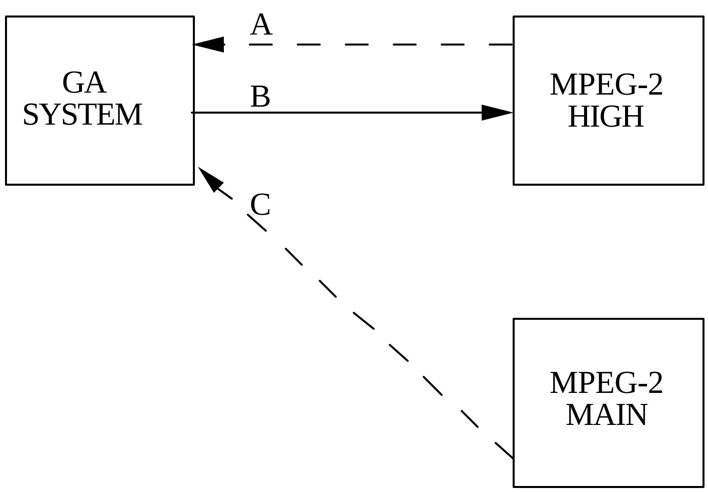


Figure 3.4-1. Inter-operability between GA and MPEG-2

3.4.2 Inter-Operability with Computers

The GA system will inter-operate with computer displays in a similar manner, as shown in Figure 3.4-2. Here, the bit stream inter-operates at either the Transport Layer or the Compression Layer. The compression layer will syntactically inter-operate, however, the computer will require some display processing to convert the video frame-rate assuming a computer display refresh rate of * 66 Hz as is commonly the case. The windowing system will also manage the format conversion between GA video formats and the display format native to the computer. The format conversions required will be: frame-rate conversion, picture aspect ratio (16:9 versus 4:3), interlaced to progressive (when required), and pel aspect ratio (when required).

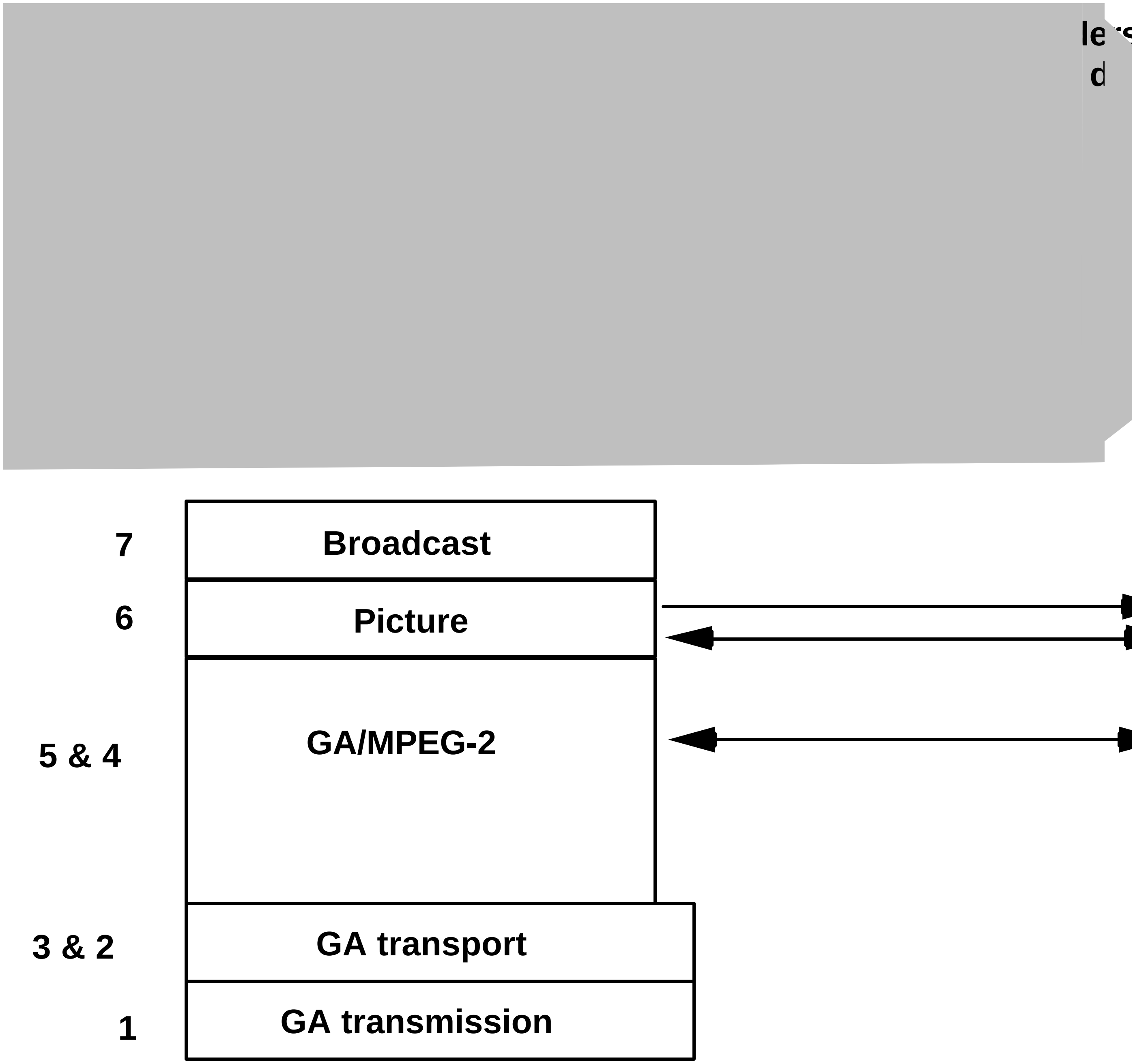


Figure 3.4-2. GA Inter-operability with Computers

3.5. Prototype Implementation Issues

A prototype video coder-decoder (codec) that demonstrates the salient features of the GA video compression specification (as described in this document) will be built. This section describes some of the features. The coding parameters and algorithms of the compression system are being fine tuned through simulation of the system in software. All those elements in the compression specification that add to the quality of the compressed/uncompressed video will be implemented. It must be pointed out that the GA video coding specification, being a bit stream/decoder standard, will allow further improvements to the encoder in the future without any incompatibility with video decoders that already exist.

The video codec will be able to compress multiple video formats depending upon the source (both interlaced and progressive). The compression technique is essentially independent of the video format, although the coding parameters change. The codec will operate at compressed bit- rates of around 17 Mbps.

3.5.1 Features

3.5.1.1 Frame Types, M and N, Refresh

Each video frame is coded in one of three modes: Intra-coding (MPEG I-pictures), Predictive coding (MPEG P-pictures) or Bi-directional predictive coding (MPEG B-pictures). The period of I-pictures (Group-of-pictures size, N) and the distance in number of frames between two anchor (I or P) pictures (known as M), can be programmed.

In the above mentioned mode of operation, I-pictures provide refresh. The codec will allow an alternate way of refreshing, which is achieved through progressive refresh. In this mode, parts of P-pictures (I-slices) are refreshed progressively. Periodic I-frames will be present in progressive refresh to facilitate editing.

3.5.1.2 Motion Estimation

Motion of up to one-half pixel accuracy will be estimated. Integer pixel motion will be estimated on the original video frames, while the sub-pixel refinement will be done on reconstructed pictures. Motion vector search range of up to ± 32 pixels vertically and ± 128 pixels horizontally

will be used for P-pictures. For B-pictures, the motion search range will be ± 32 pixels vertically and ± 64 pixels horizontally.

The specification of the GA compression group includes the dual prime prediction technique for interlaced video (it does not add additional cost to the decoder). This technique is useful for low- delay applications. In the interest of time, this technique will not be implemented.

3.5.1.3 Field Structure Pictures

The specification of the GA compression group also includes the option of using field-structure pictures when processing interlaced video. As with dual prime, this technique will not be implemented in the GA prototype system.

3.5.1.4 Movie Material

When the video codec is presented with material that has been "three-two pulled-down" (originally from a 24 fps movie source), the encoder detects this condition and utilizes the inherent redundancy in the input video to better compress it. The decoder decodes the pictures and displays them as three-two pulled down material.

3.5.1.5 Adaptive Field/Frame Motion Estimation and DCT

Interlaced pictures can be predicted and coded adaptively as fields or frames on a local (macroblock) basis. The adaptive field/frame motion prediction decision will be based on mean absolute error criterion, whereas, the adaptive field/frame DCT will be based on quantized errors or mean absolute error criteria.

3.5.1.6 Rate Control

In order to provide good overall subjective picture quality, a sophisticated adaptive quantization and rate control algorithm will be used. This algorithm will be implemented on a DSP-based subsystem. In addition, a forward analyzer (to assist the rate controller) will be implemented.

3.5.1.7 DCT Coefficient Coding

Compressed video (motion vectors, DCT coefficients, modes, and other headers) will be coded using MPEG-2 VLC tables and syntax. In particular, DCT coefficients will be coded using two VLC tables (one each for inter and intra) in one of two scan orders (zigzag and alternate scans) as prescribed by MPEG-2.

Chapter 4

AUDIO COMPRESSION SYSTEM

4.1 Introduction

The AC-3 audio coding technology is a sophisticated and flexible system for the digital representation of high quality (including multi-channel) sound. A wide range of bit rates and audio coding modes are supported. Features have been incorporated which allow the coded AC-3 data stream to service nearly all listeners, not just those with a particularly ideal listening situation. A section of the data stream referred to as *bit stream information* (bsi) has a number of fixed and optional fields that describe the coded bit stream in some detail. Some of these fields, concerned with audio levels and types of audio production facilities, allow coded bit streams from different sources to be decoded with improved uniformity of level and intended frequency response.

4.2 Technical Details

4.2.1 Overview

An AC-3 serial coded audio bit stream is made up of synchronization frames containing 6 coded Audio Blocks (AB), each of which represent 256 audio samples. A Synchronization Information (SI) header at the beginning of each frame contains information needed to acquire and maintain synchronization. A Bit Stream Information (BSI) header may follow SI, and contains parameters describing the mode of the coded audio service(s). The Audio Blocks may be followed by an Auxiliary Data (Aux Data) field.

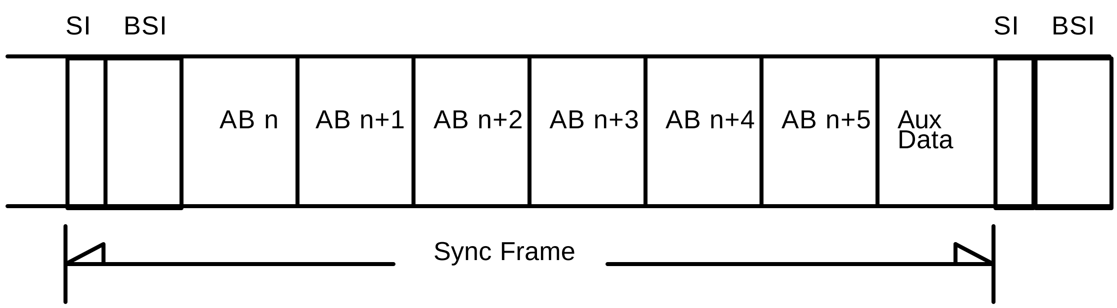


Figure 1 AC-3 Data Stream

4.2.2 Filterbank

AC-3 is a transform coder (see Figure 2) using the Time Domain Aliasing Cancellation (TDAC) transform. The filterbank is critically sampled, and of low computational complexity. Each input time point is represented in two transforms by means of the 50% overlap/add windows. The Fielder window is used which offers an optimum trade-off of close in frequency selectivity and far away rejection. The block size is 512 points which is the optimal transform length for most program material, being a good trade-off between frequency resolution (which determines coding gain and thus minimum bit rate), and complexity (primarily the buffer lengths and memory requirement in the decoder). The frequency resolution of the AC-3 filter bank is 93 Hz, uniform across the spectrum.

The 512-point transform is done every 256 points, providing a time resolution of 5.3 ms (at 48 kHz sampling rate). This time resolution is sometimes insufficient depending on the temporal characteristics of the input signal. During transients a finer time resolution is needed in order to prevent pre-noise artifacts at low bit-rates. The encoder controls the transform length, and during transients will switch to a 256-point transform for a time resolution of 2.7 ms. The algorithm which determines the optimum transform length resides only in the encoder, and may be improved or updated without affecting the decoders already in the field. The transition between blocks is seamless, and the aliasing cancellation of the transform is maintained along with critical sampling. During block switching, there is no increase in filterbank computation rate.

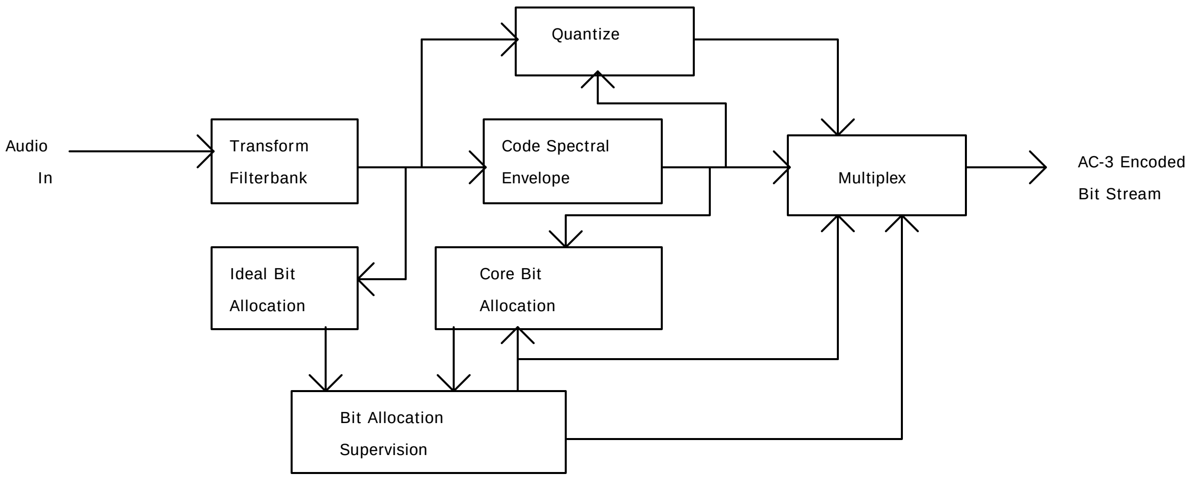


Figure 2 AC-3 Encoder

4.2.3 Spectral Envelope

The output of the 512-point TDAC transform gives 256 real-valued frequency coefficients. The logarithm of the absolute values of these coefficients forms the spectral envelope. The spectral envelope is coded with time and frequency resolution which is signal dependent. The frequency resolution of the spectral envelope can vary from the fundamental filterbank resolution of 93 Hz, up to 750 Hz, depending on the signal. The time resolution of the spectral evelope can vary from 5.3 msec to 32 msec, and is signal dependent. The algorithm which determines the time and frequency resolution of the spectral envelope is in the encoder, and may be updated or improved without affecting the decoders already in the field. The encoder decodes the encoded spectral envelope, so as to make use of the identical information that is available to the decoder. The decoded representation is used both as a reference for the quantization of the transform coefficients, and to drive the bit allocation routine.

4.2.4 Bit Allocation

The decoded spectral envelope has sufficient precision to be used for a psychoacoustic based bit allocation procedure. The encoder performs an iterative bit allocation routine altering the noise to mask ratio of all channels until the number of available bits is used up. Key hint information is coded and transmitted to the decoder which performs the core bit allocation routine only once and arrives at the identical bit allocation which was obtained by the encoder. While not as ideal as a bit allocation routine that uses the full knowledge available at the encoder, this approach saves the overhead required to explicitly transmit the bit allocation to the decoder, and allows more frequent updating of the allocation to be performed. The bits are allocated to each channel out of a common bit pool. Each individual channel may be allocated a differing number of bits.

The bit allocation may be modified by one of two methods. First, parameters of the psycoacoustic model may be altered by side information which is transmitted from the encoder to the decoder. These psychoacoustic model parameters are used in the core bit allocation routine which is computed in both the encoder and decoder. Second, the encoder may optionally send additional side information in order to explicitly alter some or all of the allocation values derived in the core bit allocation routine.

An AC-3 encoder may compute an ideal bit allocation based on full knowledge of the input signal, and compare the results of this allocation to that of the core routine. If the encoder determines that improved subjective results can be obtained by altering the parameters of the psychoacoustic model used by the core routine, it can do so and send the appropriate side information to the decoder. If the results of the core routine are still sub-optimal, the encoder may send additional side information to explicitly modify the output of the core bit allocation routine.

Since the bit allocation is dependent on the spectral envelope, the bit allocation has time and frequency resolution equal to that of the spectral envelope. The bit allocation may change as often as every 5.3 msec, and the frequency resolution may be as fine as 93 Hz.

The core bit allocation routine used by AC-3 only requires the use of a rudimentary ALU with a pair of 16 bit accumulators. Only simple instructions such as ADD, SUBTRACT, COMPARE, AND, OR, and conditional branch are required. The psychoacoustic model used by the AC-3 encoder may be changed at any time without requiring a change in the installed base of decoders. The performance of AC-3 may thus improve over time as more knowledge is gained about the human auditory system.

4.2.5 Channel Coupling

At high frequencies the human ear determines directionality based on the signal envelope rather than on the particular signal waveform. Additionally, the human auditory system cannot independently detect directionality for multiple signals which are very close together in frequency. During periods of high bit demand, AC-3 takes advantage of these phenomena by selectively "coupling" (a form of combination) channels together at high frequencies. The method used maintains, within a critical band (a narrow region of frequency), the original powers of both the individual signals and the individual speaker outputs in order to avoid audible artifacts.

Coupling derives coding gains by combining number of individual channel transform coefficients (at a given frequency) into a common coefficient (at that frequency). The combinations only occur at high frequencies, above the "coupling frequency". The set of combined coefficients form the "coupling channel". Only the spectral envelope of the coupling channel is transmitted (instead of all of the individual spectral envelopes), along with the quantized values of the coupled coefficients (instead of all of the individual coefficients). The coupled coefficients are formed into bands, with widths similar to critical band widths of the human auditory system. For each band in each coupled channel a "coupling coordinate" is also transmitted. The coupling coordinate is used by the decoder to reproduce each coupled band out of each coupled channel loudspeaker at the original signal power level.

The channels which are coupled, the threshold frequency above which coupling takes place, and the widths of the coupled bands is determined by the encoder, and may depend on the audio program content. The encoder algorithms used to determine these items may be updated at any time without requiring changes to decoders already in the field.

4.2.6 Synchronization and Acquisition

The AC-3 bit stream syntax has been designed to allow rapid synchronization and acquisition. The sync word is 16 bits long and has good autocorrelation properties. The probability of randomly finding a sync within the encoded data in a 32 msec sync frame is less than 3% (assuming byte level registration of the transport system). The first sync found will be correct 97% of the time, and it will be found within 32 msec. The sync frame may be verified by use of an embedded CRC check. If the CRC checks, one can be assured that true sync has been achieved, and decoding of audio may begin.

4.3 Features

4.3.1 Rates and Modes

The AC-3 algorithm supports industry standard sample rates of 48 kHz, 44.1 kHz, and 32 kHz. The Grand Alliance system may not support the use of more than one sample rate. Nominal bit rates are indicated by a bit field which indicates (via table lookup dependent on sample rate) the number of bytes between sync codes. Nineteen nominal bit rates of 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320, 384, 448, 512, 576, and 640 kb/s are defined in the AC-3 algorithm, with an additional 13 rate codes reserved for future definition. Again, the Grand Alliance system may not support the use of all available bit rates. The main audio service mode can range from simple monophonic (1/0) through stereo (2/0) and up to all combinations of multiple channels (2/1, 2/2, 3/0, 3/1, 3/2). Any of the modes may optionally use a low frequency effects channel (0.1 channel).

4.3.2 Error Handling

The AC-3 syntax includes the use of 16 bit CRC checks for error detection. Some implementations of receiver architecture allow error flags to be delivered to the audio decoder to identify portions of the bit stream which may contain errors. Concatenation of the use of error flags and CRC checking is a useful technique to reduce the frequency of reproduced errors.

The audio decoder must handle an occasional errored packet or dummy packet without significant audio disturbance. In the event of an error, previous audio data may be repeated. The window used for the FFT 50% overlap/add process is a very good window to crossfade the repeated time segment into the output time waveform. The implementation of error concealment does require some additional memory in the decoder, since old data must be kept available until new data is verified as error free. Since the input data rate to the decoder is much lower than the output data rate, it is most efficient to keep a section of the previous input data.

4.3.3 Program Related Information

Several types of static program related information are contained in the bit stream information section of the audio frame. Program related information which is dynamic (such as dynamic range control) is coded into each individual audio block.

4.3.3.1 Center Mix Level

When a program with three front channels is mixed down to one or two channels for mono or stereo reproduction, artistic considerations sometimes require a different mixture of the Center channel signal with respect to the Left and Right signals. In order that the original multi-channel program mix not be compromised for the sake of mono or stereo reproduction, the AC-3 syntax allows different center channel downmix coefficients to be indicated to the decoder.

4.3.3.2 Surround Mix Level

When multi-channel sound which contains surround information is mixed down to stereo or mono, the optimum mixing level of the surround content which is mixed into the output channels may vary depending on the program type. The AC-3 syntax allows differing levels for the surround mixing coefficients to be indicated to the decoder.

4.3.3.3 Dialogue Level

In order to assure that different AC-3 encoded programs reproduce a uniform level of dialogue, there is an indication of dialogue level encoded into the data stream. Audio decoders make use of this information in order to reproduce all encoded programs with uniform loudness, consistent with the requirements of ATSC document T3-186.

4.3.3.4 Dynamic Range Control

The coded audio blocks contain a dynamic range control word which is used to modify the output level of the decoder. This word is encoded by the broadcaster or production studio in order to intentionally restrict the dynamic range of the reproduced sound to that which is suitable for the broad audience. The AC-3 decoder is required to, by default, reproduce audio with the intended (by those artistically responsible for the program) restricted dynamic range. The decoder may permit the listener the option to restrict the use of the dynamic range control word during reproduction, which will have the effect of reproducing more (or all) of the original program dynamic range. The coded AC-3 bit stream can thus serve a wider range of audience, including those who require restricted dynamic range reproduction as well as those who wish to experience the full dynamic range of the original audio production.

The amplitude resolution of the dynamic range gain control word is less than 0.25 dB. This resolution, coupled with the smoothing effect of the gentle overlapping blocks of the TDAC alternating transforms, insures that there will be no audible modulation products generated by changes in the control word. During transient events, the control word may be updated as often as every 2.6 msec. The encoder may use any amount of time look ahead practical (including off-line disc based processing) to generate the most intelligent gain control words. Any algorithm may be used to generate these words (even direct manual control) and all decoders in the field will respond correctly.

4.3.3.5 Production Mix Room Level, Type

Optional fields allow the indication of the type of mixing room used to prepare the program (large cinema or small studio, resulting in slightly differing equalization) as well as the absolute sound pressure level (SPL) of the mixing session. The SPL code may be used along with the dialogue level indication to allow the reproduction of the program at the identical level used during the mixing session. This information may also be used to adjust the reproduced frequency response to compensate for acoustic reproduction at a different SPL level so as to maintain the same perceived frequency response (true loudness compensation based on differential equal loudness hearing contours).

4.3.3.6 Time Synchronization

The AC-3 data stream may be tagged with an hour, minute, second, frame, and fractional frame time code which is useful for synchronization with SMPTE time code. This information is useful to keep audio and video bit streams in sync in the production environment, and may be useful in the home environment as systems become more complex and interoperable.

4.3.3.7 Additional Information

Additional information may be appended to bit stream information and will be ignored by decoders not intended to recognize it.

4.3.4 Associated Services

The AC-3 syntax and the Grand Alliance transport system allow AC-3 bit streams to convey associated services. Each associated service may be one of the following types: visually impaired, hearing impaired, commentary, dialogue, or emergency flash message. Each associated service may have an optional indication of language.

4.3.5 Auxiliary Data

If the audio data rate is restricted to less than the indicated rate, the excess capacity will result in unused bits at the end of the data frame. This auxiliary data will be ignored by the audio decoder, and may be used for any purpose.

4.4 Compatibility

4.4.1 Compatibility with Mono and Stereo Reproduction

The multi-channel AC-3 bit-stream can serve listeners requiring mono and stereo outputs. When fewer than the full number of output channels is required, the particular down mix is done in the decoder and can be optimum for each listening situation. The program originator can indicate which downmix coefficients are appropriate for a given program. The down mix may be done in the frequency domain, so that only the desired number of output channels needs to be transformed back into the frequency domain and buffered for output. This saves on complexity for AC-3 decoders which only have mono or stereo outputs.

4.4.2 Compatibility with Dolby Surround Encoded Programs

When operated in the 2/0 mode, AC-3 will convey 2 channel Dolby Surround matrix encoded soundtracks without degrading the performance subsequently obtained when further decoding the 2 signals into 4 signals with a Dolby Pro Logic surround decoder. The AC-3 coder makes use of internal re-matrixing in order to avoid the unmasking of coding artifacts which may occur when matrix decoding. Re-matrixing places the dominant coding error in the same location as the dominant signal energy, and avoids the case of a loudspeaker reproducing only coding artifacts which may not be masked by signals from other loudspeakers. This feature is especially important when operating at low data rates, such as 2/0 mode at 192 kb/s.

4.4.3 Compatibility with Dolby Surround Decoders

The AC-3 decoder can provide a 2 channel stereo output from a multi-channel bit stream which is compatible with Dolby Surround matrix decoding. This is a useful feature due to the large number of surround decoders already in use in the field which only have 2 channel inputs.

4.5 ATSC Document T3-186

Portions of the ATSC document T3-186 are concerned with audio coding. AC-3 has been designed with the ATSC concerns in mind, and satisfies all of the requirements of T3-186. The audio related sections of T3-186 are referenced below, and the methods by which AC-3 satisfies the requirements are described.

4.5.1 T3-186 Section 1.5.4, Multi-Channel Audio

This section states the requirement for 5 channel coding, with the low frequency enhancement channel optional. Monophonic and two channel stereophonic modes are also required, along with the ability to decode the full 5 channel service into stereo or mono. AC-3 offers all of the necessary coding modes, which are further elucidated below. Lower cost mono or stereo AC-3 decoders may be built which only partially decode a full 5.1 channel data stream, mix the channels down in the frequency domain, and perform the inverse filter bank only on the needed output channels (see section 7 on complexity).

4.5.2 T3-186 Section 1.5.5, Multiple Languages

This section states the requirement to handle multiple languages. Multiple AC-3 data streams may be provided, each conveying a main service in a different language. The AC-3 bit stream information syntax allows a main audio service to be tagged with an internationally recognized language code. Associated services such as commentary and dialogue may be conveyed by AC-3 data streams and each of these may also be tagged with a language code.

4.5.3 T3-186 Section 1.5.6, Audio Services to the Visually and Hearing Impaired

The section states the requirement to allow associated services such as VI or HI to be provided along with a main service. The AC-3 syntax and Grand Alliance transport system allows single channel associated services to be delivered. These services can be tagged as to type: visually impaired, hearing impaired, or commentary.

4.5.4 T3-186 Section 1.5.7, Uniform Loudness

This section deals with uniform loudness. The AC-3 data stream contains a field for a reference level indication, and the decoder must use this information to adjust the reproduce level. This allows the inter-cutting of bit streams from different sources with different loudness calibrations and headroom without undesirable level variations occurring in the level of reproduced dialogue.

4.5.5 T3-186 Section 1.5.8, Dynamic Range Control

This section considers dynamic range control. AC-3 contains an integral dynamic range control system. The broadcast encoder, or another piece of equipment, may insert codes into the data stream which the decoder uses to dynamically alter the reproduced level. This allows the broadcaster to intentionally compress the dynamic range, but allows the listener the option to reproduce either the compressed or the original (or something in-between) dynamic range.

4.5.6 T3-186 Section 1.5.11, Error Correction and Concealment for Audio Services

The section directs the audio system to take effective methods to minimize audible disturbances caused by uncorrectable errors. The AC-3 data stream contains error detecting codes which will allow the AC-3 decoder to determine if the audio data is valid. If the audio data is not valid error concealment is performed, in which a previous block of data is decoded instead of the current errored block. The repeated block is innocuously spliced in with the gentle FFT window function and this process is typically inaudible. In the case of a television set, the transport layer may deliver error flags to indicate that certain packets of data may be in error. In the event entire AC-3 bit stream is removed from the ATV set and delivered to another piece of equipment (interoperability), the error flag may no longer be available. In this case, the final AC-3 decoder can still perform error concealment triggered by the imbedded error detecting codes.

4.5.7 T3-186 Section 3.1.1, Independent Coding Modes

AC-3 can support all of the following bit-rates for an independently coded channel.

Independent 1/0: 32 kb/s, 40 kb/s, 48 kb/s, 56 kb/s, 64 kb/s, 80 kb/s, 96 kb/s, 112 kb/s, 128 kb/s, 160 kb/s, 192 kb/s, 224 kb/s, 256 kb/s, 320 kb/s, 384 kb/s.

(Note: for reasons of simplicity, the Grand Alliance system may not require decoders to recognize all modes, and thus a subset of these may be selected. A reasonable subset would those suggested by the ATSC: 64kb/s, 96 kb/s, and 128 kb/s although some consideration will be given to an additional lower rate.)

4.5.8 T3-186 Section 3.1.2, Composite Coding Modes

AC-3 can support the following composite coding modes at bit-rates exceeding the recommended minimum indicated below. Any of the modes may optionally support the low frequency effects channel.

Composite 3/2:	256 kb/s
Composite 3/1:	256 kb/s
Composite 3/0:	192 kb/s
Composite 2/2:	256 kb/s
Composite 2/1:	192 kb/s
Composite 2/0:	112 kb/s

4.5.9 T3-186 Section 3.1.2, Associated Audio Data

The reference level field in the AC-3 data stream (discussed under 1.5.7 above) satisfies the 3.1.3.1 headroom requirement. The dynamic range control codes (discussed under 1.5.8 above) satisfies the 3.1.3.2 dynamic range requirement. The AC-3 data stream has optional fields available for additional information satisfying the 3.1.3.3 requirement for descriptor information, and the 3.1.3.4 requirement for user bits.

4.6 AC-3 Decoder Architecture

An AC-3 decoder may be efficiently implemented by five functional blocks, as shown in Figure 3. The first block contains an 8-bit serial-parallel converter and input buffer RAM. The coded AC-3 data stream enters the decoder serially and is converted to bytes and stored in RAM.

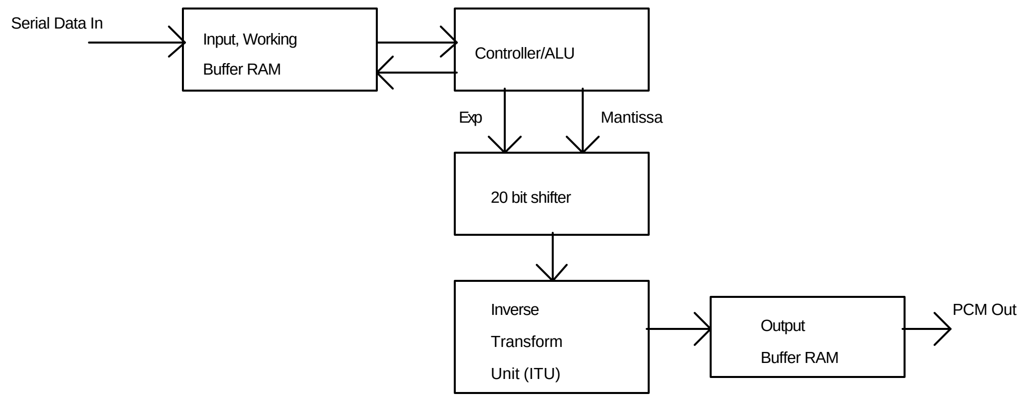


Figure 3 AC-3 Decoder

The second block contains a simple Controller/ALU incorporating two 16-bit accumulators, but no multiplier. The ALU requires the following instructions: add, subtract, compare, and, or, xor, conditional branch, and shift. The third block contains a semi-logarithmic shifter capable of implementing a 0 to 20-bit right shift on mantissas prior to inverse transformation. The semi-log design saves routing area compared to a straightforward 2^5-bit log-shifter. The Controller/ALU coordinates the input buffer and the 20-bit shifter. These three blocks perform the following functions:

1. Accept bit stream input interrupts from the serial-to-parallel converter and load the RAM.
2. Determine frame boundaries (frame synchronization).
3. Unpack control data.
4. Decode the spectral envelope.
5. Perform the core bit allocation routine.
6. Unpack mantissas, right shift up to 20 bits, and deliver data to the ITU.

The fourth circuit block, the Inverse Transform Unit (ITU), performs the frequency-to-time domain conversion (reconstruction synthesis filterbank). The ITU incorporates its own controller circuitry, and may operate somewhat autonomously from the previous circuit elements. Audio blocks representing 256 audio samples are inverse transformed using an efficient fast TDAC transform technique which minimizes computation and RAM requirements. The ITU requires the following instructions: multiply, multiply-add, and multiply-subtract.

Chapter 5

TRANSPORT SYSTEM

5.0 The Grand Alliance Transport System

5.1. Introduction

This document provides a description of the functionality and format of the Grand Alliance transport system. While tutorial in nature, the document provides sufficient technical detail to serve as the operational specification of the transport layer. Issues related to terrestrial broadcast and cable delivery of the ATV service, in the context of ACATS discussions, are addressed in this document. The authors have attempted to make this a stand-alone document, though the reader would gain additional insight from associated ISO-MPEG documents on this and related topics.

In developing the specification of the Grand Alliance transport layer, we have drawn upon the collective experience of the member companies in developing individual systems, as well as the excellent body of work created by the ISO-MPEG standards process. While any system design requires intelligent tradeoffs to be made, selection of a format based on fixed-length packets has maintained a number of simultaneous goals.

5.1.1. Program vs. Transport stream multiplexing

In general there are two approaches for multiplexing elementary bit streams from multiple applications on to a single channel. One approach is based on the use of fixed length packets and the other on variable length packetization. Both approaches have been used in the MPEG-2 standard. As illustrated in Fig. 5.1.1, the video and audio elementary streams in both cases go through an initial stage of PES packetization (discussed in greater depth later), which results in variable length PES packets. The process of generating the transmitted bit streams for the two approaches is shown to involve a difference in processing only at the final multiplexing stage.

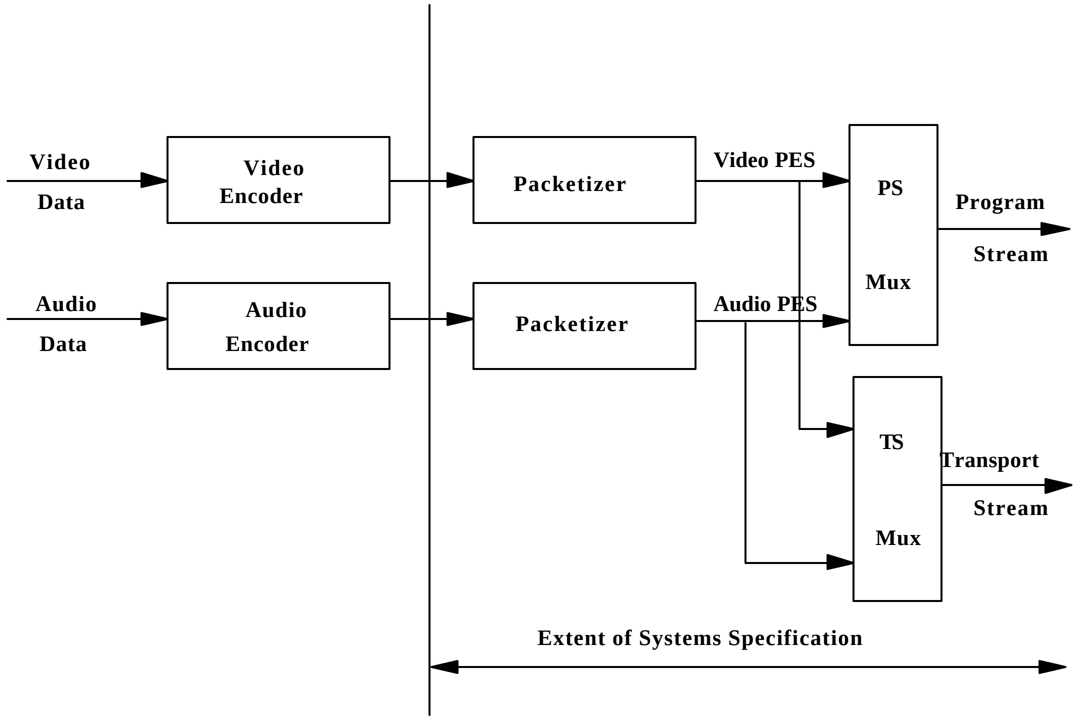


Fig 5.1.1. Comparison of system level multiplexing approaches.

Fig. 5.1.2 gives examples of bit streams for the both program and transport stream approaches, in order to clarify their difference. As shown in Fig. 5.1.2b, in a program stream approach, PES packets from various elementary bit streams are multiplexed by transmitting the bits for the complete PES packets in sequence, thus resulting in a sequence of variable length packets on the channel. (As shown in the diagram, each PES packet is preceded by a PES packet header.) In contrast to this approach, in the transport stream approach selected for the GA system, the PES packets (including the PES headers) are transmitted as the payload of fixed length transport packets. Each transport packet is preceded by a transport header which includes information for bit stream identification. As illustrated in Fig. 5.1.2a, each PES packet for a particular elementary bit stream occupies a variable number of transport packets, and data from various elementary bit streams are generally interleaved with each other at the transport packet layer, with identification of each elementary bit stream being facilitated by data in the transport headers. New PES packets always start a new transport packet in the GA system, and stuffing bytes are used to fill packets with partial PES data.

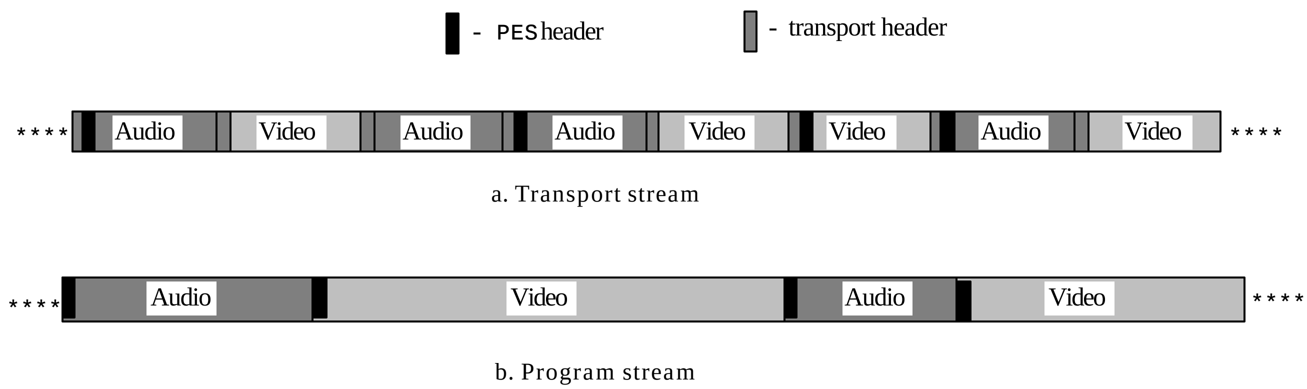


Fig. 5.1.2. Illustration of bit streams for the two packetization approaches.

The two multiplexing approaches are motivated by completely different application scenarios. Transport streams are defined for environments where errors and data loss events are likely, including storage applications and transmission on noisy channels. Program streams on the other hand are designed for relatively error-free media, e.g. CD-ROMs. Errors or loss of data within PES packets can potentially result in complete loss of synchronization in the decoding process in this case. The definition of program stream approach within MPEG-2 is also motivated by the requirement for compatibility with MPEG-1.

The transport stream approach of MPEG-2 has been found to support most of the functional requirements for the GA system, and hence forms the basis of the GA system definition. As will become clearer from discussions in the following sections, the variable packet-size based program stream approach does not meet these requirements in many aspects. Furthermore compatibility with MPEG-1 systems (which is based on the program stream concept) is not a concern for the GA. Note that the GA system transport system can still identify and carry MPEG-1 video and audio services. In general, the program and transport streams both address the same general layers of protocol functionality and therefore it does not make much sense to attempt to carry a program bit stream within a transport stream or vice-versa. Transcoding between the two formats is however feasible and one could in theory build an interface that connects from a GA bit stream to a program stream decoder. The need for such functionality is not anticipated.

Another point to note is that in other ATV scenarios such as CATV and DBS, defacto standards are being set based on the use of the fixed length packetization approach. The approach for the GA system is consistent with the need for simple interoperability with these scenarios.

5.1.2. Advantages of the fixed length packetization approach

The fixed length packetization approach offers a great deal of flexibility and some additional advantages when attempting to multiplex data related to several applications on a single bit stream. These are described in some detail in this section.

Dynamic Capacity Allocation:

While digital systems are generally described as flexible, the use of fixed length packets offers complete flexibility to allocate channel capacity among video, audio and auxiliary data services. The use of a packet id (or PID) in the packet header as a means of bit stream identification makes it possible to have a mix of video, audio and auxiliary data which is flexible and which need not be specified in advance. The entire channel capacity can be reallocated in bursts for data delivery. This capability could be used to distribute decryption keys to a large audience of receivers during the seconds preceding a popular pay-per-view program, or download program-related, computer software to a "smart receiver."

Scalability:

The transport format is scalable in the sense that availability of a larger bandwidth may also be exploited by adding more elementary bit streams at the input of the multiplexer, or even multiplexing these elementary bit streams at the second multiplexing stage with the original bit stream. This is a critical feature for network distribution, and also serves interoperability with a cable plant's capability to deliver a higher data rate within a 6 MHz channel.

Extensibility:

Because there will be possibilities for future services that we cannot anticipate today, it is extremely important that the transport architecture provide open-ended extensibility of services. New elementary bit streams could be handled at the transport layer without hardware modifications, by assigning new packet IDs at the transmitter and filtering on these new PIDs in the bit stream at the receiver. Backward compatibility is assured when new bit streams are introduced into the transport system since existing decoders will automatically ignore new PIDs. This capability could possibly be used to compatibly introduce "1000-line progressive formats" or "3D- HDTV" by sending augmentation data along with the normal ATV data.

Robustness:

Another fundamental advantage of the fixed length packetization approach is that the fixed length packet can form the basis for handling errors that occur during transmission. Error correction and detection processing (which precedes packet demultiplexing in the receiver subsystem) may be synchronized to the packet structure so that one deals at the decoder with units of packets when handling data loss due to transmission impairments. Essentially, after detecting errors during transmission, one recovers the data bit stream from the first good packet. Recovery of synchronization within each application is also aided by the transport packet header information. Without this approach, recovery of synchronization in the bit streams would have been completely dependent on the properties of each elementary bit stream.

Cost Effective Receiver Implementations:

A fixed-length packet based transport system enables simple decoder bit stream demultiplex architectures, suitable for high speed implementations. The decoder does not need detailed knowledge of the multiplexing strategy or the source bit-rate characteristics to extract individual elementary bit streams at the demultiplexer. All the receiver needs to know is the identity of the packet, which is transmitted in each packet header at fixed and known locations in the bit stream. The only important timing information is for bit level and packet level synchronization.

MPEG-2 Compatibility:

The GA transport system is based on the MPEG-2 system specification. While the MPEG-2 system layer has been designed to support many different transmission and storage scenarios, care has been taken by MPEG, as well as the Grand Alliance, to limit the burden of protocol inefficiencies caused by this generality in definition.

An additional advantage of MPEG-2 compatibility is interoperability with other MPEG-2 applications. The MPEG-2 format is likely to be used for a number of other applications, including storage of compressed bit streams, computer networking, and non-HDTV television delivery systems. MPEG-2 transport system compatibility implies that GA transport bit streams may directly be handled in these scenarios (ignoring for the moment the issue of bandwidth and processing speed).

While the GA transport format conforms to the MPEG-2 systems format, it will not exercise all the capabilities defined in the MPEG-2 transport. Therefore, a GA System decoder need not be fully MPEG-2 systems compliant, in that it will not be able to decode any arbitrary MPEG-2 systems bit streams. However, all MPEG-2 decoders should be able to decode the GA bit stream syntax at the transport system level. Documents defining the extent to which the MPEG capabilities are supported in the GA transport have been submitted to the MPEG committee and have contributed to the current working draft of the standard. (See Attachment 1.) MPEG-2 standard features not supported in the GA specification were constrained[1] if they were deemed to not be applicable to broadcast/cable delivery of ATV.

In the development of the GA transport specification, the intent has never been to limit the design by the scope of the MPEG-2 systems definition. If the MPEG-2 standard is unable to efficiently meet the requirements of the GA system, a deviation from MPEG would be in order. The Advisory Committee would be notified should there be a future deviation from MPEG, with justification for the change.

5.1.3. Overview of the transport subsystem

Fig. 5.1.3. illustrates the organization of a GA transmitter-receiver pair and the location of the transport subsystem in the overall system. The transport resides between the application (e.g. audio or video) encoding/decoding function and the transmission subsystems. At its lowest layer, the encoder transport subsystem is responsible for formatting the encoded bits and multiplexing the different components of the

---

[1]The constraint takes the form of a limitation of functionality. In this instance, certain flags will be permanently configured, and some fields will not appear in the Grand Alliance bitstream. This allows a simpler decoder, as it will not be necessary to handle elements not used for the Grand Alliance application. Often constraints are grouped as a "profile" when acknowledged by the MPEG standards body.

program for transmission. At the receiver, it is responsible for recovering the bit streams for the individual application decoders and for the corresponding error signaling. (At a higher layer, multiplexing and demultiplexing of multiple programs within a single bit stream can be achieved with an additional system level multiplexing or demultiplexing stage before/after the modem in the transmitter/receiver.) The transport subsystem also incorporates other higher level functionality related to identification of applications and, as illustrated, synchronization of the receiver. This document will describe these functions in greater detail.

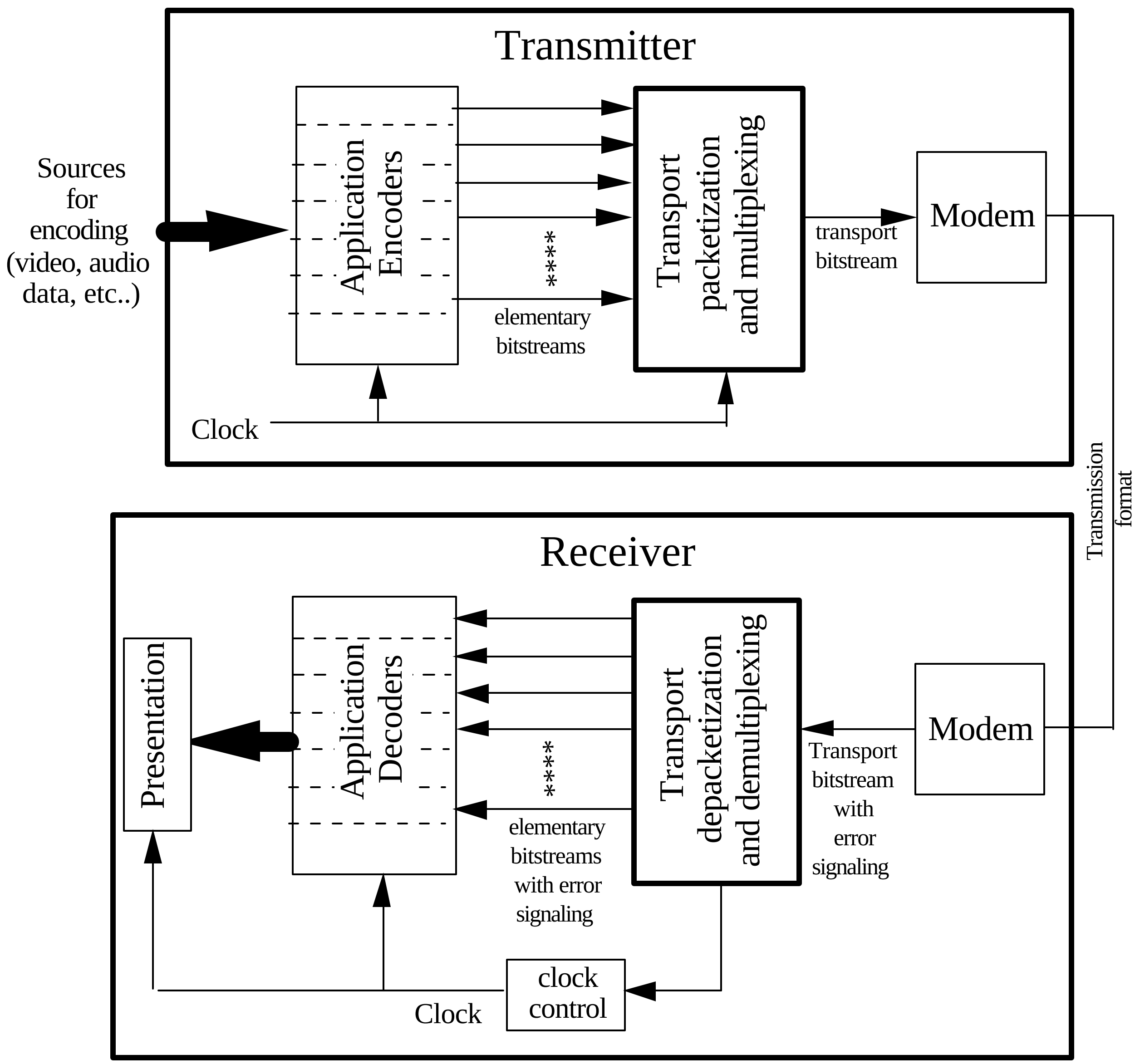


Fig. 5.1.3. Sample organization of functionality in a transmitter-receiver pair for a single GA program.

As described earlier, the data transport mechanism in the GA system is based on the use of fixed length packets that are identified by headers. Each header identifies a particular application bit stream (also called an *elementary bit stream*) which forms the payload of the packets. Applications supported include video, audio, data, program and system control information, etc.. As indicated earlier, the elementary bit streams for video and audio are themselves be wrapped in a variable length packet structure called the packet elementary stream (PES) before transport processing. The PES layer provides functionality for identification, and synchronization of decoding and presentation of the individual application. The format and functionality of a PES packet is described in a later section.[2]

Moving one level up in the description of the general organization of the GA bit streams, elementary bit streams sharing a common time base are multiplexed, along with a control data stream, into *programs*. These programs and an overall system control data stream are then asynchronously multiplexed to form a multiplexed *system*. The organization is described in detail in a later section. Note that programs in the GA system are analogous to today's NTSC broadcast channels.

At this level, the GA transport is also quite flexible in two aspects:

1. It permits you to define programs as any combination of elementary bit streams, for example the same elementary bit stream could be present in more than one program (e.g., two bit streams with the same audio), a program could be formed by combining a basic elementary bit stream and a supplementary elementary bit stream (i.e., bit streams for scalable decoders), programs could be tailored for specific needs (e.g., regional selection of language for broadcast of secondary audio), etc....

2. Flexibility at the systems layer allows different programs to be multiplexed into the system as desired, and allows the system to be reconfigured easily when required. The procedure for extraction of programs from within a system is also simple and well defined.

The GA format provides other features that are useful for both normal decoder operation and for the special features required in broadcast and cable applications. These include

1. Decoder synchronization

2. Conditional access

3. Local program insertion, etc....

The elements of these features that are relevant to the standard definition process will be discussed in detail.

The GA bit stream definition directly addresses issues related the storage and playback of programs. Although, this is not directly related to the ATV transmission problem, it is a fundamental requirement for

[2]Note that the PES layer is not required for all applications. Its use is mandated for both the video and audio in the GA system.

creating programs in advance, storing them and playing them back at the desired time. The programs are stored in the same format in which they are transmitted, i.e., as transport bit streams. The GA bit stream format also has the hooks in it to support the design of consumer digital products based on recording and playback of these bit streams, including the use of the "trick modes" that one is familiar with for current analog VCRs. It should be noted that the issues related to storage and play back of digitally compressed video bit streams are quite different from those that need to be considered for analog systems such as NTSC.

5.1.4. General bit stream interoperability issues

The question has been raised frequently about the bit stream level interoperability of the GA system. There are two sides to this issue. One is whether the GA transport bit stream can be carried on other communication systems, and the other is the ability of the GA system to carry bit streams generated from other communication systems.

The first aspect of transmitting GA bit streams in different communication systems has been addressed to some extent (e.g., for ATM interoperability) in the design of the protocol, and is described in more detail in later sections. In short, there is nothing that prevents the transmission of a GA bit stream as the payload on a different transmission system. It may be simpler to achieve this functionality in certain systems, e.g., CATV, DBS, ATM, etc.., than in others, e.g., computer networks based on protocols such as FDDI, IEEE 802.6, etc.. Since ATM is expected to form the basis of future broadband communications, it is believed that the issue of bit stream interoperability has been addressed for one of the more important transmission scenarios of the future. This is discussed in more detail in Chapter 9.

The other aspect is of transmitting other, non-GA, bit streams within the GA system. This makes more sense for bit streams linked to TV broadcast applications, e.g., CATV, DBS, etc.., but is also possible for other "private" bit streams. This function is achieved by transmitting these other bit streams as the payload of identifiable transport packets. The only requirement is to have the general nature of these bit streams recognized within the GA system context. Note that there is also a certain minimum system level processing defined by the GA that needs to be implemented to extract all (even private) bit streams. The details are made clearer in the sections that follow. It is also important to remember that the GA system is essentially a broadcast system and hence any private transmissions that may be based on a two way communications protocol will not be directly supported, unless this functionality is provided external to the GA system definition.

5.2. The packetization approach and functionality

The GA transport bit stream consists of fixed length packets with a fixed and a variable component to the header field as illustrated in Fig. 5.2.1.

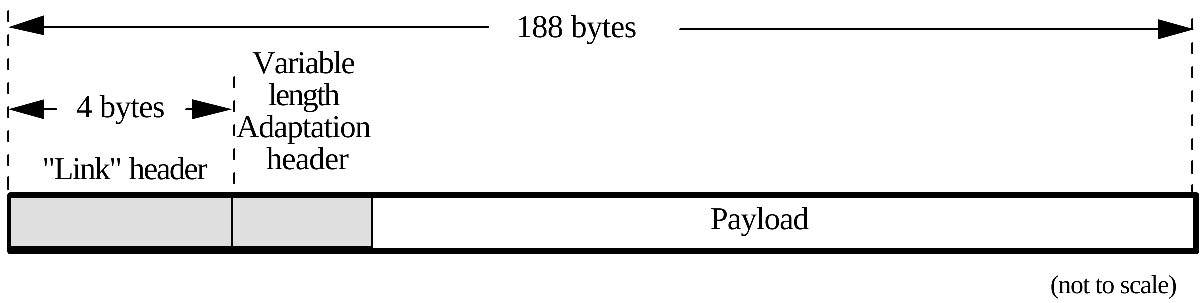


Fig. 5.2.1. GA Transport packet format

Each packet consists of 188 bytes. The choice of this packet size is motivated by a few factors. The packets need to be large enough so that the overhead due to the transport headers do not become a significant portion of the total data carried. They should not be too large that the probability of packet error becomes significant under standard operating conditions (due to inefficient error correction). It is also desirable to have packet lengths in tune with the block sizes of typical, block oriented, error correction approaches, so that packets may be synchronized to error correction blocks, and the physical layer of the system can aid the packet level synchronization process in the decoder. Another motive for the particular packet length selection is interoperability with the ATM format. The general philosophy is to transmit a single GA transport packet in four ATM cells. There are, in general, several approaches to achieve this functionality. Chapter 5.9 includes a discussion of some example approaches. If this interface is to be standardized, the issue will be settled outside the scope of the GA/ACATS process.

The contents of each packet and the nature of this data are identified by the packet headers. The packet header structure is layered and may be described as a combination of a fixed length "link" layer and a variable length adaptation layer. Each layer serves a different functionality similar to the link and transport layer functions in the OSI layers of a communications system. This link and adaptation level functionality is directly used for the terrestrial link on which the GA bit stream is transmitted. However these headers could also be completely ignored in a different system (e.g. ATM), in which the GA bit stream may just remain the payload to be carried. In this environment, the GA bit stream headers would serve more as an identifier for the contents of a data stream rather than as a means for implementing a protocol layer in the overall transmission system.

5.2.1. The "link" layer

The link layer is implemented using a four byte header field. The format of the header field is described in greater detail in a later section. Some of the important functions that are enabled by the header elements are described here.

5.2.1.1. Packet synchronization

Packet synchronization is enabled by the sync_byte, which is the first byte in a packet. The sync_byte has the same fixed, pre-assigned, value for all GA bit streams. In some implementations of decoders the packet synchronization function is done at the physical layer of the communication link (which precedes the packet demultiplexing stage), in which case this sync_byte field may be used for verification of packet synchronization function. In other decoder implementations this byte may be used as the primary source of information for establishing packet synchronization. The standard does not specify the details of the approach to be used to implement this function in a decoder but only provides the hooks in the bit stream to facilitate the function.

5.2.1.2. Packet Identification

As discussed earlier, an important element in the link header is a 13 bit field called the PID or Packet ID. This provides the mechanism for multiplexing and demultiplexing bit streams, by enabling identification of packets belonging to a particular elementary or control bit stream. Since the location of the PID field in the header is always fixed, extraction of the packets corresponding to a particular elementary bit stream is very simply achieved once packet synchronization is established by filtering packets based on PIDs. The fixed packet length makes for simple filter and demultiplexing implementations suitable for high speed transmission systems.

5.2.1.3. Error handling

Error detection is enabled at the packet layer in the decoder through the use of the continuity_counter field. At the transmitter end, the value in this field cycles from 0 through 15 for all packets with the same PID that carry a data payload (as will be seen later, the GA transport allows you to define packets that have no data payload). At the receiver end, under normal conditions, the reception of packets in a PID stream with a discontinuity in the continuity_counter value indicates that data has been lost in transmission. The transport processor at the decoder then signals the decoder for the particular elementary stream about the loss of data. This signaling approach is not included in the standard.

Because certain information (such as headers, time stamps, and program maps) is very important to the smooth and continuous operation of a system, the GA transport system has a means of increasing the robustness of this information to channel errors by providing a mechanism for the encoder to duplicate packets. Those packets that contain important information will be duplicated at the encoder. At the decoder, the duplicate packets are either used if the original packet was in error or are dropped. Semantics for identifying duplicate packets are described in the description of the continuity_counter.

5.2.1.4. Conditional Access

The transport format allows for scrambling of data in the packets. Each elementary bit stream in the system can be scrambled independently. The GA standard specifies the descrambling approach to be used but does not specify the descrambling key and how it is obtained at the decoder. The key must be delivered to the decoder within a time interval of its usefulness. There is "private" data capacity at several locations within the GA transport stream where this data might be carried. Two likely locations would be 1) as a separate private stream with it's own PID, or 2) a private field within an adaptation header carried by the PID of the signal being scrambled. The security of the conditional access system is ensured by encrypting the descrambling key when sending it to the receiver, and by updating the key frequently. As mentioned before, the key encryption, transmission, and decryption approaches are not a part of the standard and could differ in different versions of the ATV delivery system. There is no system imposed limit on the number of keys that can be used and the rate at which these may be changed. The only requirement in a receiver to meet the standard is to have an interface from the decryption hardware (e.g., a Smart-card) to the decoder that meets the standardized interface spec. The decryption approach and technology is itself not a part of the standard. For the purposes of testing, to demonstrate feasibility, only the descrambling function will be tested and the encryption keys will probably be made directly available at the decoder. Note that the generalized systems layer definition includes a mechanism for transmitting key information, including the process of identifying bit streams carrying key information and the definition of the tables that enable this function.

Information in the link header of a transport packet describes if the payload in the packet is scrambled and if so, flags the key to be used for descrambling. The header information in a packet is always transmitted in the clear, i.e., unscrambled. The amount of data to be scrambled in a packet is variable depending on the length of the adaptation header. It should be noted that some padding of the adaptation field might be necessary for certain block mode algorithms. Conditional access is discussed in greater detail in a later section.

Note that the general MPEG-2 transport definition provides the mechanism to scramble at two levels, within the PES packet structure and at the transport layer. Scrambling at the PES packet layer is primarily useful in the program stream (which is not supported in the GA system), where there is no protocol layer similar to the transport to enable this function. In the GA system scrambling will be implemented only at the transport layer.

5.2.2. The Adaptation layer

The adaptation header in the GA packet is a variable length field. Its presence is flagged in the link level section of the header. The functionality of these headers is basically related to the decoding of the elementary bit stream that is extracted using the link level functions. Some of the functions of this layer that are important to the functioning of the GA system are described here.

5.2.2.1. Synchronization and timing

Synchronization of the decoding and presentation process for the applications running at a receiver is a particularly important aspect of real time digital data delivery systems such as the GA system. Since received data is expected to be processed at a particular rate (to match the rate at which it is generated and transmitted), loss of synchronization leads to either buffer overflow or underflow at the decoder, and as a consequence, loss of presentation/display synchronization. The problems in dealing with this issue for a digital compressed bit stream are different from those for analog NTSC. In NTSC, information is transmitted for the pictures in a synchronous manner, so that one can derive a clock directly from the picture synch. In a digital compressed system the amount of data generated for each picture is variable (based on the picture coding approach and complexity), and timing cannot be derived directly from the start of picture data. Indeed, there is really no natural concept of synch pulses (that one is familiar with in NTSC) in a digital bit stream.

The solution to this issue in the GA system is to transmit timing information in the adaptation headers of selected packets, to serve as a reference for timing comparison at the decoder. This is done by transmitting a sample of a 27 MHz clock in the program_clock_reference (PCR) field, which indicates the expected time at the completion of the reading of that field from the bit stream at the transport decoder. The phase of the local clock running at the decoder is compared to the PCR value in the bit stream at the instant at which it is obtained, to determine whether the decoding process is synchronized. In general, the PCR from the bit stream does not directly change the phase of the local clock but only serves as an input to adjust the clock rate. Exceptions are during channel change and insertion of local programming. As mentioned earlier, the nominal clock rate in the GA decoder system is 27 MHz. A point to note here is that the standard only specifies the means of transmitting synchronization information to a receiver but does not specify the implementation of the synch recovery process. Note also that the audio and video sample clocks in the decoder system are locked to the system clock derived from the PCR values. This simplifies the receiver implementation in terms of the number of local oscillators required to drive the complete decoding process, and has other advantages such as rapid synch acquisition.

Details of the format for the PCR are given in a later section. Note that in this implementation the encoder and decoder system clocks are set completely independent of the modem clock. This makes for a clean separation of functionality when implementing the two subsystems, and leads to simpler interfaces. This also makes it simpler to interface GA transmitters and receivers at the transport interface to modems which may be used for transmission on other media such as CATV, DBS, computer networks, etc..

5.2.2.2. Random entry into the compressed bit stream

Random entry into the application bit streams such as video and audio is necessary to support functions such as program tuning and program switching. Random entry into an application is possible only if the coding for the elementary bit stream for the application supports this functionality directly. For example, a GA video bit stream supports random entry through the concept of Intra (or I-) frames that are coded without any prediction, and which can therefore be decoded without any prior information. The beginning of the video sequence header information preceding data for an I-frame could serve as a random entry point into a video elementary bit stream. In general, random entry points should also coincide with the start of PES packets where they are used, e.g., for video and audio. The support for random entry at the transport layer comes from a flag in the adaptation header of the packet that indicates whether the packet contains a random access point for the elementary bit stream. In addition, the data payload of packets that are random access points also start with the data that forms the random access points into the elementary bit stream itself. This approach allows the discarding of packets directly at the transport layer when switching channels and searching for a resynchronization point in the transport bit stream, and also simplifies the search for the random access point in the elementary bit stream once transport level resynchronization is achieved.

A general objective is to have random entry points into the programs as frequently as possible, to enable rapid channel switching.

5.2.2.3. Local program insertion

This functionality is important for inserting local programming, e.g., commercials, into a bit stream at a broadcast headend. In general, there are only certain fixed points in the elementary bit streams at which program insertion is allowed. The local insertion point has to be a random entry point but not all random entry points are suitable for program insertion. For example, for GA video, in addition to being a random entry point, the VBV_delay (video buffer verifier delay) needs to be at a certain system defined level to permit local program insertion.[3] This is a requirement to control the memory needed at the decoder for buffering data and to prevent buffer overflow or underflow. Local program insertion also always takes place at the transport packet layer, i.e., the data stream splice points are packet aligned. Implementation of the program insertion process by the broadcaster is aided by the use of a splice_countdown field in the adaptation header that indicates ahead of time the number of packets to countdown until the packet after which splicing and local program insertion is possible. The insertion of local programming usually results in a discontinuity in the values of the PCR received at the decoder. Since this change in PCR is completely unexpected (change in PCR values are usually only expected during program change), the decoder clock could be thrown completely out of synchronization. To prevent this from happening, information is transmitted in the adaptation header of the first packet after the splicing point to notify the decoder of the change of PCR values (so that it can change the clock phase directly instead of attempting to modify the clock rate). In addition there are constraints on 1) the length of the bit stream that is to be spliced in, to assure that the buffer occupancies at the decoder both with and without the splice would be consistent, and 2) the initial VBV value assumed when encoding the bit stream to be spliced in, in order to prevent decoder buffer underflow or overflow.

The details of the syntax elements that support splicing and local program insertion are described in the chapter on the transport format. More specifics of the particular implementation for the GA system will be described in a separate section.

In addition to the functions described above, the adaptation header includes capabilities for extension of the header, for supporting new functionality, and also for defining data that is private and whose format and meaning are not defined in the public domain. These elements may be useful in extending the GA transport beyond the current range of expectations for usage.

---

[3]The VBV_delay information is computed and transmitted as a part of the header data for a picture in the compressed video bit stream. It defines how full the decoder video buffer should be just before the bits for the current picture are extracted from the buffer, if the decoder and encoder processes are synchronized.

5.3. Higher Level Multiplexing functionality

As described earlier, the overall multiplexing approach can be described as a combination of multiplexing at two different layers. In the first layer one forms program transport streams by multiplexing one or more elementary bit streams at the transport layer, and in the second layer the program transport streams are combined (using asynchronous packet multiplexing) to form the overall system. The functional layer in the system that contains both this program and system level information that is going to be described is called the PSI or Program Specific Information.

5.3.1. Single Program Transport Multiplex

A GA program transport bit stream[4] is formed by multiplexing individual transport packetized elementary bit streams (with or without PES packetization) sharing a common time-base, and a control bit stream that describes the program. Each elementary bit stream, and the control bit stream (also called the elementary stream map in Fig. 5.3.1), are identified by their unique PIDs in the link header field. The organization of this multiplex function is illustrated in Fig. 5.3.1. The control bit stream contains the program_map_table that describes the elementary stream map. The program_map_table includes information about the PIDs of the transport streams that make up the program, the identification of the applications that are being transmitted on these bit streams, the relationship between these bit streams, etc.. The details of the program_map_table syntax and the functionality of each syntax element are given in a later section. The identification of a bit stream carrying a program_map_table is done at the system layers to be described next.

The transport syntax allows a program to be comprised of a large number of elementary bit streams, with no restriction on the types of applications required within a program. A program transport stream does not need to contain compressed video or audio bit streams, or, for example, it could contain multiple audio bit streams for a given video bit stream. The data applications that can be carried are flexible, the only constraint being that there should be an appropriate stream_type ID assignment for recognition of the application corresponding to the bit stream in a GA decoder. The list of application types that will be a supported in the initial GA system are given in the section on services supported by the GA system. Note that the initial selection of applications does not limit the future. (Indeed, it is quite impossible for one to anticipate all possible future applications!)

Note that many of the link level functions are carried out independently, without program level coordination, for the different elementary bit streams that make up a program. This includes functions such as PID manipulation, bit stream filtering, scrambling and descrambling, definition of random entry packets, etc.. The coordination between the elements of a program is primarily controlled at the presentation (display) stage based on the use of the common time base. This common time base is set up by the fact that all elementary bit streams in a program derive timing information from a single clock, the information for which is transmitted via the PCR on one of the elementary bit streams that constitute the program. The data for timing of presentation is present in the elementary bit streams for individual applications.

---

[4]The terminology can be confusing. A **program** is analogous to a channel in NTSC. A **program stream** refers to a particular bitstream format, described in the introduction, that is not being used in the GA system. A **program transport stream** is a term used to describe a transport bitstream that has been generated for a program.

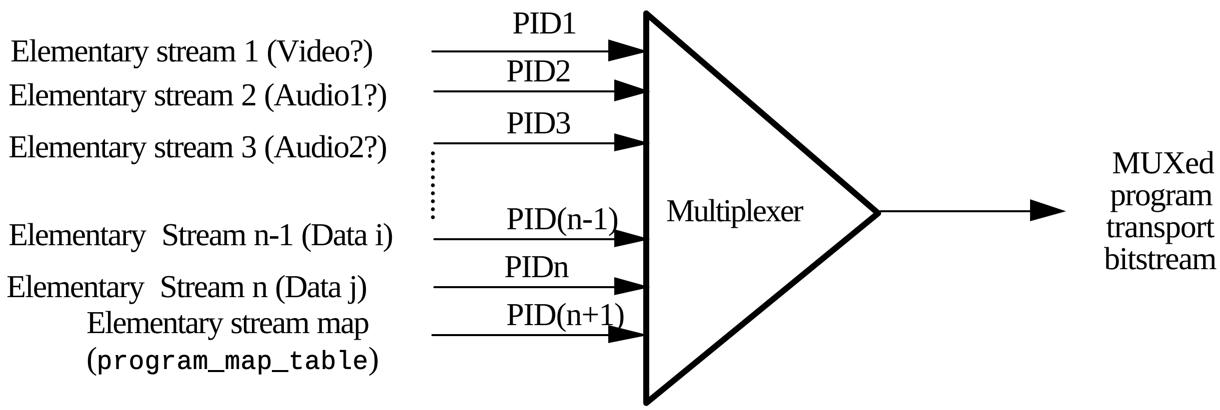


Fig. 5.3.1. Illustration of the multiplex function to form a program transport stream.

Although there is no restriction on the PID values that can be assigned within a program transport multiplex, in light of the fact that PIDs need to be unique at a system level, a standardized PID assignment approach should be considered for the GA system. A suggestion is to use the LSB bits in the PIDs to identify the stream type.

5.3.2. System Multiplex

The system multiplex is the process of multiplexing different program transport streams. In addition to the transport bit streams (with the corresponding PIDs) that define the individual programs, a system level control bit stream with PID = 0 is defined. This bit stream carries the program_association_table that maps program identities to their program transport streams. The program identity is represented by a number in the program_association_table. A program corresponds to what is traditionally called a channel, e.g., HBO™, ESPN™, etc.. The map indicates the PID of the bit stream containing the program_map_table for a program. Thus, the process of identifying a program and its contents takes place in two stages: first one uses the program_association_table in the PID = 0 bit stream to identify the PID of the bit stream carrying the program_map_table for the program, in the next stage one obtains the PIDs of the elementary bit streams that make up the program from the appropriate program_map_table. Once this step is completed the filters at a demultiplexer can be set to receive the transport bit streams that correspond to the program of interest.

The system layer of multiplexing is illustrated in Fig. 5.3.2. Note that during the process of system level multiplexing, there is the possibility of PIDs on different program streams being identical at the input. This poses a problem since PIDs for different bit streams need to be unique. A solution to this problem lies at the multiplexing stage, where some of the PIDs could be modified just before the multiplex operation. The changes have to be recorded in both the program_association_table and the program_map_table. Hardware implementation of the PID reassignment function in real time is helped by the fact that this process is synchronous at the packet clock rate. The other approach, of course, is to make sure up front that the PIDs being used in the programs that make up the system are unique. This is not always possible with stored bit streams.

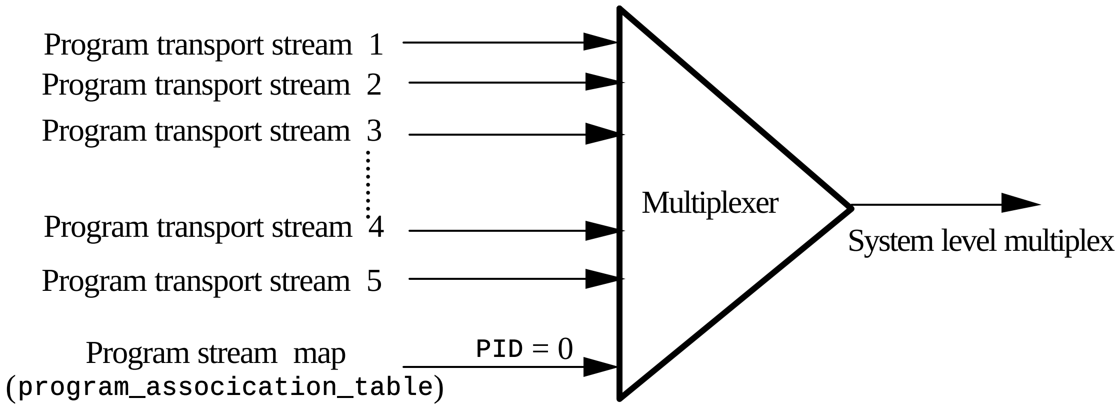


Fig. 5.3.2. Illustration of the multiplex function to form the system level bit stream.

Note that the architecture of the GA bit stream is scalable. Multiple system level bit streams can be multiplexed together on a higher bandwidth channel by extracting the program_association_tables from each system multiplexed bit stream and reconstructing a new PID = 0 bit stream. Note again that PIDs may have to be reassigned in this case.

Note also that in all descriptions of the higher level multiplexing functionality no mention is made of the functioning of the multiplexer and multiplexing policy that should be used. This function is not a part of the standard and is up to individual designers. Because its basic function is one of filtering, the transport demultiplexer will function on any GA bit stream regardless of the multiplexing algorithm used.

Fig 5.3.3 illustrates the entire process of extracting elementary bit streams for a program at a receiver. It also serves as one possible implementation approach (although not the most efficient! In practice the same demultiplexer hardware could be used to extract both the program_association_table and the program_map_table control bitstreams.). This also represents the minimum functionality required at the transport layer to extract any application bit stream (including those that may be private).

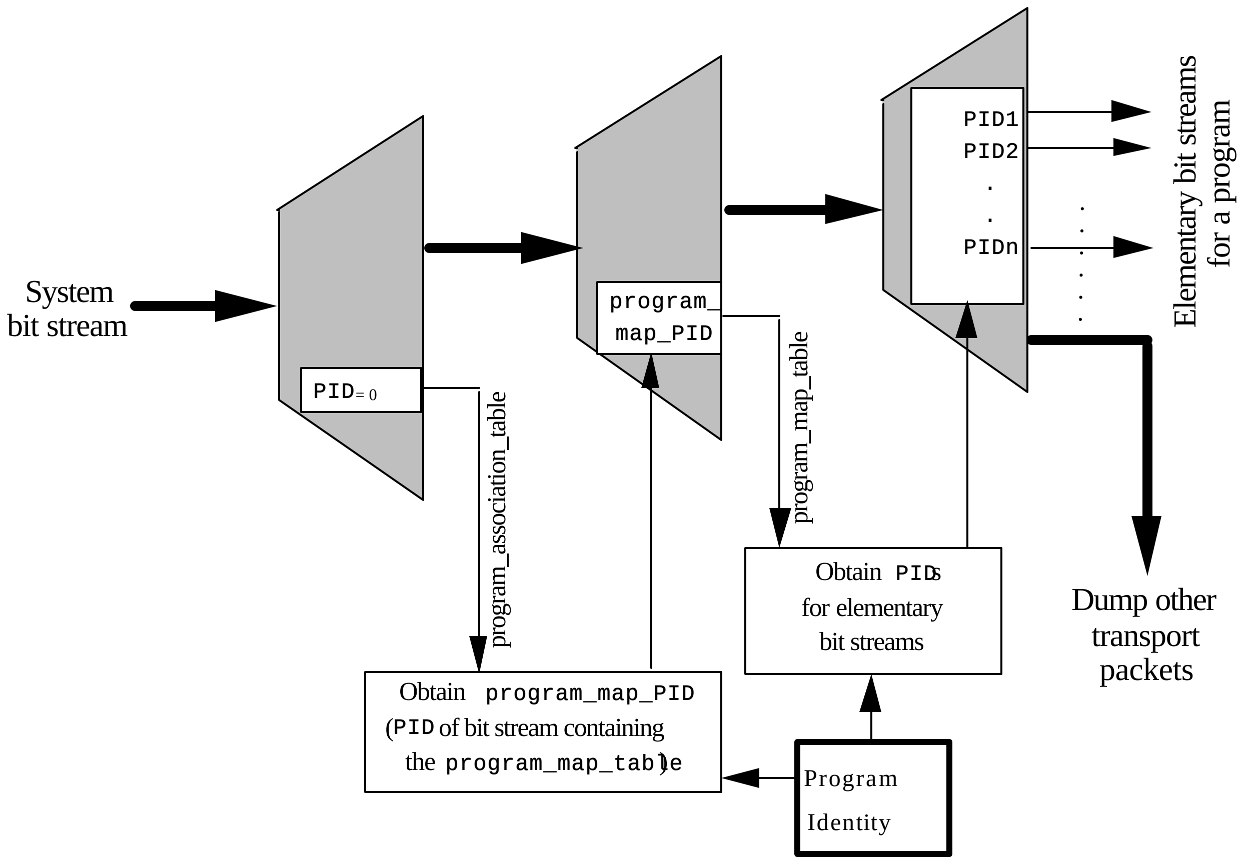


Fig 5.3.3. Illustration of transport demultiplexing process for a program.

Note that once the packets are obtained for each elementary bit stream in the program, further processing stages of obtaining the random entry points for each component bit stream, decoder system clock synchronization, presentation (or decoding) synchronization, etc.., need to take place before the receiver decoding process reaches normal operating conditions for receiving a program.

It is important to clarify here that the layered approach to defining the multiplexing function does not necessarily imply that program and system multiplexing should always be implemented in separate stages. A hardware implementation that includes both the program and system level multiplexing within a single multiplexer stage is allowed, as long as the multiplexed output bit stream has the correct properties as defined in this document.

5.4. Features and Services Supported by the GA ATV System

As we have demonstrated, the GA Transport architecture has been designed to be maximally flexible and is capable of supporting a vast number of services through its system multiplex. The transport, however, is not the only means by which information can be delivered in the ATV system. The GA video syntax also provides means for delivery of predefined information.

Although this is a global issue for the GA ATV system, it is appropriate to review the services supported in the context of the transport specification, since this sub-system has ultimate responsibility for the multiplexing of supported services. In this chapter, we will review a number of functions identified by ATSC document T3/186 and the SMPTE headers/descriptors group and describe how they are carried within the ATV system. In some cases, these functions are a matter to be communicated between the studio that sources the program and the final encoder that transmits the bit stream. While the final transmitted bit stream does not necessarily need to carry this information to the receiver, it is important to see the capabilities of the ATV system to carry this information in a network distribution capacity.

5.4.1. Features Supported within the Grand Alliance Video Syntax

Pan & Scan

Pan & scan information is supported within the GA video syntax. This information is transmitted as an extension within the picture layer syntax. The pan & scan extension allows decoders to define a rectangular region which may be panned around the entire coded image. This facility could be used to identify a 4:3 aspect ratio window within a 16:9 coded image.

Field/Frame Rate and Film Pull-down

The GA video syntax provides means for transmitting the frame rate of the coded bit stream. This allows the encoder to maximize coding efficiency by not transmitting redundant fields, and signals the decoder the proper order for displaying the decoded pictures. The GA syntax supports frame rates of 23.976, 24, 29.97, 30, 59.94 and 60 Hz as well as an extension for future capabilities. The frame rate syntax is found within the video sequence layer.

Picture Structure Information

This information details the sampling structure used in the coded image, including samples per line, lines per frame, and scanning format (interlace or progressive). This information is supported by the video syntax and is found within the sequence layer.

Picture Aspect ratio

The video syntax provides a field for sample aspect ratio within the sequence layer. This information combined with the picture structure fields, allows the picture aspect ratio to be determined.

Color field Identification

This information is supported by the GA video syntax and helps the decoder re-encode the image to an NTSC compatible output with reduced NTSC artifacts.

Colorimetry

Information on the colorimetry characteristics of the video to be encoded are supported by syntax in the video sequence layer. This includes description of the color primaries, transfer characteristics and the color matrix coefficients.

5.4.2. Features Supported as Multiplexed Services within the Grand Alliance Transport System

As mentioned earlier, the GA transport scheme provides great flexibility for multiplexing a variety of services in addition to video and audio elementary streams. Several possible services that could be supported under this transport definition are summarized below.

Audio Compression Types and Language Identification

The transport layer syntax defines a program map which permits identification of individual audio services by their compression algorithm as well as signaling the presence of a secondary language channel that can be selected by the viewer.

Program Information

This service could be provided to the decoder as an ancillary data service with its own PID. This could take the form of a TV guide that is personalized by the service provider. The information would require only a low refresh rate that would not consume a significant amount of the channel bandwidth.

Other Program-related Information

There is a large body of program-related information that could be identified for use at the decoder. MPEG-2 systems syntax, on which the GA system is based, currently supports copyright notification, but has not defined a separate capability for program classification data. This and other program-related information could be addressed as private data.

5.4.3. Support of Closed Captioning and Emergency Alert Messages (Modified 12-07-94)

The Grand Alliance has been participating in dialogue with the EIA committee on ATV Closed Caption Standards to ensure proper support of this important feature. The standards group reviewed a number of important requirements for carriage of the closed caption service in a digitally compressed ATV system. In response to those stated needs, the Grand Alliance has indicated that closed captioning would be carried as user data within the video picture layer.

Support of closed caption in this manner, allows a fixed amount of channel capacity to be dedicated to the service, while maintaining absolute synchronization with each ATV frame. Carriage of this data as a separate service would require a separate synchronization mechanism as exists within the audio decoder. This synchronization was an essential requirement stated by the standards group. Additionally, this mechanism allows for relatively easy editing of the closed caption data downstream. It is expected that the closed caption data will require a dedicated allocation of 9600 bits per second.

Recently, the ATVCC group has resumed a dialogue with the Grand Alliance and ACATS Expert group, clarifying the ATV closed captioning transport requirements. The ATVCC now believes that a closed caption data carried in the video user data could have some limitations. They have requested that the Grand Alliance reevaluate their position and suggest alternative means of carriage, such as a separate PID, for the ATV closed caption data. Carriage of closed caption data as a separate service would require the closed caption data to be PES packetized, including Presentation Time Stamps (PTSs) for synchronization with the audio and video services. As outlined elsewhere in this specification, the closed caption PID would be related to the other services in the program by the program_map_table. As pointed out above, such an approach would place an additional cost burden on the decoder for synchronization and PID filtering. (It should be noted that the GA prototype hardware will support carrying information in user data, as well as demonstrating a separate data service ancillary to the main audio and video services of the ATV program. The test plan also will document the latency of these alternative mechanisms. A complete closed caption decoder, however, is not supported at this time.)

Our work with this standards body will continue, and in the event that it is determined to carry closed caption data in an alternative manner, the documentation will be revised.

To efficiently accommodate emergency alert messages such as storm warnings, the information should be carried in a manner consistent with the closed caption data. The Grand Alliance will document a syntax capability to allow such data to follow the closed caption data. This syntax will be appropriate regardless of whether the information is carried as user data in the video service, or as a separate PID.

This implies that emergency alert messages, such as storm warnings not coded into the video, would require graphical overlay hardware in all receivers if the service provider is required to reach all receivers.

5.4.4. Features not Anticipated to be Transmitted by the Grand Alliance System to the Consumer Receiver

Telecine Source Identification:

Source identification information that could inform the encoder whether the video was originally film or video could assist the detection of redundant fields and thereby improve coding efficiency. There is no syntax provided in the video bit stream to carry this external information, however it could be multiplexed in at the systems level as an ancillary data service.

Image Processing History:

It is projected that future encoders could take advantage of knowledge of any algorithms that have been applied to the image sequence. This is knowledge that would need to be passed from the studio to the encoder, and would not be sent to the decoder. It could be carried in an ancillary data stream in a wider bandwidth network distribution system.

Scene Change:

Automatic scene change detection algorithms are used in some encoders to improve coding efficiency. There has not been an interface specified for this information to be transmitted to the encoder, but it would aid in the coding algorithm. Such scene change information, if supported by a production facility, could provide useful information to the video encoder at both the compression and transport levels and we look forward to working with standards bodies to make this effective.

Conditional Access Identification:

Implementation of Conditional Access systems is supported by the transport syntax, with bits defined in the packet header. Delivering information about the conditional access information, including key information is an issue that would be addressed as private data in the GA syntax. This issue is covered further in a separate chapter on Conditional Access.

Universal Identifier:

Definition of Universal Identifier information has not yet been finalized by SMPTE. The issue of the registration descriptor is addressed more completely in chapter 9.

5.5. The Transport format and protocol

This chapter defines the syntax elements for the transport layer of the GA bit stream. All syntax elements need to be recognized at some level in a GA receiver. Most trigger a response in the transport decoder. A few are present for interoperability with MPEG-2.

5.5.1. Link level headers (Modified 12-07-94)

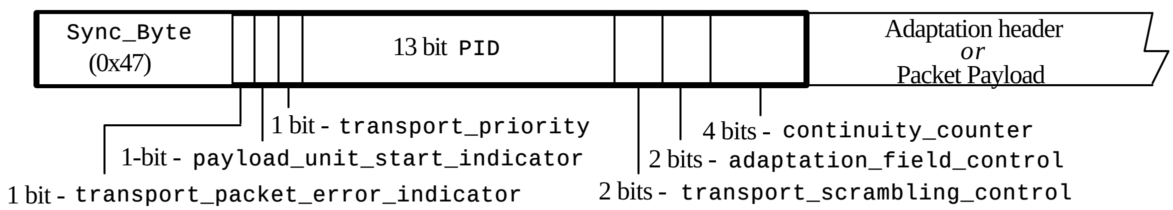


Fig. 5.5.1. Link header format for the GA system transport packet.

Fig. 5.5.1. shows the link layer headers with the functionality assigned to each bit. Some of these have been discussed earlier. The table below spells out these functions in detail. These general functions may not all be used on the broadcast channel, but are useful for transmitting the same bit stream over other links, including cable links, computer networks, etc.. In short, these provide interoperability features.

field	General function	GA usage
Sync_byte Value - 0x47	Packet synchronization.	As defined. See section 5.2.1.1.
transport_packet_error_ indicator	Indicates if packet is erroneous 0 - no error 1 - erroneous packet	Can be used for error signaling from modem to transport demultiplexer. If this bit is set the payload is not supposed to be used.
payload_unit_start_ indicator	Indicates if a PES packet header or the start of a table containing program specific information (PSI) is present in the payload in this packet. The PES packet header always begins the payload of the packet. The starting byte of the PSI table in the packet is indicated using a pointer field to be described later. 0 - no PES header or start of PSI table present 1 - PES header present	As defined for resynch into the transport stream.

transport_priority	Priority indicator at input to transmission channels/networks which support prioritization. 0 - lower priority 1 - higher priority	The GA transmission system is currently not expected to support prioritization. If it does, this bit will be set during transport packetization process, to route packets to the transmission path with the appropriate priority.
PID	Packet Identifier for mux/demux.	As defined. See section 5.2.1.2.
transport_scrambling_ control	Indicates the descrambling key to use for the packet 00 - not scrambled others - user defined.	00 - not scrambled 10 - "even" key 11 - "odd" key 01 -not scrambled, state may be used as a flag for private use defined by the service provider.[5] See section 5.2.1.4.
adaptation_field_control	Indicates if an adaptation field follows 00 - reserved 01 - no adaptation field, payload only 10 - adaptation field only, no payload 11 - adaptation field followed by payload	As defined for PIDs corresponding to elementary bitstreams. No adaptation headers are allowed (i.e., states 10 and 11 not allowed) for transport packets containing PSI information other than for signaling with the discontinuity_indicator that the version_number (Section 2.4.4.4 of ISO/IEC 13818-1) may be discontinuous.[6]
continuity_counter	Increments by one for each packet with a given PID and transport priority. Used at the decoder to detect lost packets. Not incremented for packets with adaptation field of 00 or 10. If two consecutive transport packets of the same PID have the same continuity_counter value and the adaptation_field_control equals '01' or '11', the two transport packets shall be considered duplicate.	As defined. See section 5.2.1.3.

5.5.2. Adaptation level headers

The presence of the adaptation header field is signaled in the adaptation_field_control of the link layer as described before. The adaptation header itself consists of information useful for higher level decoding functions. The header format is based on the use of flags to indicate the presence of the particular extensions to the field.

[5] If left reserved this state could not be used in a Grand Alliance transport bitstream. By defining its use, a service provider could use this state to indicate a special use of the payload, (e.g. entitlement messages). the contents of this packet are not scrambled.

[6] This restriction allows the PMT bandwidth to be limited to a reasonable level without significantly restricting the opportunities and locations for private data to be transmitted.

The header starts with a fixed length component that is always present (if an adaptation header is transmitted). The format is shown in Fig. 5.5.2.

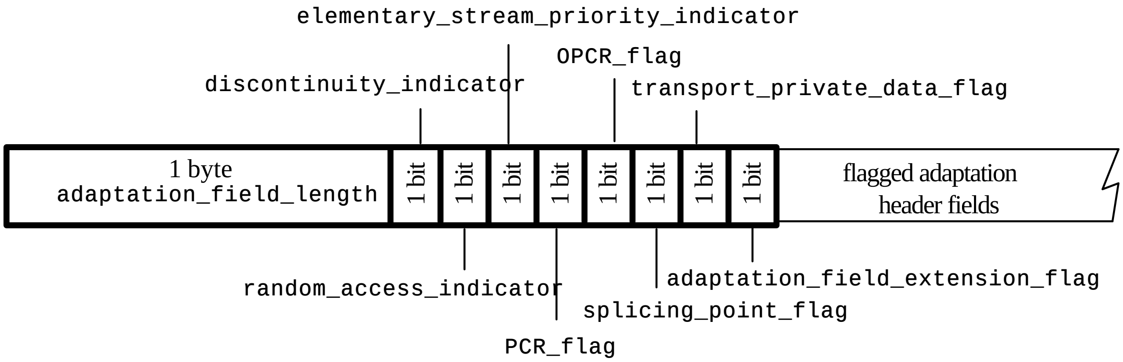


Fig. 5.5.2. Format for the fixed length component of the adaptation header.

The adaptation_field_length specifies the number of bytes that follow it in the adaptation header. The adaptation header could include stuffing bytes after the last adaptation header component field. (Stuffing bytes have a value of 0xff and are not interpreted at the decoder.) In this case the adaptation_field_length also reflects the stuffing bytes. The value in the adaptation_field_length field can also be used by the decoder to skip over the entire adaptation header, and to directly advance to the data payload in the packet if desired.

The presence of additional adaptation header fields is indicated by the state of the last five single bit flags shown in Fig. 5.5.2. (with a value of '1' indicating that a particular field is present). The three flags at the beginning do not result in extensions to the adaptation header and are described in the table below.

field	General function	GA usage
discontinuity_indicator	Indicates if there is a discontinuity in the PCR values that will be received from this packet onwards. This happens when bit streams are spliced. This flag should be used at the receiver to change the phase of the local clock.	As defined for local program insertion. See section 5.2.2.3.
random_access_indicator	Indicates that the packet contains data that can serve as random access point into the bit stream. As an example, these can correspond to the start of sequence header information in the GA video bit stream.	As defined. See section 5.2.2.2.
elementary_stream_ priority_indicator	Logical indication of priority of the data being transmitted in the packet.	Not used. If set it is ignored at a GA decoder.

As mentioned earlier, the other components of the adaptation header appear based on the state of the flags shown in Fig. 5.5.2. The order in which these components appear in the bit stream is the same as the order of the flags. Based on the type of adaptation header information being conveyed, the data in these fields may be either fixed length or variable length. These fields are described in detail next.

5.5.2.1. The PCR and OPCR fields (Modified 12-07-94)

The use of the PCR has been described in detail in section 5.2.2.1. This section deals mainly with the format for transmission.

Overall Format

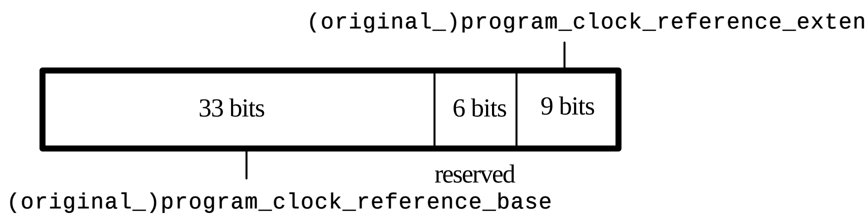


Fig. 5.5.3. The (O)PCR header format.

Functionality

field	General function	GA usage
PCR	Indicates intended time of arrival of last byte of the program_clock_ reference_base at target decoder. Used for synchronization of the system decoding process. This field can be modified during the transmission process.	As defined. The PCR will be transmitted at least once every 100 milliseconds.
OPCR	Indicates intended time of arrival of last byte of the original_program_ clock_reference_base at target decoder for a single program. This field is not modified during transmission.	May be used for recording and playback of single programs. Not used in the GA receiver in the decoding process.

Functional details of the (O)PCR format

The total PCR value is based on the state of a 27 MHz clock. The 9 bit extension field cycles from 0 to 299 at 27 MHz, at which point the value in the 33 bit field is incremented by one. (This results in the 33 bit field being compatible with the 33 bit field that are used for the 90 KHz clock of MPEG-1. Backward compatibility was a concern in the MPEG-2 system design.) The cycle time of the PCR value is approximately 26 hours.

5.5.2.2. The transport_private_data and adaptation_field_extension fields

Format

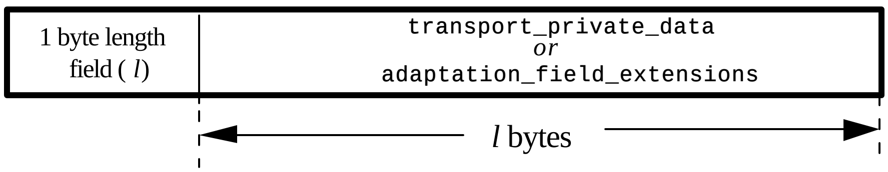


Fig. 5.5.4. The transport_private_data and adaptation_field_extension header format.

Functionality

field	General function	GA usage
transport_private_data	For private data not recognizable by the general MPEG decoders. Meant for short bursts of control information.	To be decided
adaptation_header_ extensions	For future extensions of the adaptation header which may have not been thought of yet.	Not used currently.

5.5.2.3. The splice_countdown field

This field is useful for local program insertion as described in section 5.2.2.3.

Format: This is a one byte field that is present if the splicing_point_flag is set.

Functionality

General Function	GA Usage
Indicates number of packets present in the bit stream, with the same PID as current packet, until a splicing point packet. The splicing point packet is defined as the packet containing a point in the elementary bit stream from which point onwards data can be removed and replaced by another bit stream, so that the resulting transport bit stream is valid according to MPEG-2 rules. Transmitted as a 2s-complement value.	Used for supporting insertion of local programming.

5.5.3. PSIs and the pointer_field (Modified 12-07-94)

As mentioned in chapter 5.3, the program_association_table and the program_map_tables that describe the organization of a multiplexed GA bit stream are a part of the PSI layer of the GA system. PSI tables, in general, are transmitted in the appropriate bit stream sequentially, without any gap between the tables. This implies that tables do not necessarily start at the beginning of a transport packet. This also implies that in order to decode specific tables, there needs to be some indication of where these begin in the bit stream. This functionality is achieved with the pointer_field. The pointer_field, if present, is the first byte of the payload of a packet (after the link and adaptation headers). The pointer_field is present in the packet if a PSI table begins in the packet, an event which is signaled at the link level, by setting the payload_unit_start_indicator (described in section 5.5.1) to '1'. The pointer_field indicates the number of bytes that follow it before the start of a PSI table. As an example a pointer_field value of 0x00 indicates that a new PSI table begins immediately following it.

5.5.4. The program_association_table

As discussed in section 5.3.2, the program_association_table is transmitted as the payload of the bit stream with PID = 0 and describes how program numbers associated with programs, (e.g., HBO™, ESPN™, etc.) map on to bit streams containing the program_map_tables for these programs. This section discusses the syntax of the table in some depth. The program_association_table may be transmitted as multiple program_association_segments with each segment having a maximum length of 1024 bytes. The transport decoder can extract individual table segments from the bit stream in whatever order it desires. As shown in Fig. 5.5.5, each table segment has a fixed length 10 byte header component for table segment identification, a variable length component that depends on the number of entries contained, and a 4 byte CRC-32 field.

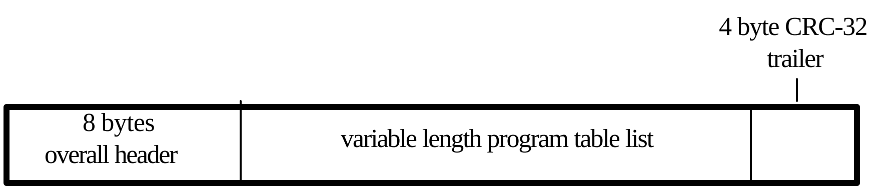


Fig. 5.5.5. High level overall picture of the program_association_segment.

i. The fixed length overall header component is shown below in Fig. 5.5.6 and is described in the table below.

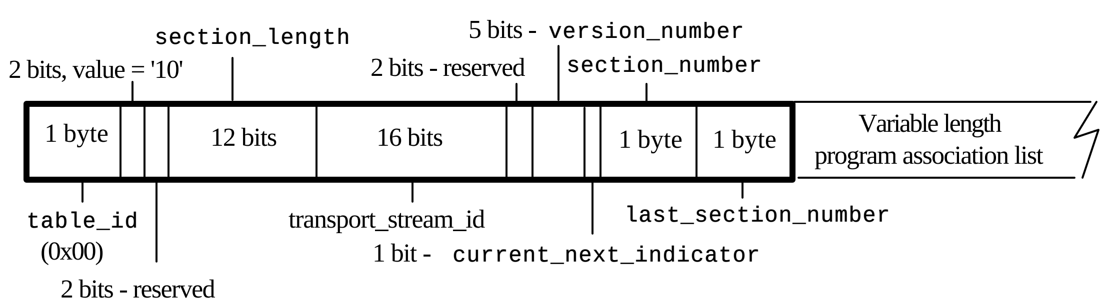


Fig. 5.5.6. Fixed length header component of the program_association_table.

field	General function	GA usage
table_id	Indicates the nature of the table. 0x00 indicates a program_association_table	As defined.
section_length	Length of the section of the program_association_table. This length includes all bytes following this field, up to and including the CRC. The two most significant bits of this field are set to 0, i.e., maximum value is 1024. This field allows the transport decoder to skip sections when reading from the bit stream if desired.	As defined.
transport_stream_id	Identification of a particular multiplex from several in the network.	*Should correspond to a channel number for the terrestrial application.*
version_number	Incremented each time there is a change in the program_association_ table being transmitted.	As defined
current_next_indicator	Value of 1 indicates that the map is currently valid. Value of 0 indicates that the map is not currently being used and will be used next.	As defined
section_number	Identifies the particular section being transmitted.	As defined
last_section_number	Section_number for the last section in the program_association_ table. Needed to confirm when an entire program_association_table has been received at the decoder.	As defined

The value of the reserved bits is undefined and the GA system should not interpret these. On the other hand the 2 bit '10' value following the table_id needs to be received correctly.

ii. The variable length component of the table consists of program_count number of fixed length entries corresponding to each program, and stuffing_bytes (to make up the program_association_segment_length). The format for each fixed length entry is shown in Fig. 5.5.7.

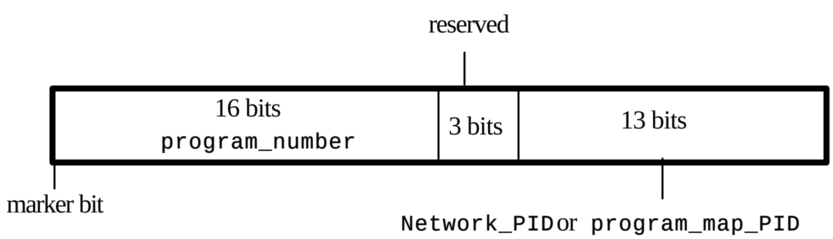


Fig.5.5.7. Format of each entry in the program_association_table.

The program identity '0' is reserved for the network_PID, i.e., the PID of the bit stream carrying information about the configuration of the overall system. The nature of this bit stream is yet to be defined for the GA system. The format for this bit stream is completely open since it is meant to be a private bit stream. For all other program identities, the program_map_PID is the PID of the bit stream containing the program_map_table for the particular program.

iii. The program_association_table ends with a four byte CRC field that contains results of a CRC done over the entire program map segment, starting at the segment_start_code_prefix. The CRC is based on the polynomial $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$.

5.5.5. The program_map_table

As discussed in the previous section, the program_map_table is transmitted as the payload of the bit stream with PID = program_map_PID (as indicated in the program_association_table). The program_map_table carries information about the applications that make up programs. Basically each program_map_table is transmitted as a single TS_program_map_section. The format for a TS_program_map_section can be described as a combination of an overall header field, fields that describe each program within the table and a trailer CRC field, as shown in Fig. 5.5.8. The CRC used is the same as for the program_association_table. In general, each program_map_PID may contain more than one TS_program_map_section, each describing a different program.

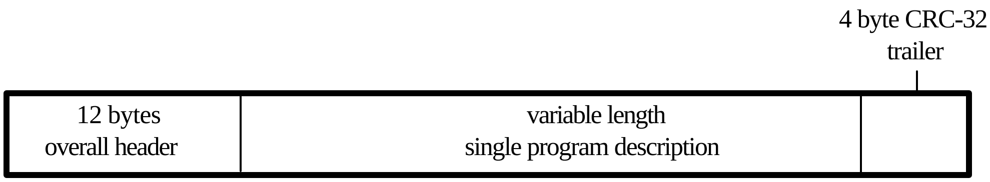


Fig. 5.5.8. High level overall view of the TS_program_map_section.

5.5.5.1. The overall TS_program_map_segment header format

The header format for a TS_program_map_section is shown in Fig. 5.5.9. The format of the first 8 bytes is the same and has similar functionality as that for the program_association_table. The similarity in format is intended to facilitate simple software decoding of the headers. Note that the table_id for TS_program_map_section is different from that for the program_association_table.

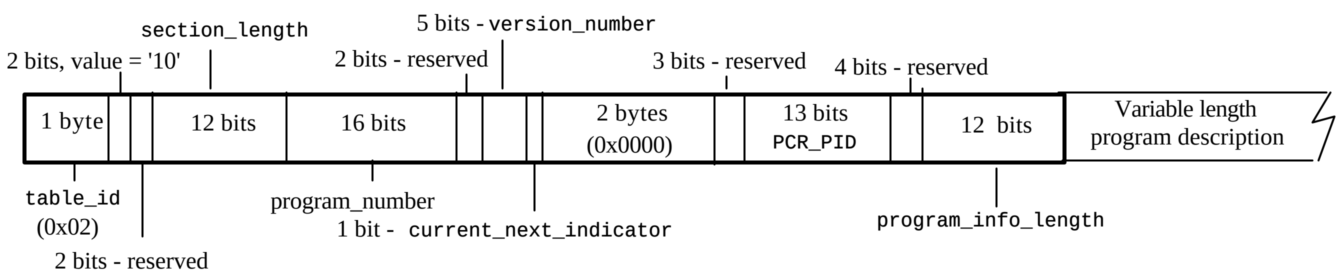


Fig. 5.5.9. Fixed length header for the TS_program_map_section.

The two bytes that were used to identify the transport_stream_id in the association table are now used to identify the program_number of the program whose description follows. The section identification functions are not required for this table since the description of each program is defined to have to fit into one section. Hence these fields are set to '0' as shown in the figure. The other common header fields between the program_association_table and the TS_program_map_section have functionality as described in the table following Fig. 5.5.8. The additional header fields in a TS_program_map_section are the 13 bit PCR_PID field that identifies the PID of the particular packetized elementary bit stream in the program that contains the PCR values for the program, and the program_info_length field, that indicates the number of bytes of program_descriptors that follow this header field.

The overall program description that follows the header described above consists of the optional, variable length, program_descriptor field (whose length was indicated by the program_info_length field shown in Fig. 5.5.9), followed by a descriptions of each of the individual elementary bit streams that make up the program, i.e., there are one or more elementary bit stream descriptions for each TS_program_map_section.

5.5.5.2 An elementary stream description

Each elementary stream description for has a 5 byte fixed length component and a variable length elementary_stream_descriptor component as shown in Fig. 5.5.10.

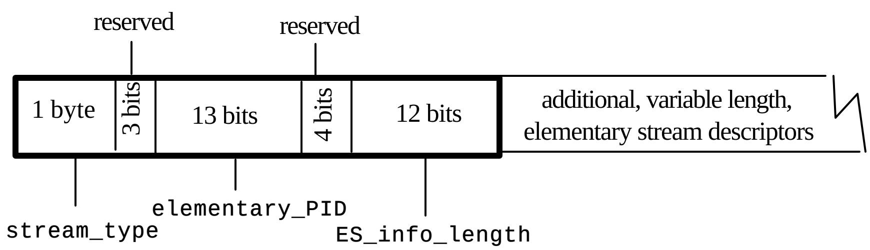


Fig. 5.5.10. The fixed length component of the elementary stream description.

The functionality of the different fields is as follows.

field	General function	GA usage
stream_type	Indicates the application being considered in this elementary stream.	??.
elementary_pid	Indicates the PID of the transport bit stream containing the elementary bit stream.	As defined
ES_info_length	Indicates the length of a variable length elementary_stream_descriptor field that immediately follows.	As defined.

As before reserved bits are ignored.

5.5.6. Descriptors

Descriptors are transmitted in the program_descriptor and the elementary_stream_descriptor fields to describe certain characteristics of the program or the elementary bit stream. In general, each program_descriptor and the elementary_stream_descriptor can consist of number of individual descriptor field elements transmitted sequentially.

Two factors need to be considered in order to use descriptors. In the first place, there has to be an mechanism for indicating the presence of the descriptors. In the PSI tables that have been described, this functionality is achieved by the length field that precedes the descriptor, with a value of zero indicating that no descriptor are present. A second function is the identification of the descriptor itself. This is achieved within the descriptor header itself, which consists of a one byte descriptor_tag field followed by a one byte descriptor_length field that specifies the number of bytes in the descriptor following the descriptor_length field. The set of valid descriptor_tags in the GA systems is the same as that defined for MPEG-2. The table of tags and the format for each descriptor are not described in this document.

5.5.7 The PSI paradigms and constraints (Section added 12-07-94)

The time required to acquire programs is of concern in the GA system design since NTSC users are used to a certain response time from the receivers when they switch channels. The approach used to ensure rapid access is a combination of defining a repetition rate for transmission of different PSI bit stream elements and assigning PID values to bit streams that are related to the program number under consideration. The program paradigms are such that recognition of the paradigm helps speed up the acquisition of programs in the receiver. A receiver that is unaware of the program paradigms will also be able to acquire the program, except that it could take an long time to complete the acquisition process.

5.5.7.1 The program paradigms

The program paradigms provide a simple mechanism for quickly recognizing the programs of interest and further identifying the transport bit streams containing elements of the program that are important for quick acquisition. The basic philosophy is to select PIDs for the transport bit stream that are related to a program number (Program #). The program number, combined with the specification of the carrier frequency (corresponding to the channel number) serves as the complete identification of the program.

TV programs can only be assigned Program #s 1 to 255. Non-TV programs are also not allowed to use these particular Program #s. This allows the receiver to filter the TV programs quickly by looking at the MSB of the Program #. This process is further aided by specifying TV programs to be described only within section 0 of the program_association_table (see 5.4 for definition of a section).

We further define:

base_PID=Program # * 0010h.

In pseudo C code: base_PID = Program # << 4 .

The paradigm to identify the transport bit streams containing certain elements of the program is defined thus:

Name	PID Definition	Description
PMT_PID	base_PID+0x0000h	PID for the bit stream containing the program_map_table for the program.
Video_PID	base_PID+0x0001h	PID for the bit stream containing the video for the program.
PCR_PID	base_PID+0x0001h	Implies the video bit stream also carries the PCR values for the GA program
Audio(A)_PID	base_PID+0x0004h	PID for the bit stream containing the primary audio for the program.
Audio(B)_PID	base_PID+0x0005h	PID for the bit stream containing the secondary audio for the program.
Data_PID	base_PID+0x000Ah	PID for the bit stream containing the data for the program.

Example for program 52 (0x0034h):

base_PID is 0x0340h.

PMT_PID is 0x0340h.

Video_PID is 0x0341h.

PCR_PID is 0x0341h.

Audio(A)_PID is 0x0344h.

Audio(B)_PID is 0x0345h.

Data_PID is 0x034Ah.

This paradigm enables immediate access to services defined in the paradigm once the program number is known. It seems reasonable to expect the channel number and program number to be defined in a printed TV Guide prior to broadcast. Services not defined by the paradigm but included with the program (such as second data channel) would require decoding the program_map_table to obtain their PIDS. PIDs for a program are restricted to those unassigned PIDs between the PMT_PIDs or to PIDs that are not associated with base_PIDs that are in use on the channel.

Note that references to two Audio services as A and B while descriptive for today's NTSC service, it does not adequately describe the audio content for the flexible system discussed here. The reader should consider these to represent a primary audio service, and some secondary audio service to be identified in the future.

<u>Additional constraints include:</u>

1. Only one program is described in a PSI transport bit stream corresponding to a particular PID value. A transport bit stream containing a program_map_table may not be used to transmit any other kind of PSI table (identified by a different table_id).

2. As indicated in section 5.1 in the description of the adaptation_field_control, no adaptation headers are allowed for transport packets containing PSI information.

5.5.7.2 Repetition rates

The maximum spacing between occurrences of section 0 of the program_association_table is 100 msecs.

The maximum spacing between occurrences of a program_map_table containing TV program information is 400 msecs.

5.6. The PES packet format (Modified 12-07-94)

As described before, some elementary bit streams, including the GA video and compressed audio, will go through a PES layer packetization prior to the GA transport layer. The PES header carries various rate, timing, and data descriptive information, as set by the encoder. The PES packetization interval is application dependent. The resulting PES packets are of variable length with a maximum size of 2^{16} bytes, when the PES packet length field is set to its maximum value. This value is set to zero for the GA video stream, implying that the packet size is unconstrained and that the header information cannot be used to skip over the particular PES packet. Note also that the PES packet format has been defined to also be of use as an input bit stream for Digital Storage Media (DSM) applications. Although the DSM format will not be used for GA broadcast application, some of the PES header fields related to the DSM functions are also described in this chapter. Note that the ability to handle input bit streams in the DSM format is not essential for a GA receiver, but may be useful for VCR applications.

PES packets for video, including new PTS and DTS values, occur once every picture (or video access unit) for a terrestrial broadcast. The PES packets are also aligned to the first occurrence of a sequence, GOP or picture start code after the end of a picture, i.e., the first bytes seen within the payload of a video PES packet belong to either a sequence, GOP or picture start code. Further, new PES packet data always starts a new transport packet, and stuffing bytes are used in the adaptation header of the transport packets to ensure that PES packets always end on transport packet boundaries.[7]

Note that the format for carrying the PES packet within the GA transport is a subset of the general definition in MPEG. These choices were made to simplify the implementation of the GA receiver and to also help error recovery.

A PES packet consists of a PES_packet_start_code, PES header flags, PES packet header fields, and a payload (or data block), as shown in Fig. 5.6.1. It is created by the application encoder. The packet payload is a stream of contiguous bytes of a single elementary stream. For video and audio packets, the payload is a sequence of access units from the encoder. The access units correspond to the video pictures and audio frames.

Figure 5.6.1. Structural Overview of a PES Packet

Each elementary stream is identified by a unique stream_id. The PES packets from each encoder carry the corresponding stream_id. PES packets carrying various types of elementary streams can be multiplexed to form a program or transport stream in accordance with Part 1 of the MPEG-2 standard. This chapter deals with

[7]In the April 14th document, the GA has a number of possible constraints under consideration for forming PES packets. The issues were outlined in the original text. After substantial discussion within the GA and with the Expert Group, the text was modified to include a description of those constraints.

the organization of PES packets for MPEG-2 compliant streams only. Specifically, the stream_id field can take on a number of values, indicating the type of data in the payload (See Table for valid values). This section does not define the PES packet structure for 'private data', i.e. stream_id must be 'Reserved', 'Padding', or 'MPEG Audio' or 'MPEG Video'. For 'private data', the PES packet structure is user defined.

The preliminary fields, the packet_start_code_prefix, stream_id, and PES_packet_length, are described in the table below.

Field	Description	GA Usage
packet_start_code_prefix	Indicates the start of a new packet. Together with the stream_id, it forms the packet_start_code. Takes on the value 0 x 00 00 01.	As defined
stream_id	Specifies the type and number of the stream, to which the packet belongs. 1011 1100 -- Reserved Stream 1011 1101 -- Private Stream 1 1011 1110 -- Padding Stream 1011 1111 -- Private Stream 2 110x xxxx -- MPEG Audio Stream Number xxxxx 1110 xxxx -- MPEG Video Stream Number xxxx 1111 xxxx -- Reserved Data Stream Number xxxx	As defined
PES_packet_length	Specifies the number of bytes remaining in the packet after the this Field.	0 x 00 00 - Only allowed value for video. Details for audio yet to be determined

5.6.1. PES header Flags (Modified 12-07-94)

A breakdown of the PES header flags is shown in Figure 5.6.2. These flags are a combination of indicators of the properties of the bit stream and indicators of the existence of additional fields in the PES header. The following table describes the flags present the header. The flags not supported by the GA system are set to '0' and form the basis of some of the "constraints" discussed earlier. (These entries are shaded in the table.) Note that the abbreviations in the tables are from the flag names in Fig. 5.6.2.

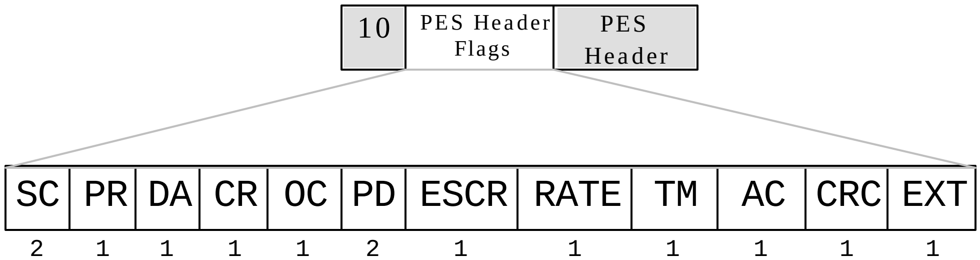


Figure 5.6.2. PES header flags in their relative positions (all sizes in bits)

Flag	Description	GA Usage
PES_scrambling_control (SC)	Indicates the scrambling of the PES packet payload. 00 -- Not Scrambled 01 -- User Defined 10 -- User Defined 11 -- User Defined	Set to 00.
PES_priority (PR)	Indicates the priority of this packet with respect to other packets whose PR field is not set. 1 -- Higher priority 0 -- Same priority	GA does not care how this field is set. (It may be used by applications where necessary.)
data_alignment_indicator (DA)	Indicates the nature of alignment of the first start code occurring in the payload. The type of data in the payload is indicated by the data_stream_alignment_descriptor. 1 - Aligned 0 - No indication of alignment	Must be aligned for video. To be decided for Audio.
copyright (CR)	Indicates the copyright nature of the associated PES packet payload. 1 -- Copyrighted 0 -- Not copyrighted	The GA has yet to define its use of this field.

original_or_copy (OC)	Indicates whether the associated PES packet payload is the original program or a copy. 1 -- Original 0 -- Copy	The GA has yet to define its use of this field.
PTS_DTS_flags (PD)	This flag indicates whether the PTS or PTS and DTS are in the PES header. 00 -- Neither PTS nor DTS is present in header. 1x -- PTS field present 11 -- PTS and DTS field present in header.	This flag is set to 11 when video data alignment indicator is set.[8]
ESCR_flag (ESCR)	Indicates whether the Elementary Stream Clock Reference field is present in the PES Header.	Set to '0'.
ES_rate_flag (RATE)	Indicates whether the Elementary Stream Rate field is present in the PES Header.	Set to '0'.
DSM_trick_mode_flag (TM)	Indicates the presence of an 8 bit field describing the mode of operation of the DSM (Digital Storage Media). 1 -- Field present 0 -- Field not present	Set to '0' for the broadcast transmission. May be used for trick modes when using bit streams specifically generated for VCR type operations.
additional_copy_info_flag (AC)	Indicates the presence of the additional_copy_info field. 1 -- Field Present 0 -- Field not present	The GA has yet to define its use of this field.
PES_CRC_flag (CRC)	Indicates whether a CRC field is present in the PES packet.	Set to '0'.
PES_extension_flag (EXT)	This flag is set as necessary to indicate that extension flags are set in the PES header. Its use includes support of private data. 1 -- Field present 0 -- Field not present	As defined.

[8]The text for the PTS_DTS_flags has been modified to be consistent with the constraint described above under PES packets. Video PES packets will be aligned, and both PTS and DTS will be present in the header of the aligned packet.

5.6.2. The PES header

The PES header immediately follows the field PES_header_length, which indicates the header size in bytes. The size of the header includes all the header fields, any extension fields, and stuffing_bytes. The flags described in the previous section indicate the organization of the PES header, i.e. which fields it does and does not contain. In essence, all the fields of the PES header are optional. Certain applications require particular fields to be set appropriately. For example, GA transport of video PES packets requires that the data_alignment_indicator be set. The trick mode flag is not set in this case. For DSM retrieval of video, the opposite is true. It is the application encoder's function to set the appropriate flags, and encode the corresponding fields. The fields are further described in the following sections. The association between the flags and the corresponding fields is obvious.

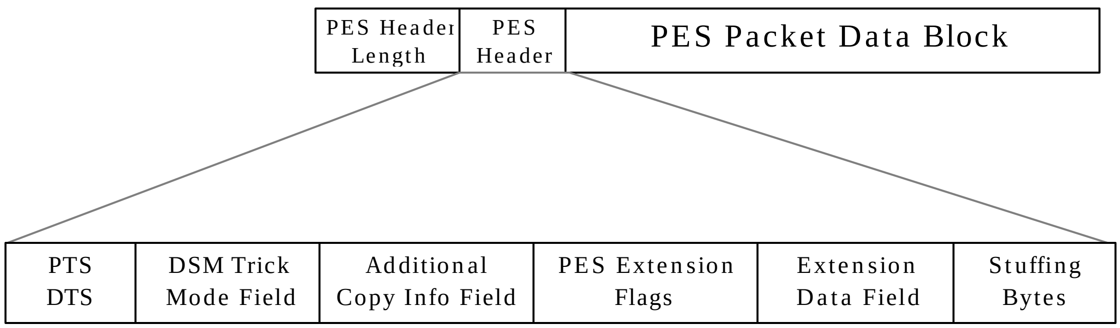


Figure 5.6.3. Organization of PES header.

The PES header Fields are organized according to Fig. 5.6.3 for the GA PES packets for video elementary streams. Most fields require marker bits to be inserted, as described later, in order to avoid the occurrence of long strings of 0's which could resemble a start code.

PTS and DTS

The presentation_time_stamp (PTS) informs the decoder of the intended time of presentation of a presentation unit, and the decoding_time_stamp (PTS) is the intended time of decoding of an access unit. An access unit is an encoded presentation unit. When it is encoded, the PTS refers to the presentation unit corresponding to the first access unit occurring in the packet. If an access unit does not occur in a PES packet, it shall not contain a PTS. Here, a video access unit is said to occur if the first byte of the picture start code is present in the PES packet payload, and an audio access unit occurs if the first byte of the synchronization word of an audio frame is present. Under normal conditions, the DTS may be derived from the PTS. So, it is not required to encode the DTS. Consequently, encoding the DTS may indicate special decoding requirements to the decoder. Under no circumstance does the DTS occur by itself; it must occur along with the PTS although the converse is not true. The PTS field is organized as shown, if it present without the DTS.

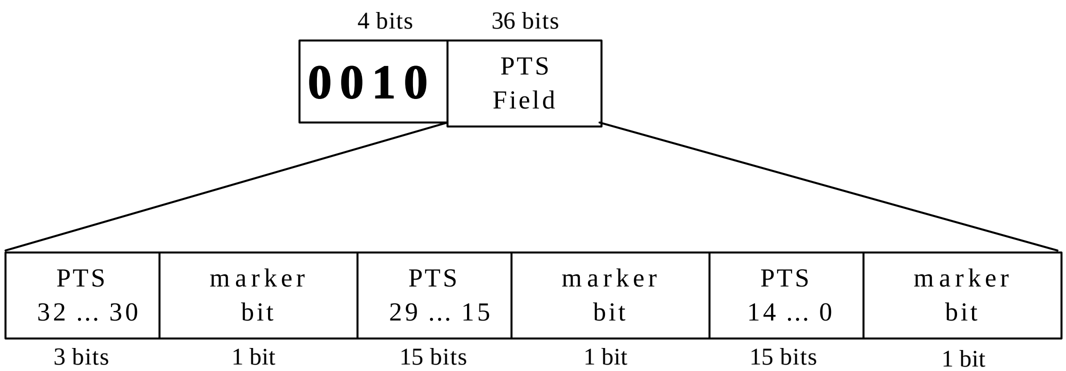


Figure 5.6.4. Organization of the PTS field when only the PTS is encoded.

If both the PTS and DTS are sent, the following organization is required.

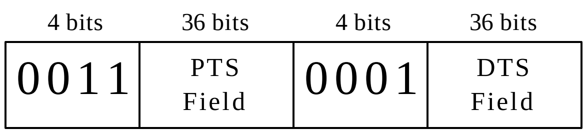


Figure 5.6.5. Organization of the PTS and DTS field when both PTS and DTS are encoded

.

Here, the DTS field is defined in the same manner as the PTS field.

DSM Trick Mode Field[9]

The DSM trick mode field is an eight bit field, indicating the nature of the information encoded in the PES packet. The first three bits of this field form the identifier trick_mode_control, indicating the nature of the DSM mode. There are four modes to the DSM, as summarized in the table below.

[9]It is emphasized once again that the ability to deal with trick mode bit streams is not a basic requirement for a GA receiver and this description is here only for completeness. This field is not present in the broadcast GA bit stream. Since the objective in this document is not to describe in detail the DSM mode of operation, all that is presented is the DSM syntax with some description of the respective fields. For more detailed information the reader is directed to the MPEG-2 Systems document.

Value	Description
‘000’	Fast Forward
‘001’	Slow Motion
‘010’	Freeze Frame
‘011’	Fast Reverse
‘1xx’	Reserved

DSM Trick Modes

Depending on the value of trick_mode_control, a combination of four identifiers are encoded as described in the table below.

Identifier	Description
field_id	This identifier is valid for interlaced pictures only. It is a 2 bit field which identifies how the current frame is to be displayed. ‘00’ -- Display field 1 only ‘01’ -- Display field 2 only ‘10’ -- Display complete frame ‘11’ -- Reserved
frequency_truncation	This 2 bit field indicates the selection of coefficients from the DSM. ‘00’ - Only DC coefficients are sent ‘01’ - The first three coefficients in scan order on average ‘10’ - The first six coefficients in scan order on average This field is for informational purposes only. i.e. the DSM may at times send more than the specified number of coefficients and at other times less. However, that information is not normative.
intra_slice_refresh	This 1 bit field indicates that each picture is composed of intra slices with possible gaps between them. The decoder should replace the missing slices by repeating the collocated sites from the previous decoded picture.
field_rep_cntrl	This field indicates how many times the decoder should repeat field #1 as both top and bottom fields alternatively. After field #1 has been displayed, the decoder should repeat field #2 the same number of times. This identifier being set to 0 is equivalent to a freeze frame with field_id being set to ‘10’.

Fast Forward and Fast Reverse Modes: The format in this case is shown in Fig. 5.6.6. The decoder is told how many coefficients are encoded, how the fields are to be displayed, and how to replace any missing slices in the access units.

trick_mode_control	field_id	intra_slice_refresh	frequency_truncation
3 bits	2 bits	1 bit	2 bits

Fig. 5.6.6. Trick Mode Field in Fast Forward

and Fast Reverse Modes

Slow Motion Mode: In this mode, the DSM informs the decoder how many times a particular field is to be repeated. The identifier field_rep_cntrl is encoded as shown.

trick_mode_control	field_rep_control
3 bits	5 bits

Fig. 5.6.7. Trick Mode Field in Slow Motion Mode

Freeze Frame Mode: Only the identifier is encoded in this mode. It indicates to the decoder how to display the frozen picture. The field is organized as shown in Figure 5.6.8.

trick_mode_control	field_id	Reserved
3 bits	5 bits	5 bits

Fig. 5.6.8. Organization of Trick Mode Field for

Freeze Frame Mode.

Additional Copy Info

This is a one byte field with a marker bit up front and 7 bits of information. The use of these seven bits is yet to be determined.

PES Extension Flags

The header could contain additional flags if the EXT flag (shown in Fig. 5.6.2) is set. These flags are transmitted in a one byte data field as shown in Fig. 5.6.9.

PES private data flag	pack header field flag	program packet sequence counter flag	STD Buffer Flag	Reserved	PES extension field flag
1 bit	1 bit	1 bit	1 bit	3 bits	1 bit

Figure 5.6.9. Organization of the PES Extension Flags Field.

The flags indicate whether further extensions to the PES header exist. The table below describes the nature of this additional data. As with the flags defined previously, the flag is set to '1' if the header field is present.

Field	General Description
PES_private_data_flag	ndicates whether the PES packet contains private data.
pack_header_field_flag	ndicates whether an MPEG-1 systems pack header or an MPEG-2 program stream pack eader is present in the PES Header.
program_packet_sequence_counter_flag	ndicates the presence of a PES packet counter, which allows the decoder to retrieve the ackets in the correct order, from the PES/program multiplex.

STD_buffer_flag	ıdicates whether the STD_buffer_scale and the STD_buffer_size flags are encoded in the PES eader.
PES_extension_field_flag	Indicates the presence of additional data in PES header.

For the GA system, as indicated in the constraint document submitted to MPEG, the flags that are shaded in the table are always set to '0'. It is likely that the EXT flag will set to '0' in the header flags field unless there is information to be transmitted in the PES_extension_field. This field is meant for functions that may have been missed in the initial design specification.

PES Extension Field

This field is shown in Fig. 5.6.10. The length of the PES_extension_field data is given by PES_extension_field_length.

marker bit	PES Extension Field Length	Reserved
1 bit	7 bits	

Figure 5.6.10. Organization of the Extension Field

5.7. Conditional Access

The transport protocol implements functions useful for supporting conditional access. The functionality that is available is flexible and complete in the sense of supporting all transmission aspects of applicable key encryption and descrambling approaches that may be used. Conditional access is also flexible in the sense that it can be exercised on a elementary stream by stream basis, including the ability to selectively scramble bit streams in a program if desired.

A conditional access system operates on the principle of randomizing the transmitted data so that unauthorized decoders cannot decode the signal. Authorized decoders are delivered a "key" which initializes the circuit which inverts the bit randomization. In subsequent discussion, we use the term scrambling to mean the pseudo-random inversion of data bits based on a "key" which is valid for a short time. We use the term encryption to mean the process of transforming the "key" into an encrypted key by a means which protects the key from unauthorized users. From a cryptographic point of the view, this transformation of the key is the only part of the system which protects the data from a highly motivated pirate. The scrambling portion of the process alone, in the absence of key encryption, can be defeated. Conditional Access (CA) is a blanket term for the system which implements the key encryption and distribution. The primary requirements which a scrambling and CA subsystem must meet for digital TV delivery are:

- Protection of programmer's revenues

  - robust against piracy.

- Private encryption system for each program provider.

- Standard consumer instruments

  - no secrets in consumer equipment

- Mobility of consumer equipment

- Consumer equipment should be cost effective

5.7.1. General Description

There are two features of the GA Transport system which support conditional access. The first feature is the two bit transport_scrambling_control field which signals the decoder whether the transport packet was scrambled or not. In the case that it was scrambled, the field identifies which scrambling key was used. As will be shown shortly, the use of two bits in the transport_scrambling_control to define the descrambling process is a necessary and sufficient bound for the key distribution function. The second feature is the ability to insert "private" data at several places in the GA Transport stream. These include entirely private streams and private fields in the adaptation header of the transport bit stream being scrambled. These private fields can be used to transmit the encrypted scrambling key to the decoding device.

The key distribution and usage process is clarified in Fig. 5.7.1. Basically, when the bit stream is scrambled, one descrambling key needs to be in use while the other is being received and decrypted. Two keys are transmitted at any time, with the keys being linked to a transport_scrambling_control value as shown in the figure. The transmission of a key should begin well before it is going to be used, to allow time to decrypt it. Note that this function does not bound the total number of keys that may be used during an entire transmission session.

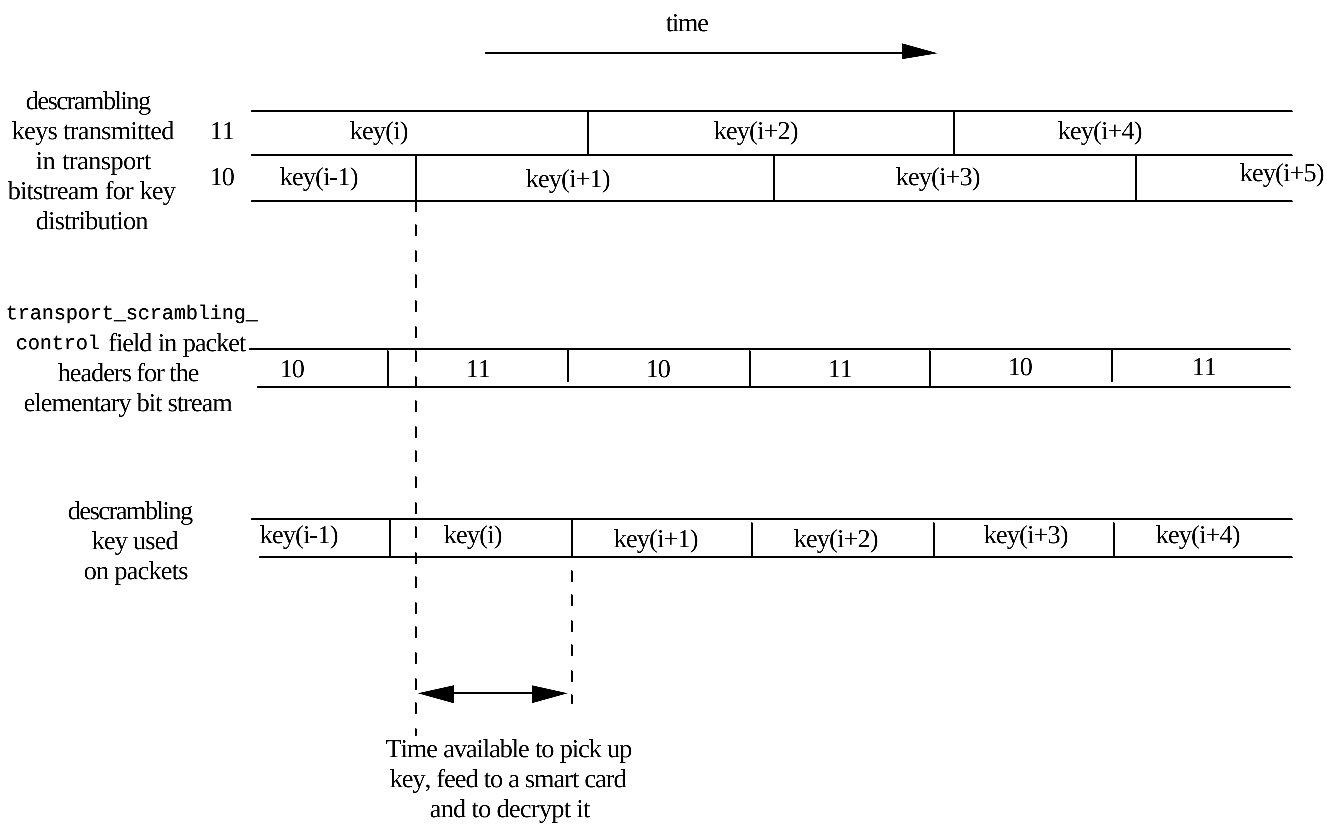


Fig. 5.7.1. Illustration of key distribution and usage process.

As stated previously, the amount of data to be scrambled in a packet is variable depending on the length of the adaptation header. It should be noted that some padding of the adaptation field might be necessary for certain block mode algorithms.

5.7.2. Example of Conditional Access Implementation

In this section, we go through a simple example of a conditional access implementation. Consider the receiver architecture shown in Figure 5.7.2. The high speed manipulations required to implement the descrambling are embedded in the transport demultiplexer, where they are shown as a DES block. Note that other scrambling schemes, such as stream ciphers based on a Pseudo Random Binary Sequence (PRBS), could be employed. The PRBS uses a shift register implementation, where it the initial register value is reset periodically for error robustness. The data security is achieved by the "key" which properly configures the descrambler. This element is delivered to the decoder through an ancillary data service, and is encrypted by the conditional access administrator. In the equipment at the customer's premises, the key is decrypted within the outboard Smart-Card. The Smart-Card interface will conform to ISO standard ISO-7816, which permits a variety of implementations and conditional access solutions.

The Smart-Card maintains a short list of two key's, commonly denoted as the "odd" key and the "even" key. The proper key to be used to descramble is signaled in the transport prefix in the transport_scrambling_control field. The transport_scrambling_control takes on one of the following 4 states:

transport_scrambling_control	*Description*
00	Not Scrambled
01	Reserved
10	"even" key
11	"odd" key

The scrambling in this example is a block cipher called the "Electronic Code Book" mode of the Digital Encryption Standard (DES). For DES, the key is a 56 bit binary sequence. The television electronics are "standard", while the Smart-Card implementation is proprietary to the service provider.

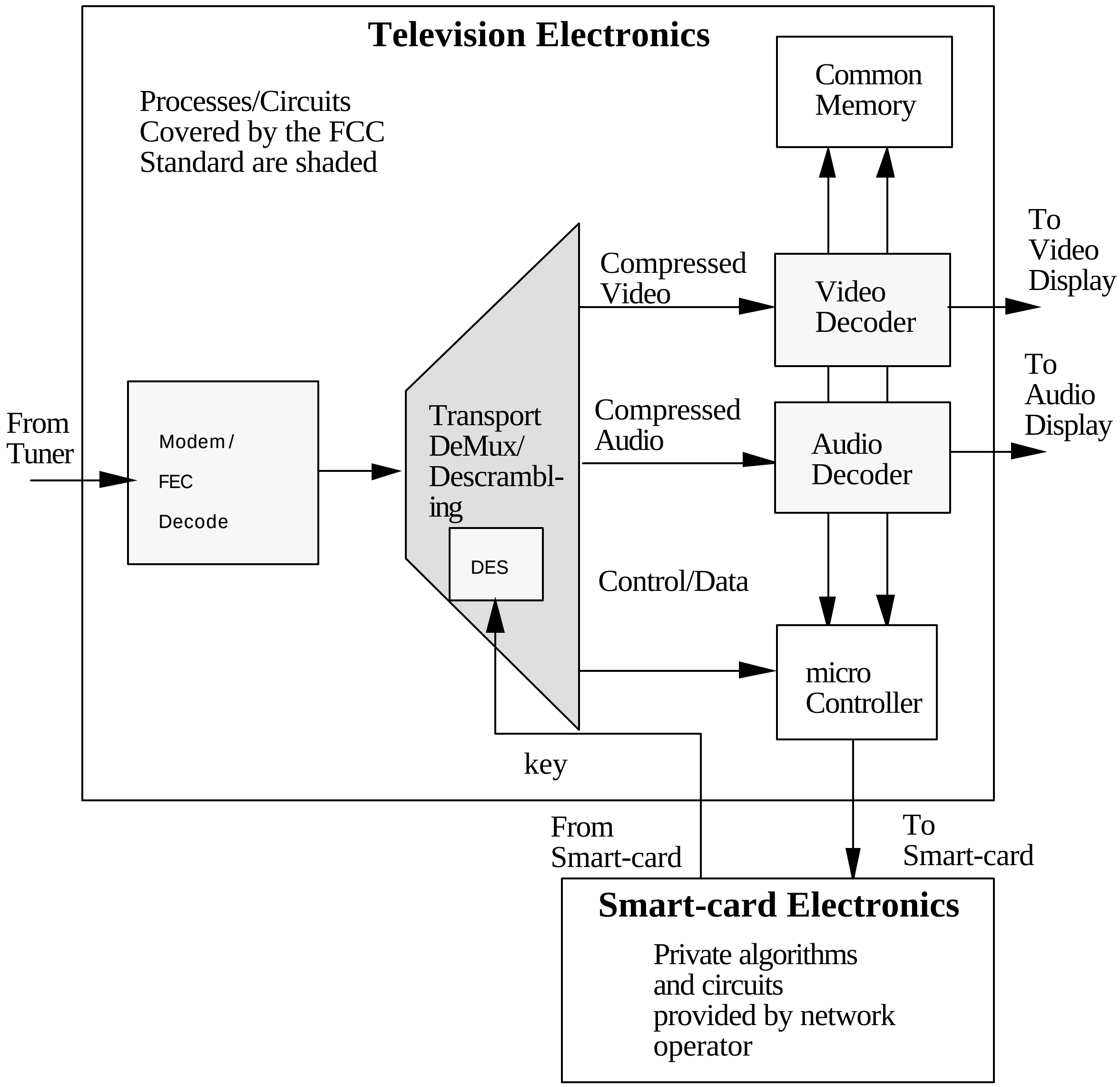


Fig. 5.7.2. Decoder with Example Conditional Access Implementation

There is significant flexibility for multihop encryption and nesting of encryption systems to ensure data security at every point in the transmission chain. Fig. 5.7.3 illustrates nested encryption systems, where the service provider provides authorization, keys and the scrambled data to the end user through Encryption System A. During transit, System B is employed by the carrier to protect the data while in the communications network. In fact, there are two implementation choices even for this segment of the delivery system. The provider can use both a second layer of scrambling in conjunction with a different authorization and key distribution. (This system need not comply with the transmission "Standard's" method.) Alternatively, System B could simply encrypt or scramble the encrypted keys distributed by System A, without the requirement of scrambling the actual service data. The main appeal of this method is that it is a low bandwidth/complexity solution, while its drawback is that it requires some knowledge of the System A key distribution method.

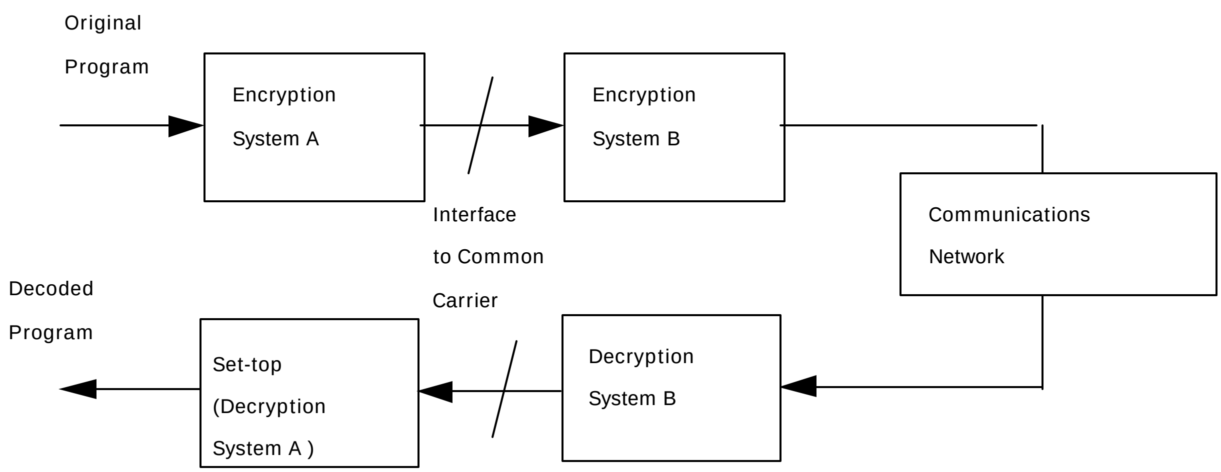


Figure 5.7.3. Nested encryption systems.

In Fig. 5.7.4, the two Encryption Systems are connected in series. System B is again used to protect the integrity of the data while in the communications network. System A is used by the local affiliate or cable company to authorize reception of the service within its own service area.

In fact, the two topologies discussed above can be combined, where either System A or System B could be a Nested or Series configuration. The number of nesting/series combinations can be arbitrarily large.

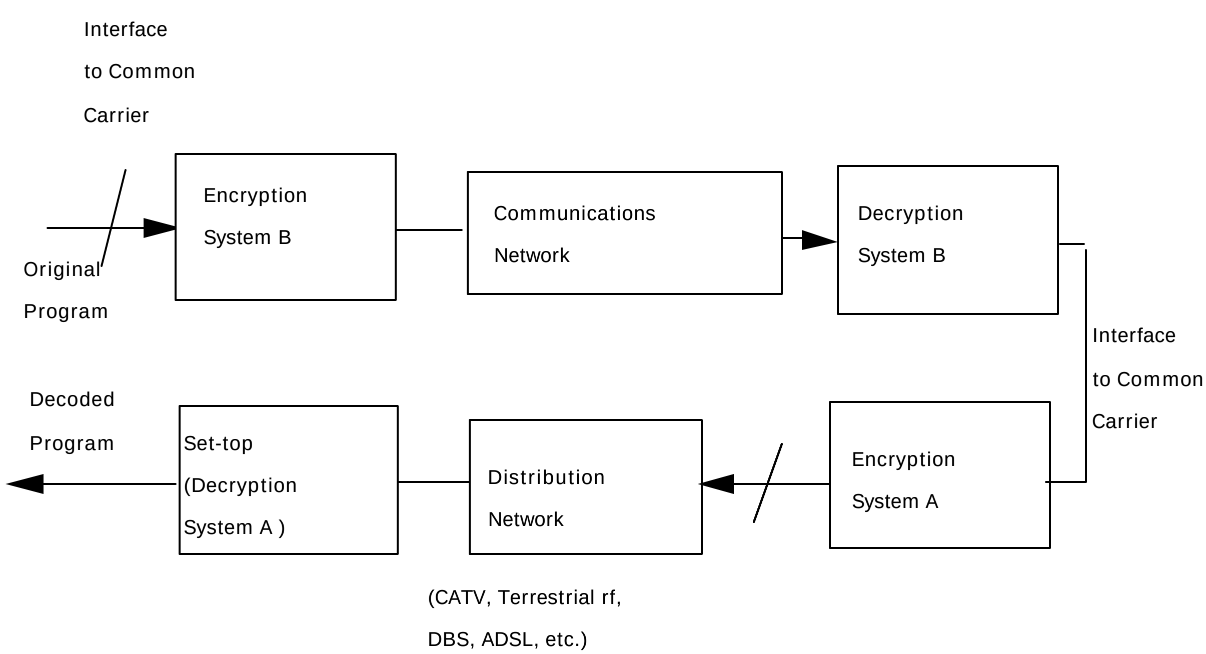


Figure 5.7.4. Series encryption systems.

5.8. Local Program Insertion

The Grand Alliance Transport supports insertion of programs and commercials, by use of flags and features dedicated to this purpose in the transport packets Adaptation Header. The use of these syntax elements will need to be within some imposed constraints to ensure proper operation of the video decoders. Furthermore, there will be some constraints on some of current common practices, imposed not by the GA transport, but rather by virtue of the compressed digital data format.

The functionality of program insertion and switching of channels at a broadcast head-end are quite similar, the difference being in the time constants involved in the splicing process, and also in the fact that in the program insertion the bit stream is switched back to the old program after insertion is complete, while in the channel switching case one most likely switches over to yet another program at the end of the splice. There are other detailed issues related to the hardware implementation that may differ for these two cases, including input source devices and buffering requirements. For example, if program insertion is to take place on a bit stream obtained directly from a network feed, and if the network feed does not include place-holders for program insertion, the input program transport stream will need to be buffered up for the duration of the program insertion. If the program is obtained from a local device, e.g., a video server or a tape machine, it may be possible to pause the input process for the duration of the program insertion. Neither of these is an issue for channel switching.

5.8.1. Systems level view

There are two layers of processing functionality to address when doing program insertion. The lower layer functionality is related to splicing of transport bit streams for the individual elements of the program. The higher level functionality is related to coordination of this process between the different elementary bit streams which make up the program transport stream. Fig. 5.8.1 illustrates the correct approach to implement program insertion.

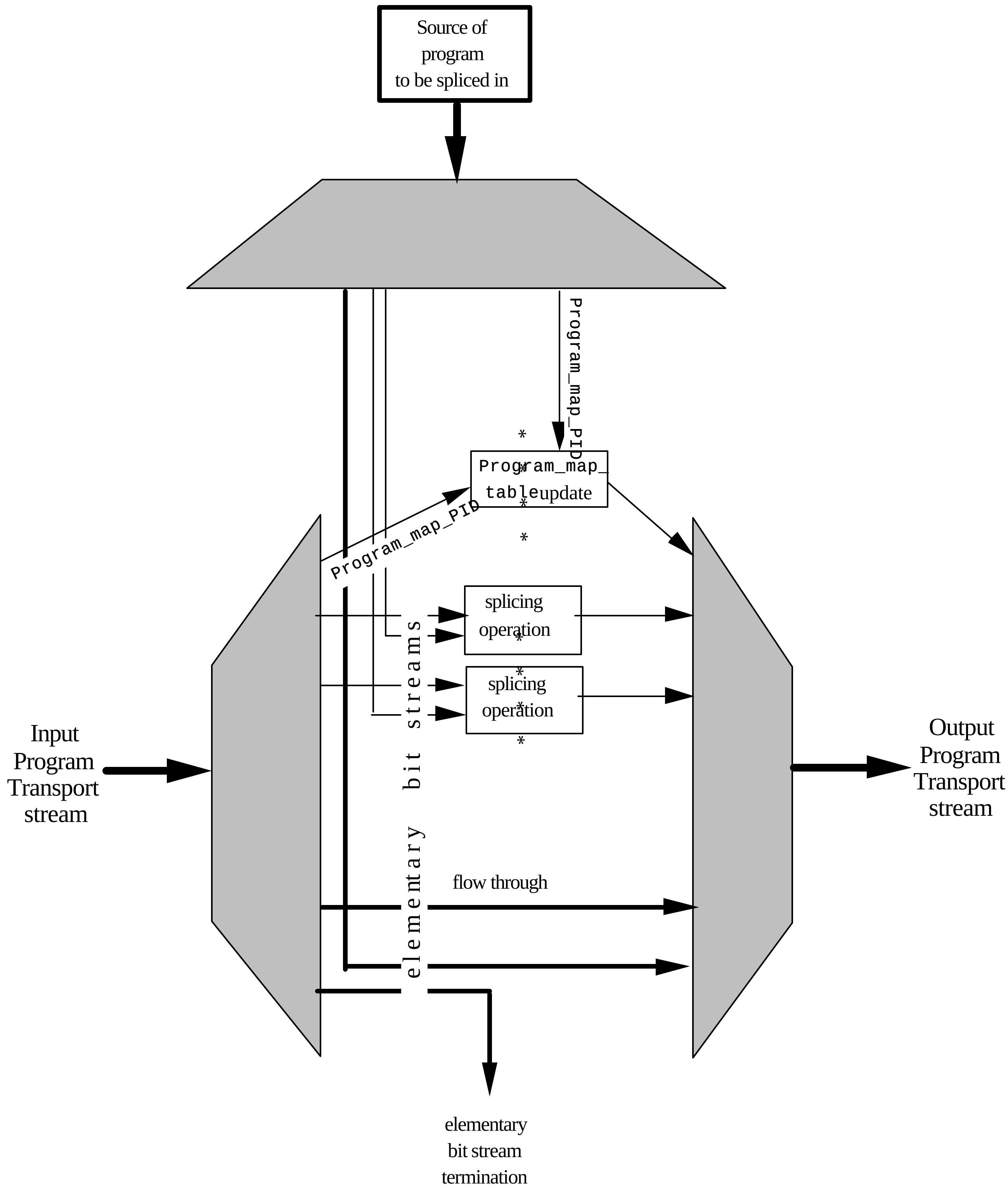


Figure 5.8.1 Example Program Insertion Architecture

The first step for program insertion to take place at a broadcast head-end is to extract (by demultiplexing) the packets, identified by the PIDS, of the individual elementary bit streams that make up the program, including the bit stream carrying the program_map_table. Once these packets have been extracted, as illustrated in Fig. 8.1, program insertion can take place on an individual PID basis. If applicable, some packets may be passed through without modification. There is also the flexibility to add and drop elementary bit streams. The splicing process for each PID is described in the next section.

When program insertion takes place, the program_map_table needs to be modified to reflect the properties of the program transport stream that is being spliced in. As described in the section on its syntax, the definition of the program_map_table allows the signaling of a change in the contents of a program transport stream ahead of time. The changes in the program definition could involve a change in the number of elementary bit streams that make up the program, either by addition or removal of bit streams, change in the PIDs used for the elementary bit streams, etc...

The bit-rate of the program after splicing should have a known relationship to the bit-rate of the program before splicing, in most scenarios. Unless dynamic bit-rate allocation is possible for a program at the system multiplexer (based on instantaneous bandwidth requirements), an increase in bit-rate after splicing can cause buffer overflow. A decrease in bit rate may be handled by transmitting null packets (packets with no information) or by allocating the extra bandwidth to other programs on a dynamic basis. These capabilities depend on the implementation of the system level multiplexing function, a function that is not a part of the GA Transport specification.

Bit rate constraints may also be imposed on individual elementary bit streams for the program that is inserted, e.g., for compressed video, input and output bit rates need to be the same. In a perfect program insertion set up, the splice points for the different elementary streams in a program should be coordinated to correspond to the same instant in time in the overall program (which may not correspond to the same instant in time for each elementary bit stream) to permit seamless transition. Additional constraints on selecting the splicing points exist for particular applications such as video (i.e., VBV_delay value).

5.8.2. Basics of elementary bit stream insertion

The interface for elementary bit stream insertion is at the transport layer of the protocol. This means that bit stream insertion always takes place in units of transport packets. The primary features enabling local elementary bit stream insertion are the discontinuity_indicator field and the splice_countdown fields in the transport header. The discontinuity_indicator signals the decoder that the PCR is changing to a new time base. This simply informs the decoder that the change in the bit stream is not due to an error in the channel, but rather is intended by the program provider. The implication for the decoder is that it should continue normal decoding, and it is the encoders responsibility to make sure that the bit stream has been constructed in a compliant manner (that is that decoders don't crash due to overflow or underflow.)

The splice_countdown field in the adaptation header is used to signal a head-end or intermediate digital switch that a subsequent packet is the point for switching in a new bit stream. The count-down is a positive number which decrements on each subsequent packet of that service. The value of "0" is the last packet in the original sequence, and the value "-1" is resident in the packet which should initiate the switch over. The count will continue to decrement for channel error resilience. The behavior is shown in Figure 5.8.2.

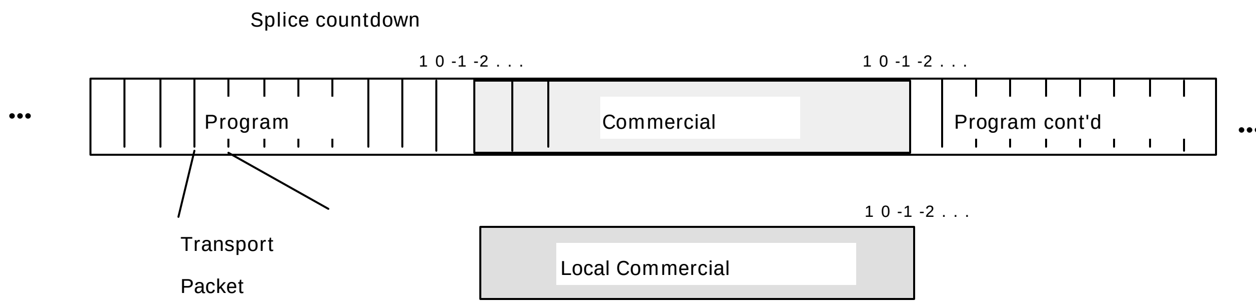


Figure 5.8.2. Local program insertion keyed on Splice countdown.

The affiliate or headend equipment sets itself up for the switch based on the descending countdown. At the "-1" point, the local commercial is inserted while the network commercial continues. At the second trailing "-1", the affiliate returns to the network feed. This technique can be used for either local programs or local commercials. It does depend on the video encoder constraining its bit generation at the splice points so that the decoder buffer does not overflow.

The GA transport encoder places some constraints on the encoding which are more stringent than the MPEG-2 requirements. The added constraint is that the PES header is followed immediately by a video access unit. This will speed acquisition. The first packet in an insertion will contain the PCR value, with the PCR discontinuity bit set to "1" to inform the decoder that a splice has occurred. The first payload in the stream will begin with a PES Header, which will have a PTS resident, so that the decoder can determine the display time immediately. Because the PES header also has the data_alignment_indicator set, the first data following the header will be the start of the video sequence layer. Consequently, the decoder has all the information available to begin decoding immediately after receiving the beginning of the spliced commercial. (In general, an MPEG-2 stream does not have these constraints imposed, and hence does not have guaranteed performance at the splice points.)

5.8.3. Restrictions

Compressed digital technology does impose some restrictions on affiliate operations which may differ from present practice. Although all present practices can be replicated by completely decoding and recoding the video, there is a desire to implement as much as possible in the compressed video domain. The following comments are made with respect to processing the compressed video.

Local commercials and network commercials will need to be strictly controlled to be the same number of packets, and the same number of frames of video. This is contrary to the present practice where local inserts may differ from the planned network inserts by several seconds.

A second restriction is that affiliate pix-in-pix and affiliate text overlays can only be accomplished by decoding and recoding.

5.8.4. Imperfect program insertion

It may not always be possible for the program insertion process to meet the precise requirements of a seamless splice. This could be due to several reasons including the presence of infrequent splice points in the incoming bit stream or non-availability of the hardware required for precise splicing at the network affiliate. There are two scenarios for imperfect splicing. In the first scenario the network affiliate attempts to splice in the entire program as a whole, without attempting to align each of the component elementary bit streams. In this case, the exact splicing can take place for only one of the elementary streams. Since video is the most important component of the program, perfect alignment will be obtained for video. In this case the output of the other elementary bit streams will not be presented at the output until synchronization of these bit streams is achieved. As an example, audio should be muted until its elementary bit stream is synchronized.

In the second and most uncoordinated splicing approach, the splicing takes place without any attempt at coordination with the input bit stream. In this case the video presentation process is affected around the splicing point. If the splicing takes place when the VBV level in the existing bit stream is less than it should be for perfect splice, there will be a period of time for which data is lost for the existing bit stream. In this case the decoder should freeze the last displayed frame. In the other case where VBV is fuller than expected, the decoder video data buffer may eventually overflow during the time period of the spliced in bit stream. The decoder will then have to initiate a resynchronization procedure in the middle of the program, freezing the display to the last decoded picture while this process is taking place. Note that when the process of splicing in a bit stream does not take place correctly, there will also be a disruption in service at the splice back to the original bit stream. It is the recommendation of the GA that this type of splicing be strictly prohibited, since it leads to a very noticeable interruption of service.

It is important to note that the process of facilitating frequent opportunities for splicing in a program bit stream is not within the control of the transport layer of the system. The transport only provides the mechanism of implementing the splice itself. Hence decisions on determining the possible frequency of commercial insertion should also involve the people involved in the design of the source coding algorithms for applications like video and audio.

5.9. Compatibility with other Transport Systems

The GA transport system is compatible with two of the most important alternative transport systems, namely the MPEG-2 transport stream definition, and also the ATM definition being finalized for Broadband ISDN. Furthermore, since several of the CATV (e.g., Digicipher II) and DBS systems being designed are considering use of the MPEG-2 Transport layer syntax, the degree of interoperability with such deployed systems should be quite high (possibly requiring a translation if the CATV or DBS system deploys a slightly incompatible MPEG-2 variant).

5.9.1. Interoperability with MPEG-2

In the development of the GA transport specification, the intent has never been to limit the design by the scope of the MPEG-2 systems definition. The GA system is interoperable with MPEG-2 decoders since the GA Transport is currently a constrained subset of the MPEG-2 Transport syntax. The constraints are imposed for reasons of increased performance of channel acquisition, bandwidth efficiency and decoder complexity. If, in the course of future work, the MPEG-2 standard is unable to efficiently meet the requirements of the GA system, a deviation from MPEG would be in order.

The ATV system requires definition of bit streams and services beyond the compressed video and audio services. A means of identifying such bit streams is necessary in the ATV system, but is not part of the MPEG-2 definition. There is a method of encoding such a registration descriptor when a an authority to administrate registration is identified. This identification is implemented by the registration_descriptor in the PSI stream.

5.9.2. Interoperability with ATM

The GA transport packet size is selected to ease transferring these packets in a link layer that supports Asynchronous Transfer Mode (ATM) transmission. There are several methods for mapping the Transport packet into the ATM format. Three techniques are presented, although the industry may converge to a different solution than those presented here.

5.9.2.1. ATM Cell and Transport Packet Structures

Figure 5.9.1 shows the format of an ATM cell. The cell consists of two parts: a five byte header and a forty-eight byte information field. The header, primarily significant for networking purposes, consists of the following fields:

GFC a four bit Generic Flow Control field used to control the flow of traffic across the User Network Interface (UNI). Exact mechanisms for flow control are under investigation.

VPI an eight bit network Virtual Path Identifier.

VCI a sixteen bit network Virtual Circuit Identifier.

PT a three bit Payload Type (i.e., user information type ID).

CLP a one bit Cell Loss Priority flag (eligibility of the cell for discard by the network under congested conditions).

HEC an eight bit Header Error Control field for ATM header error correction

AAL ATM Adaptation Layer bytes (user specific header).

The ATM User Data Field consists of forty-eight bytes, where up to four of these bytes can be allocated to an Adaptation Layer.

Figure 5.9.2 shows the format of the Grand Alliance transport packet. A one hundred eighty-four byte packet data field (possibly including an optional and conditional adaptation field) is preceded by a four byte prefix.

5.9.2.2. Null AAL Byte ATM Cell Formation

The simplest method to form ATM cells from the Transport layer is the null AAL byte structure shown in Figure 5.9.3. The Transport packet is partitioned into forty-eight byte payloads, applied directly to the information fields of the ATM cell. The five byte ATM header is appended. Since the Transport packet length is not an integer multiple of the ATM cell payload, there will be only occasional alignment of the Transport header with the start of the ATM cell information field.

5.9.2.3. Single AAL Byte ATM Cell Formation

Alignment of the Transport packet and ATM cell is accommodated by parsing the Transport packet into forty-seven byte segments, shown in Figure 5.9.4. Four such segments will exactly encompass a Transport packet. A one byte AAL is appended, along with the five byte ATM header to fulfill the fifty-three byte ATM cell requirement. The AAL byte can carry useful information concerning the transport data within the ATM cell. It can be viewed as an adaptation field for the contained data, conveying the original position of the ATM payload within the Transport packet, for example, as well as other information. For example, ATM standards presently provide for five different AALs, such as AAL Type 1 for accommodating connection oriented constant bit rate services, and AAL Type 2 for handling connection oriented variable bit rate data services.

5.9.2.4. Dual AAL Byte ATM Cell Formation

An alternative solution to cell/packet alignment is shown in Figure 5.9.5. The transport header is discarded, and the remaining one hundred eighty-four byte payload is segmented into 46 byte increments. To these are added two AAL bytes and the five byte ATM header for each ATM packet. The idea here is that there may be a duplication in functionality in the ATM header and the link level transport header fields. A particular header field to consider for duplication of functionality is the PID. If the PID can be associated with a specific VPI and VCI used in the ATM headers of the packets carrying the data payload, and this PID mapping information can be sent to the destination terminal when the virtual path/circuit is set up (using the ATM signaling channel), it does not have to be transmitted for every GA packet. The PID can then be reconstructed at the destination (using the information transmitted at call setup), and can then be appended to the one hundred eighty-four byte payload (reconstructed from four ATM packets) to obtain the complete GA packets. Transport header information that cannot be reconstructed (e.g., adaptation field control) should be carried as a part of the two AAL bytes for each ATM cell, along with other additional information. Note that the above is only a suggested approach and does not represent a complete design.

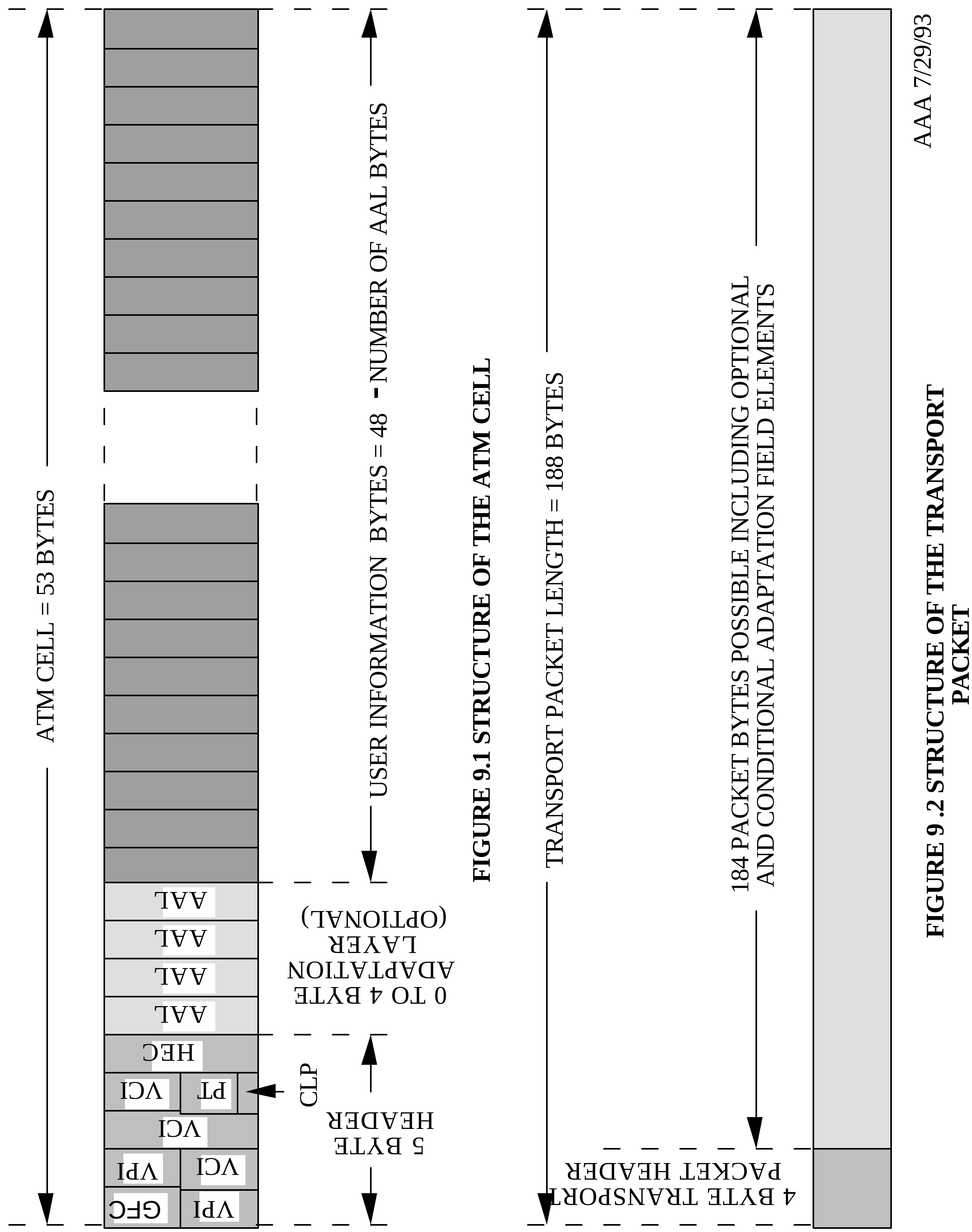


Figure 5.9.1. Structure of the ATM Cell and Figure 5.9.2. Structure of the Transport Packet

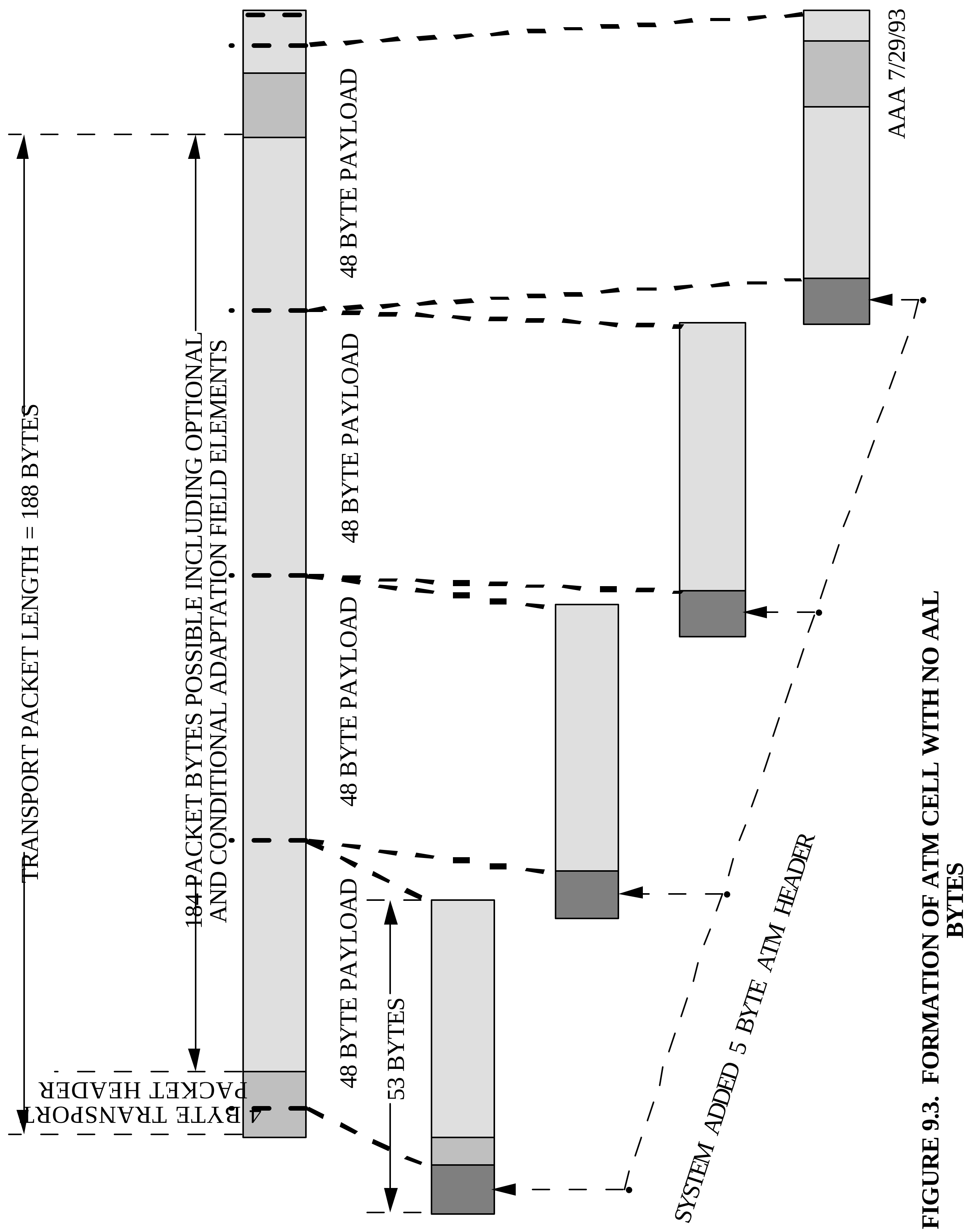


Figure 5.9.3 Formation of ATM Cell with no AAL Bytes

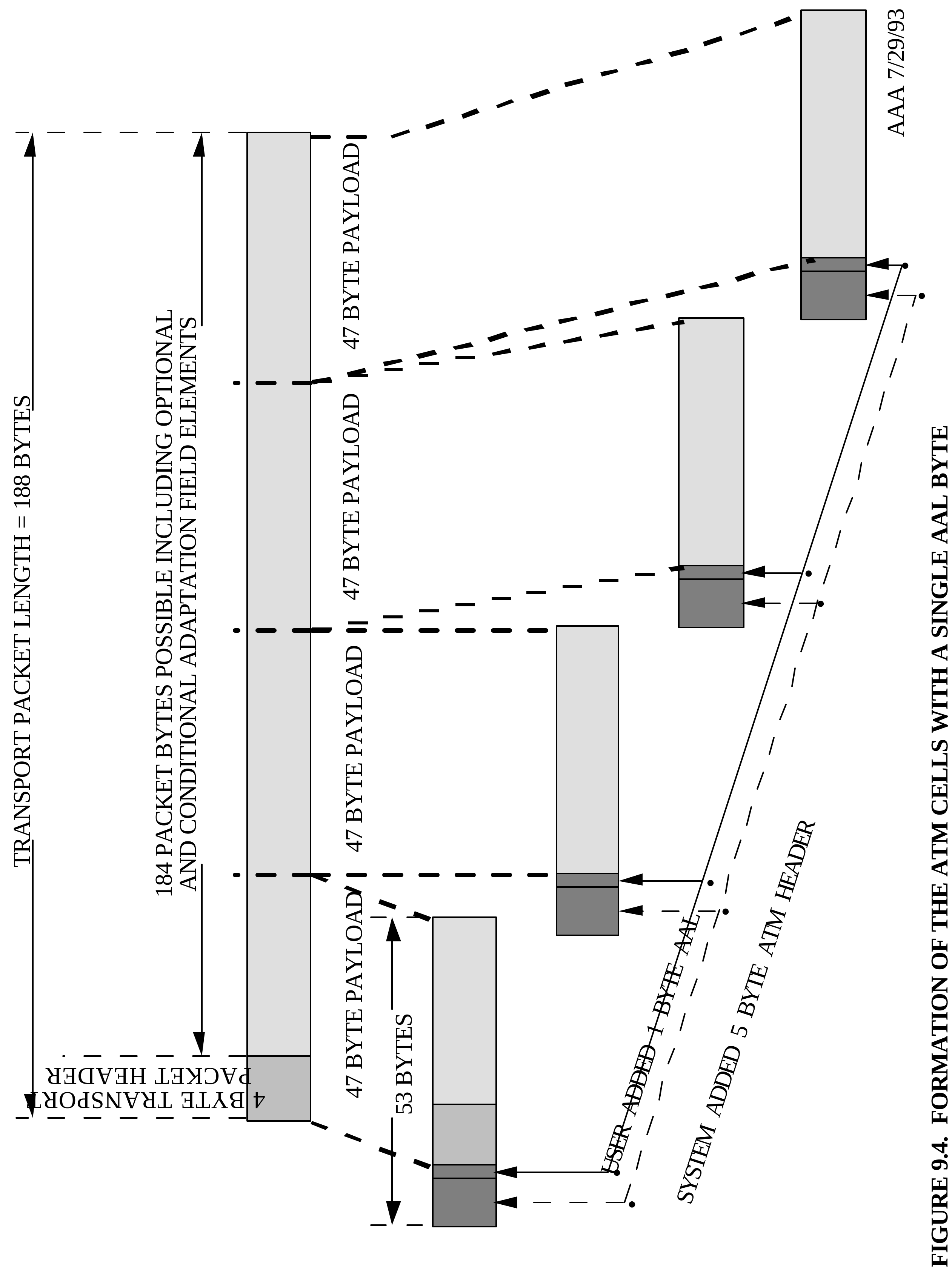


Figure 5.9.4 Formation of the ATM Cells with a single AAL Byte

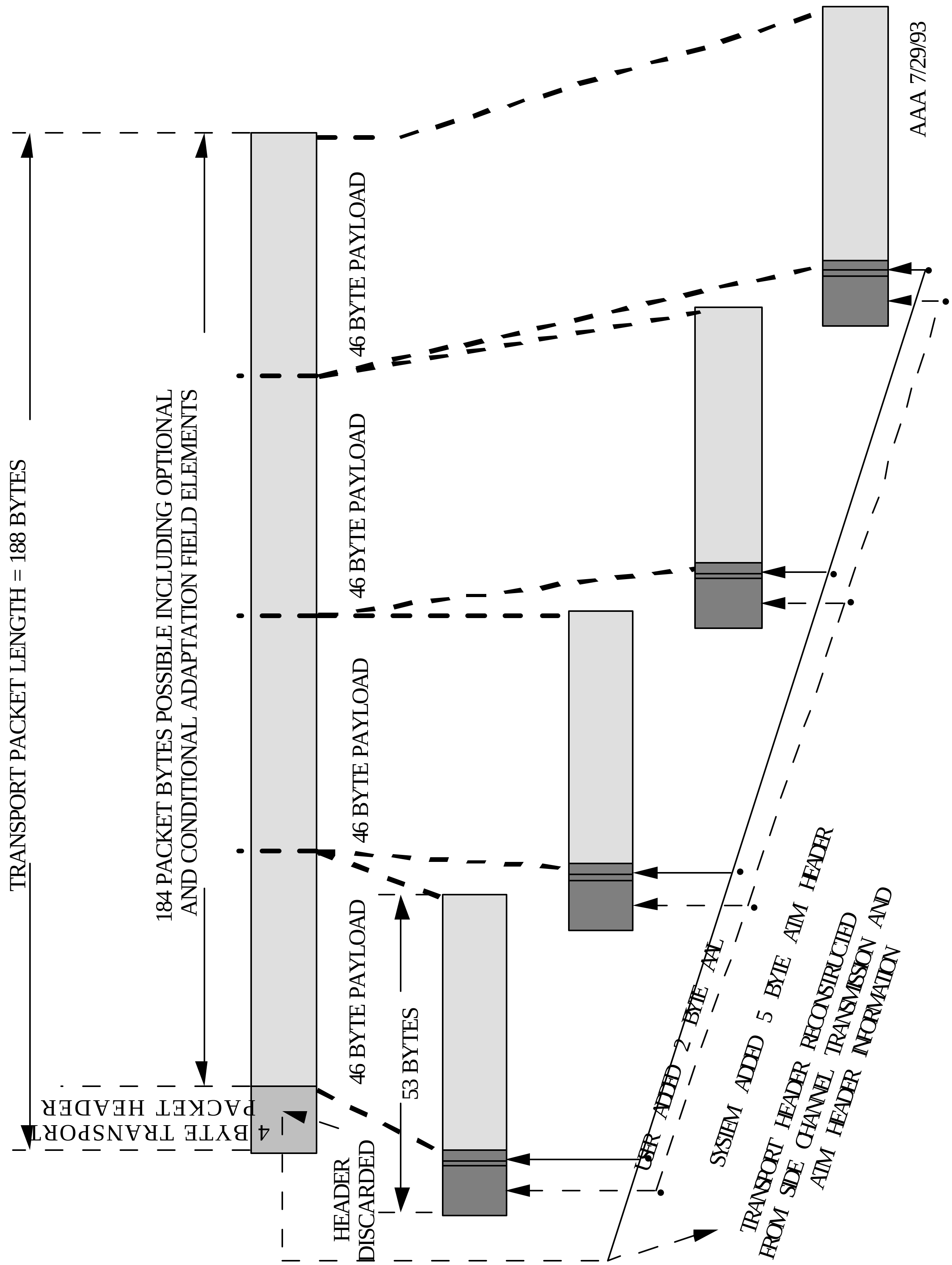


FIGURE 9.5. FORMATION OF AN ATM CELL WITH DUAL AAL BYTE

Figure 5.9.5 Formation of an ATM Cell with Dual AAL Byte

Chapter 6

TRANSMISSION SYSTEM

6.1. Introduction And System Overview (Modified 12-7-94)

The Grand Alliance (G-A) vestigial sideband (VSB) digital transmission system provides the basis for a family of transmission systems suitable for data transmission over a variety of media. This family shares the same pilot, symbol rate, data frame structure, interleaving, Reed Solomon coding and synchronization pulsed. The VSB system has two modes: a simulcast terrestrial broadcast mode, and a high data rate cable mode. The terrestrial broadcast mode, which transmits an ATV signal on all NTSC channels including currently unusable taboo NTSC channels with minimal interference to or from NTSC channels, is optimized for maximum service area, and supports one ATV signal in a 6 MHz channel. The high data rate cable mode, which trades off some robustness for twice the data rate, supports two ATV signals in one 6 MHz channel.

The VSB transmission system (both modes) take advantage of a pilot, a segment synch, and a training sequence for robust acquisition and operation. The two systems also share identical carrier, sync, and clock recovery circuits, as well as phase correctors and equalizers. Additionally, both systems use the same Reed-Solomon (R-S) code for forward error correction (FEC).

In order to maximize service area, the terrestrial broadcast mode incorporates both an NTSC rejection filter (in the receiver) and trellis coding. In contrast to the 4-VSB system tested during the first round of ATV testing, precoding at the transmitter is incorporated in the trellis code. When the NTSC rejection filter is activated in the receiver, the trellis decoder is switched to a trellis code corresponding to the encoder trellis code concatenated with the filter.

The cable mode, on the other hand, does not have as severe an environment to work in as that of the terrestrial system. Therefore, a higher data rate is transmitted in the form of more data levels (bits/symbol). For cost considerations, no trellis coding or NTSC interference rejection filters are employed.

VSB transmission inherently requires only processing the in-phase (I) channel signal. sampled at the symbol rate, thus optimizing the receiver for low cost implementation. The decoder only requires one A/D converter and a real (not complex) equalizer operating at the symbol rate of 10.76 Msamples/second.

The parameters for the two VSB transmission modes are shown in Table 1.

6.2. Terrestrial VSB System Description

6.2.1 System Information (Modified 12-7-94)

The VSB transmission system transmits data according to the data frame depicted in Fig 1. The frame is organized into segments each with 832 symbols. Each transmitted segment consists of a four symbol segment synch followed by 828 data plus FEC symbols. Each transmitted segment consists of one byte (4 symbols) of synch, 187 bytes of data, and 20 R-S parity bytes. This corresponds to a 188 byte packet consisting of 187 data bytes and one byte of synch.

The exact symbol rate is:

4.5/286 MHz x 684, which is approximately, 10.76 MHz. The first term, 4.5/286, is NTSC horizontal scan rate which is a widely known reference. The symbol rate tolerance should be no greater than $\pm$10 ppm, i.e. $\pm$01077%.

The parameters of a segment may be calculated as follows:

f_{seg} ~(10.76 Msymbols/sec / 832 symbols/seg ~(12.93 ksymbols/sec

For terrestrial broadcast mode, each segment corresponds to one R-S correction block of 207 bytes as follows:

207 bytes/block x 8 bits/byte = 3 bits/symbol x 2/3 rate x 828 symbols = 1656 bits/segment

For the high data rate cable mode, each segment corresponds to two R-S correction blocks of 207 bytes as follows:

2 blocks x 207 bytes/block x 8 bits/byte = 4 bits/symbol x 828 symbols = 3312 bits/segment

As shown in Fig 1, each data frame begins with a first data field synch segment followed by 312 data segments, a second data field synch segment, and another 312 data segment.

Except for the binary data segment and data field synchs, all other transmitted symbols are multi-level. For the terrestrial broadcast mode, 8-level symbols (3 bits/symbol) are transmitted while for the high data rate cable mode, 16-level symbols (4 bits/symbol) are used. These are called trellis coded 8-VSB and 16-VSB, respectively.

The multi-level symbols combined with the data segment and data field cinches are used to suppress-carrier modulate a single carrier. However, before transmission, most of the lower sideband is removed. The resulting spectrum is flat, except for the band edges where a root-raised cosine response results in 620 KHz transition regions which includes 11.5% excess bandwidth. The VSB transmission spectrum relative to an NTSC spectrum is shown in Fig 2. The cumulative distribution function (CDF) of the peak-to-average power ratio as measured on a low power transmitted signal with no non-linearities is plotted in Fig 3, with 99.9% of the power envelope within 6.3 dB of the average power.

At the suppressed carrier frequency of 310 KHz from the lower band edge, a small pilot is also added to the signal. The pilot is used in the VSB receiver to achieve carrier lock. Pilot power adds 0.3 dB to the total signal power but helps reduce implementation loss by more than that. With the aid of the pilot, the VSB transmission system achieves virtual theoretical performance (i.e. no implementation loss). The pilot is positioned in the vestigial sideband region of a cochannel NTSC signal and does not contribute to cochannel interference into NTSC.

The terrestrial VSB system was designed with robustness in mind. Forward error correction in the form of R-S and trellis coding, along with 1/6 data field interleaving, provide a rugged system that can endure both white noise and interference. The terrestrial VSB system can operate in a signal-to-white noise (S/N) environments of 14.9 dB. The 8-VSB, 4-state segment error probability curve in Fig 4 shows a segment error probability of 1.93 x

10^{-4}. This is equivalent to 2.5 segment errors/second which has been established by measurement as the threshold of visibility of errors. At 14.9 dB S/N a BER of $3x10^{-6}$ has been established by measurements with white noise impairment.

6.2.2 Transmitter Broadcast Mode

A functional block diagram of the VSB terrestrial broadcast transmitter is shown in Fig 5. Descriptions of each block follow.

6.2.2.1 Data Randomizer (Modified 12-7-94)

A data randomizer is used on all input data to randomize the data payload (NOT including syncs, or R-S parity bytes). This ensures random data is transmitted even when constant data is applied to the system, as might happen when a data input is disconnected.

The data randomizer exclusive OR's all the incoming data bytes with a 16-bit maximum length pseudo random sequence (PRS) locked to the data frame. The PRS is generated in a 16-bit shift register that has 9 feedback taps. Eight of the shift register outputs are selected as the fixed randomizing byte, where each bit from this byte is used to individually exclusive OR the corresponding input data bit. The generator polynomial and circuit is shown in Figure 6.

The sequence is initialized to ìF180 hexî during the segment sync interval before the first data segment. The data bits are XORed MSB to MSB ... LSB to LSB. The first data byte is XORed with CO, the second with 6D, the third with 3F.... until the parity or sync interval occurs. Then the sequence is held and parity and sync data is inserted without randomization.

6.2.2.2 Reed-Solomon Encoder (Modified 12-7-94)

The Reed-Solomon (R-S) error protection code, known for its burst noise correction capability, is a very efficient block code for its parity overhead. Since it works with bytes (8-bits), any number of bit errors can occur within a byte and not affect its error correction ability. Therefore, there is no direct proportional relationship between byte error rate and bit error rate. Only at very low error rates do the two converge, but the bit error rate is never smaller than the byte error rate for the same S/N ratio. The measure of performance for a block codes is its packet error rate versus signal-to-noise ratio or signal-to-interference ratio. Since ATV data is to be transmitted in MPEG transport packets, the most common method of describing data transmission performance in noise and interference conditions is probability of data packet error.

The R-S code used in the VSB transmission system is a t=10 (207,187) code. The data block size is 187 bytes, with 20 R-S parity bytes added at the end of the segment for error correction. A total R-S block size of 207 bytes is transmitted per data segment (equivalent to 208 bytes including synch). A t=10 R-S code, with 20 parity bytes, can correct up to 10 byte errors per block. In creating bytes from the serial bit stream the MSB is first. The parity generator polynomial and the primitive field generator polynomial and circuits are shown in Figure 7.

6.2.2.3 Interleaver (Modified 12-7-94)

Although the R-S code is particularly powerful in protecting against burst errors, the data is interleaved for further protection. Burst errors result from both impairments and trellis decoding mistakes where a single error can propagate through the trellis decoder, multiply, and become a burst error. Long trellis decoders (large K) cause correspondingly longer burst errors to be created.

The goal of the interleaver is to spread the data bytes from the same R-S block over time so that a long burst of noise or interference is necessary to encompass more than 10 data bytes and overrun the R-S error protection. Experience with NTSC terrestrial broadcasts has shown that impulse noise is often clustered in 120 Hz power line bands.

The interleaver employed in the VSB transmission system is a 52 data segment (intersegment) convolutional byte interleaver. Interleaving is provided to a depth of about 1/6 of a data field (4 msec deep). Only data bytes are interleaved. Field and segment sync are not included in the interleave process, since they are not added until later. The transport layer delivers the data bytes in burst form. Therefore, the interleaver of necessity starts and stops. The interleaver is synchronized to the first data byte of the data field. The convolutional interleaver is shown in Figure 8. Note that the first data byte of each data field is applied to Byte Shift Register Number 1 and encounters no delay. The system will tolerate an error burst of up to 193 µsec. Earlier NTSC terrestrial reception experience showed that an error burst correction capability of 47µsec would be insufficient for the case of ATV terrestrial reception in the presence of power line related impulse noise by about 4 to 1.

Intrasegment interleaving described in the next section is also performed for the benefit of the trellis coding process.

6.2.2.4 Trellis Coded Modulation (Modified 12-7-94)

The terrestrial broadcast VSB transmission mode employs a 2/3 rate (R=2/3) trellis code (with one unencoded bit which is pre-coded). That is, one input bit (LSB) is encoded into two output bits using a 1/2 rate convolution code (implemented by a feedback convolutional encoder to be compatible with the receiver comb filter and trellis decoder) while the other input (MSB) bit is precoded to complement the comb filter in the receiver. In the creation of serial bits from parallel bytes the MSB is first (7,6,5,4,3,2,1,0).The signaling waveform used with the trellis code is an 8-level (3 bit) 1-dimensional constellation. The transmitted signal is commonly referred to as 8-VSB. To minimize the complexity of interleaving and trellis decoder hardware, as well as to minimize error propagation, a relatively simple (short) 4-state trellis encoder is used. Long trellis codes, which cause longer burst errors and require more interleaving, were not found to be beneficial.

The 8-VSB trellis encoder, pre-coder, and 8-level symbol mapper are shown in Figure 9.

Although trellis codes produce improvements in signal-to-noise ratio (S/N) threshold against white noise, they do not perform well for impulse or burst noise. Besides electromechanical sources of burst noise, burst noise is also caused by NTSC cochannel interference and phase noise which can cause data-dependent cross talk. To further reduce the effects of burst errors, and to simplify the trellis decoder when the NTSC rejection filter is used, trellis code (intrasegment) interleaving is used. This uses twelve identical trellis encoders and pre-coders operating on interleaved data symbols. The code interleaving is accomplished by encoding symbols {0,12,24,36...} as one group, symbols {1,13,25,37} as a second group, symbols {2,14,26,38,...} as a third group, and so on for a total of 12 groups. The 12 groups are required since the creation of spectral nulls in the receiver requires a comb filter with delays of 12 symbols. This will be discussed further later. The trellis code and pre-coder intrasegment interleaver is shown in Figure 10 which feeds the mapper previously shown in Figure 9. Referring to Figure 10, data bytes are fed from the byte interleaver to the trellis coder and pre-coder, and they are processed as whole bytes by each of the 12 encoders. Each byte produces 4 symbols from a single encoder.

The 12-to-1 interleaving of the encoded data symbols is complicated by the fact that segment sync symbols are embedded within the data stream. The requirement imposed by the comb filter is that symbols in the final transmitted data stream coming from each of the encoders are spaced by twelve symbols. This constraint must also hold across the segment sync. To meet this specification, the output multiplexer shown in Figure 10 advances by four symbols on each segment boundary ,however, the state of the trellis encoder is not advanced. The effect is as if the encoding process is continuous, which is necessary. The data coming out of the multiplexer follows normal ordering from encoder 1 through 12 for the first segment of the frame, but on the second segment the order changes and symbols are read from encoders 5 through 12, and then 1 through 4. The third segment reads from encoder 9 through 12 and then 1 through 8. This three-segment pattern repeats through the 312 data segments of the frame. Table 2 shows the interleaving sequence for the first three data segments of the frame. After the segment sync is inserted, the ordering of the data symbols is such that symbols from each encoder occur at a spacing of twelve symbols.

Figure 11 shows the encoder implementation that has been used. This encoder handles not only the encoding of the data stream, but it also produces a correctly interleaved data stream according to the previous discussion. The encoder operates at the symbol rate. It accepts data bits in pairs (X2 and X1) at the symbol rate with gaps in the stream that correspond to positions where the segment sync would otherwise be inserted. The three multiplexers are controlled such that the lower input is selected only during the four segment sync symbols. This has the effect of re-aligning the output data stream such that the twelve symbol spacing requirement is met. The sync insertion block overwrites the symbols that occur at the segment sync location, which correspond to the gaps in the data stream input to the encoder.

Segment	Block 1	Block 2	...	Block 69
1	D1 D2 D3 ... D12	D1 D2 D3 ... D12	...	D1 D2 D3 ... D12
2	D5 D6 D7 ... D4	D5 D6 D7 ... D4	...	D5 D6 D7 ... D4
3	D9 D10 D11 ... D8	D9 D10 D11 ... D8	...	D9 D10 D11 ... D8

Table 2. - Sequence of interleaving of encoded data symbols.

The theoretical performance of the concatenated trellis and R-S code used in the terrestrial broadcast VSB system was shown earlier in Fig 4. The combination of short trellis codes and code interleaving is a superior error correction technique for use in channels with white gaussian noise and Rayleigh distribution fading.

6.2.2.5 Synchronization Information (Modified 12-7-94)

Synchronization information is added to the digital data signal in order to facilitate packet and symbol clock acquisition and phase-lock during extreme noise and interference conditions. The encoded trellis data is passed through a multiplexer that inserts the various synchronization signals (data segment sync & data field sync).

A two level (binary) 4-symbol Data Segment Sync is inserted into the 8-level digital data stream at the beginning of each data segment. (The MPEG sync byte is replaced by data segment sync in the VSB transmission system and re-inserted in the receiver for delivery to the MPEG transport system. Note! The MPEG sync and the data segment sync have a one-to-one correspondence only for 2 bits/symbol, i.e., 4-VSB or 2/3 rate coded 8-VSB.) The data segment sync embedded in random data is illustrated in Fig 12. A complete segment consists of 832 symbols: 4 sync symbols, and 828 data plus parity symbols. The segment sync is binary (2-level) in order to make packet and clock recovery rugged. The levels selected (±5) for segment sync insure robustness, but do not cause undue interference into cochannel NTSC signals. They also have an average value of zero so that when the DC pilot is added, they will not alter the desired value of pilot. The same sync pattern occurs regularly at 77.3 SYMBOL 109 \f "Symbol" sec intervals, and is the only signals repeating at this rate. The periodic recurrence of these four symbols makes possible their reliable detection at the receiver under severe noise and/or interference. Unlike the data, the four data segment sync symbols are not Reed-Solomon or trellis encoded, nor are they interleaved.

The data is not only divided into data segments, but also into data fields. The data field consists of 313 segments. Each data field starts with one complete data segment of Data Field Sync, as shown in Fig 13.

Each symbol represents one bit of data. The data levels are equivalent to a 2 VSB signal. The 832 symbols in this segment are defined below. Refer to Figure 13.

SYNC

This corresponds to segment sync and is defined as 1001.

PN511

This pseudo random sequence is defined as X9 + X7 + X6 + X4 + X3 + X + 1 with a preload value of 010000000. The sequence is:

0000000101111111110010101010111001100110100010001001111000011101

0111110100110101001110110011101001000101100011110010000101000111

1100111101010001010011000011000100000100001111110000010101000000

1100111111101110101010010110011000110111011110110100101001001110

0111000101110100001101001111101100010101101111001101101011101101

1001011011011100100100101110001110010111101000110101100001001101

1111000100101011110001100101000010001100000111101111110101101010

110010011001000111011100001011010000011011000000100100000001110

PN63

This pseudo random sequence is repeated three times. It is defined as X6 + X + 1 with a preload value of 100111. The middle PN63 is inverted on every other field sync. This aids in avoiding effects of hardware DC offsets in the receiver. The sequence is:

111001001011011101100110101011111100000100001100010100111101000

Note: The generators for the PN63 and PN511 sequences are shown in Figure 14.

VSB MODE

These 24 bits determine the VSB mode for the data in the frame. The first two bytes are reserved. The suggested fill pattern is 0000111100001111. The next byte is defined as:

PABCPABC

where P is the even parity bit, the MSB of the byte, and A, B, C are the actual mode bits.

PABC	
0000	RESERVED
1001	RESERVED
1010	RESERVED
0011	RESERVED
1100	16 VSB CABLE (GRAND ALLIANCE)
0101	8 VSB TRELLIS CODED TERRESTRIAL* (GRAND ALLIANCE)
0110	RESERVED
1111	RESERVED

* In this mode, the preceding 16 bits are defined as 0000PABCPABC1111 to satisfy the post comb filter.

RSVD

The last 104 bits are reserved space. It is suggested that this be filled with a continuation of the PN63 sequence. In the 8 VSB Trellis mode, 92 bits are reserved followed by the 12 symbol definition below.

PRECODE

The last 12 symbols of the segment have a specific requirement for the 8 VSB Trellis mode. In this mode, these 12 symbols must correspond to the last 12 symbols of the previous segment. This is necessary to preload the trellis decoder and the post comb filter.

All of the above symbol sequences are pre-loaded before the beginning of the data field sync. The 511PN sequence is useful for long ghosts and the 63PN sequence is useful for short ghosts.

The Data Field Sync serves five purposes. First, it provides a means to determine the beginning of each data field. Second, it is also used by the equalizer in the ATV receiver as a training reference signal to remove intersymbol and other interferences. Third, it allows the receiver to determine whether the interference rejection filter (12 symbol subtractive comb) should be used. Fourth, it is used for system diagnostic measurements, such as signal-to-noise and channel response. Fifth, the phase tracker in the receiver uses the field sync to reset its circuitry and determine its loop parameters. Just like the data segment sync, the Data Field Sync is not Reed-Solomon or trellis encoded, nor is it interleaved.

6.2.2.6 Pilot Insertion (Modified 12-7-94)

A rugged system must be able to acquire a signal and maintain lock in the presence of very heavy noise and interference. A small pilot added to the suppressed carrier RF data signal is adequate to allow robust carrier recovery in the receiver during these extreme conditions.

A small (digital) DC level (1.25) is added to every symbol (data and syncs) of the digital baseband data plus sync signal (±1, ±3, ±5, ±7). This has the effect of adding a small in-phase pilot to the data signal. Digital addition of the pilot (at baseband) provides a highly stable and accurate pilot. The frequency of the pilot will be the same as the suppressed carrier frequency. Since the data is essentially guaranteed to be random over long intervals, all data states are equally probable. Using the above eight values for the random data symbols, the total average data power of the 8-level data signal is 21. After adding the pilot, the total average signal power is 22.56. Thus, the pilot represents an increased transmitted power of 0.3 dB and is 11.3dB below data signal power. In the interference-limited environment, the pilot is not a contributing factor. Also, the pilot power has no significant effect on the ATV transmitter hardware (e.g. power dissipation or peak-to-average power ratio).

6.2.2.7 Pre-Equalizer Filter (Modified 12-7-94)

A pre-equalizer filter is available for use in over-the-air broadcasts, where the high power transmitter may have significant in-band ripple or roll off at band edges. This linear distortion can be detected by an equalizer in a reference demodulator ("ideal" receiver) located at the transmitter site that is receiving a small sample portion of the antenna signal feed provided by a directional coupler which is recommended to be located at the sending end of the antenna feed transmission line. (from a "sample"). The reference demodulator equalizer tap weights can be transferred into the transmitter pre-equalizer for pre- correction of transmitter linear distortion.

A suitable The pre-equalizer is an 80 tap, feedforward transversal filter. The taps are symbol spaced (93nsec) with the main tap being approximately at the center, giving approximately +/-3.7(sec correction range. It operates on the I channel data signal (there is no Q channel data in the transmitter), and shapes the frequency spectrum of the IF signal so that there is a flat in band spectrum at the output of the high power transmitter that feeds the antenna for transmission. There is no effect on the out-of-band spectrum of the transmitted signal.

6.2.2.8 VSB Modulator (Modified 12-7-94)

The VSB modulator receives the 10.76 Msymbols/sec, 8-level trellis composite data signal (pilot plus syncs added). For minimal intersymbol interference, the data signal must be properly shaped (filtered) before it is transmitted over the 6 MHz channel. A linear phase raised-cosine Nyquist filter is employed in the concatenated transmitter and receiver, as shown in Fig 15. The system filter response is essentially flat across the entire band, except for the transition regions at each end of the band. Due to the vestigial-sideband nature of the transmitted signal, the same skirt selectivity on both sides is not required, although it is has been so implemented. The optimum system arrangement is to divide the roll off equally between the transmitter and receiver filters. Therefore, root-raised cosine filters are used. Fig 2 illustrates the ATV transmitter spectrum in comparison with a typical NTSC spectrum.

The transmitter VSB filtering may be implemented by complex-filtering the baseband data signal, creating precision-filtered and stable in- phase and quadrature-phase modulation signals. This filtering process provides the root-raised cosine Nyquist filtering as well as the sin x/x compensation for the D/A converters. The orthogonal baseband signals are converted to analog form (D/A converters) and then modulated on quadrature IF carriers to create the vestigial sideband IF signal by sideband cancellation (phasing method). The nominal frequency of the IF carrier (and small in-phase pilot) is 46.69 MHz, which is equal to the IF center frequency (44.000 MHz) plus the symbol rate divided by 4 (10.762 MHz/4=2.6905 MHz). Additional adjacent channel suppression (beyond that achieved by sideband cancellation) is may be performed by a linear phase, flat amplitude response SAW filter. Adjacent channel energy spillage at the IF output is at least 57 dB down from the desired ATV signal power in the recent implementation. Other implementations for VSB filtering are possible which may include the prefilter of the previous section. There is no mandatory filter response. A portion of the in-band filtering and equalizing task may be assigned to the receiver as long as the threshold performance is not significantly degraded. These limits are under study. An acceptable out-of-band spectrum level has yet to be recommended.

6.2.2.9 Upconverter And RF Carrier Frequency Offsets (Modified 12-7-94)

Modern NTSC TV transmitters use a two-step modulation process. The first step usually is modulation of the data onto an IF carrier, which is the same frequency for all channels, followed by translation to the desired RF channel. The VSB transmitter applies this same two-step modulation process. The RF upconverter translates the filtered flat IF data signal spectrum to the desired RF channel. For the same coverage as an NTSC transmitter, the average power of the ATV signal is 12 dB less than the NTSC peak sync power.

The frequency of the RF upconverter oscillator in ATV terrestrial broadcasts will typically be the same as that used for NTSC (except for NTSC offsets). The phase noise level should be better than -104dBc @ 20 KHz. However, in extreme cochannel situations, the ATV system is designed to take advantage of precise RF carrier frequency offsets with respect to the NTSC cochannel carrier. Since the VSB data signal sends repetitive synchronizing information (segment syncs), precise offset causes NTSC cochannel carrier interference into the VSB receiver to phase alternate from sync to sync. The VSB receiver circuits average successive syncs to cancel the interference and make data segment sync detection more reliable.

For ATV cochannel interference into NTSC, the interference is noise-like and does not change with precise offset. Even the ATV pilot interference into NTSC does not benefit from precise frequency offset because it is so small (11.3 dB below the data power) and falls far down the Nyquist slope (20 dB or more) of NTSC receivers.

Although it might be postulated that an ATV transmitter can be located so that receivers experience equal interference from two worst-case cochannel NTSC stations (e.g. three-way triangle), such a situation is so unlikely that the ATV received signal is assumed to have only one dominant NTSC cochannel. The ATV cochannel pilot should be offset in the RF upconverter from the dominant NTSC picture carrier by an odd multiple of half the data segment rate. An optimum offset between the pilot frequency and that of the cochannel NTSC picture carrier has yet to be established for the 832 symbol segment case. A consequential spectrum shift of the VSB signal into the upper adjacent channel is required. A first proposal is illustrated (Fig 22) and further described later in the section on the receiver NTSC rejection filter. An additional offset of 0, +10 KHz, or -10 KHz is required to track the principal NTSC interferer.

For ATV to ATV cochannel interference, precise carrier offset prevents possible misconvergence of the adaptive equalizer. If perchance the two ATV Data Field Sync signals should fall within the same data segment time, the adaptive equalizer could misinterpret the interference as a ghost. To prevent this, a carrier offset of $f_{seg}/2$ = 6.47 KHz is proposed for close ATV-to-ATV cochannel situations. This causes the interference to have no effect in the adaptive equalizer. Carrier frequency assignments for ATV stations which include NTSC interferer frequency offsets and ATV-to-ATV interferer offsets probably ought to be handled by an allocation table, but frequency coordination might also be desirable in some extreme cases.

6.2.3 Receiver Broadcast Mode

Fig 16 shows the receiver block diagram of the VSB terrestrial broadcast transmission system. Descriptions of each block follow.

6.2.3.1 Tuner (Modified 12-7-94)

The tuner, illustrated in Fig 17, as implemented in the prototype submitted for test receives the 6 MHz ATV signal (UHF or VHF) from the antenna. It is a high-side injection double-conversion type with a first IF frequency of 920 MHz. This puts the image frequencies above 1 GHz, making them easy to reject by a fixed front end filter. This selection of first IF frequency is high enough so that the input bandpass filter selectivity prevents the local oscillator (978-1723 MHz) from leaking out the tuner front end and interfering with UHF channels, yet it is low enough for second harmonics of UHF channels (470-806 MHz) to fall above the first IF bandpass. Harmonics of cable channels could possibly occur in the first IF passband but are not a real problem because of the relatively flat spectrum (within 10 dB) and small signal levels (-28 dBm or less) used in cable systems.

The tuner input has a bandpass filter that limits the frequency range to 50-810 MHz, rejecting all other non-television signals that may fall within the tuner's image frequency range (beyond 920 MHz). In addition, a broadband tracking filter rejects other television signals, especially those much larger in signal power than the desired ATV signal power. This tracking filter is not narrow, nor is it critically tuned, as is the case of present day NTSC tuners that must reject image signals only 90 MHz away from the desired channel. Minimal channel tilt, if any, exists due to this tracking filter.

A 10 dB gain, wideband RF amplifier increases the signal level into the first mixer, and is the dominant determining factor of receiver noise figure (7-9 dB over entire VHF, UHF, and cable bands). The first mixer, a highly linear double-balanced design to minimize even harmonic generation, is driven by a synthesized low phase noise local oscillator (LO) above the first IF frequency (high-side injection). Both the channel tuning (first LO) and broadband tracking filtering (input bandpass filter) are controlled by microprocessor. The tuner is capable of tuning the entire VHF and UHF broadcast bands as well as all standard, IRC, and HRC cable bands.

The mixer is followed by an LC filter in tandem with a narrow 920 MHz bandpass ceramic resonator filter. The LC filter provides selectivity against the harmonic and subharmonic spurious responses of the ceramic resonators. The 920 MHz ceramic resonator bandpass filter has a -1 dB bandwidth of about 6 MHz. A 920 MHz IF amplifier is placed between the two filters. Delayed AGC of the first IF signal is applied immediately following the first LC filter. The 30 dB range AGC circuit protects the remaining active stages from large signal overload.

The second mixer is driven by the second LO, which is an 876 MHz voltage-controlled SAW oscillator. It is controlled by the frequency and phase-locked loop (FPLL) synchronous detector. The second mixer, whose output is the desired 44 MHz second IF frequency, drives a constant gain 44 MHz amplifier. The output of the tuner feeds the IF SAW filter and synchronous detection circuitry.

The tuner is made out of standard consumer electronic components, and is housed in a stamped metal enclosure.

6.2.3.2 Channel Filtering And VSB Carrier Recovery

Carrier recovery is performed on the small pilot carrier by an FPLL circuit, illustrated in Fig 18. The first LO is synthesized by a PLL and controlled by a microprocessor. The third LO is a fixed reference oscillator. Any frequency drift or deviation from nominal has to be compensated in the second LO. Control for the second LO comes from the FPLL synchronous detector, which integrally contains both a frequency loop and a phase-locked loop in one circuit. The frequency loop provides a wide frequency pull-in range of ±100 KHz while the phase-locked loop has a narrow bandwidth (less than 2 KHz).

During frequency acquisition, the frequency loop uses both the in-phase (I) and quadrature-phase (Q) pilot signals. All other data processing circuits in the receiver use only the I-channel signal. Prior to phase-lock, as is the condition after a channel change, the automatic frequency control (AFC) lowpass filter acts on the beat signal created by the frequency difference between the VCO and the incoming pilot. The high frequency data (as well as noise and interference) is mostly rejected by the AFC filter, leaving only the pilot beat frequency. After limiting this pilot beat signal to a constant amplitude (±1) square wave, and using it to multiply the quadrature signal, a traditional bipolar S-curve AFC characteristic is obtained. The polarity of the S-curve error signal depends upon whether the VCO frequency is above or below the incoming IF signal. Filtered and integrated by the automatic phase control (APC) lowpass filter, this DC signal adjusts the tuner's second LO to reduce the frequency difference.

When the frequency difference comes close to zero, the APC loop takes over and phase-locks the incoming IF signal to the third LO. This is a normal phase-locked loop circuit, with the exception that it is bi-phase stable. However, the correct phase-lock polarity is determined by forcing the polarity of the pilot to be equal to the known transmitted positive polarity. Once locked, the detected pilot signal is constant, the limiter output feeding the third multiplier is at a constant +1, and only the phase-locked loop is active (frequency loop automatically disabled). The APC lowpass filter is wide enough to reliably allow ±100 KHz frequency pull-in, yet narrow enough to consistently reject all strong white noise (including data) and NTSC cochannel interference signals. The PLL has a bandwidth that is narrow enough to reject most of the AM and PM generated by the data, yet is wide enough to track out any phase noise on the signal (and, hence, on the pilot) out to about 2 KHz. Tracking out low frequency phase noise (as well as low frequency FM components) allows the phase tracking loop, discussed later, to be more effective.

The prototype receiver can acquire a signal and maintain lock at a signal-to-noise of 0 dB or less, and in the presence of heavy interference.

6.2.3.3 Segment Sync And Symbol Clock Recovery

The repetitive data segment syncs are detected from among the synchronously detected random data by a narrow bandwidth filter. From the data segment syncs, a properly phased 10.76 MHz symbol clock is created along with a coherent AGC control signal. A block diagram of this circuit is shown in Fig 19.

The 10.76 Msymbols/sec I-channel composite baseband data signal (syncs and data) from the synchronous detector is converted by an A/D converter for digital processing. Traditional analog data eyes can be viewed after synchronous detection. However, after conversion to a digital signal, the data eyes cannot be seen due to the sampling process. A PLL is used to derive a clean 10.76 MHz symbol clock for the receiver.

With the PLL free-running, the data segment sync detector containing a 4-symbol sync correlator looks for the two level syncs occurring at the specified repetition rate. The repetitive segment sync is detected while the random data is not, enabling the PLL to lock on the sampled sync from the A/D converter, and achieve data symbol clock synchronization. Upon reaching a predefined level of confidence (using a confidence counter) that the segment sync has been found, subsequent receiver loops are enabled.

Data segment sync detection and clock recovery both work reliably at signal-to-noise

ratios of 0 dB or less, and in the presence of heavy interference.

6.2.3.4 Non-Coherent And Coherent AGC

Prior to carrier and clock synchronization, non-coherent automatic gain control (AGC) is performed whenever any signal (locked or unlocked signal, or noise/interference) overruns the A/D converter. The IF and RF gains are reduced accordingly, with the appropriate AGC "delay" applied.

When data segment syncs are detected, coherent AGC occurs using the measured segment sync amplitudes. The amplitude of the bipolar syncs, relative to the discrete levels of the random data, is determined in the transmitter. Once the syncs are detected in the receiver, they are compared to a reference value, with the difference (error) integrated. The integrator output then controls the IF and "delayed" RF gains, forcing them to whatever values provide the correct sync amplitudes.

6.2.3.5 Data Field Synchronization

Data Field Sync detection, shown in Fig 21, is achieved by comparing each received data segment from the A/D converter (after interference rejection filtering to minimize cochannel interference), to ideal field #1 and field #2 reference signals in the receiver. Oversampling of the field sync is NOT necessary since a precision data segment and symbol clock has already been reliably created by the clock recovery circuit. Therefore, the field sync recovery circuit knows exactly where a valid field sync correlation should occur within each data segment, and only needs to perform a symbol by symbol difference. Upon reaching a predetermined level of confidence (using a confidence counter) that field syncs have been detected on given data segments, the Data Field Sync signal becomes available for use by subsequent circuits. The polarity of the three alternating 63 bit pseudo random (PN) sequences determine whether field 1 or field 2 is detected.

This procedure makes field sync detection robust, even in heavy noise, interference, or ghost conditions. Field sync recovery can reliably occur at signal-to-noise ratios of 0 dB or less, and in the presence of heavy interference.

6.2.3.6 Interference Rejection Filter (Modified 12-7-94)

The interference rejection properties of the VSB transmission system are based on the frequency location of the principal components of the NTSC cochannel interfering signal within the 6 MHz TV channel and the periodic nulls of a VSB receiver baseband comb filter.

Fig 22a shows the location and approximate magnitude of the three principal NTSC components: (1) the visual carrier (V) located 1.25 MHz from the lower band edge, (2) the chrominance subcarrier (C) located 3.58 MHz higher than the visual carrier frequency, and (3) the aural carrier (A) located 4.5 MHz higher than the visual carrier frequency.

The NTSC interference rejection filter (comb) is a one tap linear feed-forward filter, as shown in Fig 23. Fig 22b shows the frequency response of the comb filter, which provides periodic spectral nulls spaced 57* fh (10.762 MHz/12, or 896.85 KHz) apart. There are 7 nulls within the 6 MHz channel. The NTSC visual carrier frequency falls close to the second null from the lower band edge. The 6th null from the lower band edge is correctly placed for the NTSC chrominance subcarrier, and the 7th null from the lower band edge is near the NTSC aural carrier.

Comparing Fig 22a and Fig 22b shows that the visual carrier falls 2.1 KHz below the second comb filter null, the chroma subcarrier falls near at the 6th null, and the aural carrier falls 13.6 KHz above the 7th null. (Note, the aural carrier is at least 7 dB below its visual carrier).

The comb filter, while providing rejection of steady-state signals located at the null frequencies, has a finite response time of 12 symbols (1.115 SYMBOL 109 \f "Symbol" secs). Thus, if the NTSC interfering signal has a sudden step in carrier level (low to high or high to low), one cycle of the zero beat frequency (offset) between the ATV and NTSC carrier frequencies will pass through the comb filter at an amplitude proportional to the NTSC step size as instantaneous interference. Examples of such steps of NTSC carrier are: leading and trailing edge of sync (40 IRE units). If the desired to undesired (D/U) signal power ratio is large enough, data slicing errors will occur. However, interleaving will spread the interference and will make it easier for the Reed-Solomon code to correct them (R-S can correct up to 10 byte errors/segment).

Although the comb filter reduces the NTSC interference, the data is also modified. The 7 data eyes (8 levels) are converted to 14 data eyes (15 levels). This doubling of the eyes is caused by the partial response process which is a special case of intersymbol interference that does not close the data eye but creates double the number of eyes of the same magnitude. The modified data signal can be properly decoded by the trellis decoder, and will be described in later sections. Note, because of time sampling, only the maximum data eye value is seen after A/D conversion.

The detail at the band edges for the overall channel is shown in Fig 22c and Fig 22d. Fig 22d shows that the frequency relationship (57 19/22f_H between NTSC visual carrier and ATV carrier) requires a shift in the ATV spectrum with respect to the nominal channel. The shift equals +45.8 KHz, or about +0.76%. This is slightly higher than currently applied channel offsets and reaches into the upper adjacent channel at a level of about -40 dB. If that is another ATV channel, its spectrum is also shifted upward, therefore no spectral overlapping occurs. If it is an NTSC channel, the shift is below the (RF equivalent of the) Nyquist slope of an NTSC receiver where there is high attenuation, and it is slightly above its customary lower adjacent channel sound trap. No adverse effects of the shift have been found nor are they foreseen. An additional shift of the ATV spectrum is used in order to track the dominant NTSC interferer which may be assigned an offset of -10 KHz, 0KHz, or 10 KHz.

NTSC interference can be detected by the circuit shown in Fig 23, where the signal-to-interference plus noise ratio of the binary Data Field Sync is measured at the input and output of the comb filter, and compared to each other. This is accomplished by creating two error signals. The first is created by comparing the received signal with a stored reference of the field sync. The second is created by comparing the rejection filter output with a combed version of the internally stored reference field sync. The errors are squared and integrated. After a predetermined level of confidence is achieved, the path with the largest signal-to-noise ratio (lowest interference energy) is switched in and out of the system automatically.

There is a reason to not leave the rejection comb filter switched in all the time. The comb filter, while providing needed cochannel interference benefits, degrades white noise performance by 3 dB. This is due to the fact that the filter output is the subtraction of two full gain paths, and since white noise is uncorrelated from symbol to symbol, the noise power doubles. There is an additional 0.3 dB degradation due to the 12 symbol differential coding. See below under Trellis Decoding. If little or no NTSC interference is present, the comb filter is automatically switched out of the data path. When the NTSC service is phased out, the comb filter is omitted from ATV receivers.

6.2.3.7 Channel Equalizer (Modified 12-7-94)

The equalizer/ghost canceller compensates for linear channel distortions, such as tilt and ghosts. These distortions can come from the transmission channel or from imperfect components within the receiver.

The equalizer uses a Least-Mean-Square (LMS) algorithm and can adapt on the transmitted binary Training Sequence as well as on the random data. The LMS algorithm computes how to adjust the filter taps in order to reduce the error present at the output of the equalizer. It does this by generating an estimate of the error present in the output signal. This error signal is used to compute a cross-correlation with various delayed data signals. These correlations correspond to the adjustment that needs to be made for each tap to reduce the error at the output.

The equalizer algorithm can achieve equalization through three means: It can adapt on the binary training sequence; It can adapt on data symbols throughout the frame when the eyes are open; or, it can adapt on data when the eyes are closed (blind equalization). The principle difference among these three methods is how the error estimate is generated.

For adapting on the training sequence, the training signal presents a fixed data pattern in the data stream. Since the data pattern is known, the exact error is generated by subtracting the training sequence from the output.

The training sequence alone, however, may not be enough to track dynamic ghosts since these require tap adjustments more often than the training sequence is transmitted. Therefore, once equalization is achieved, the equalizer can switch to adapting on data symbols throughout the frame, and produce an accurate error estimate by slicing the data with an 8-level slicer and subtracting it from the output signal.

For fast dynamic ghosts (e.g., airplane flutter) it is necessary to use a blind equalization mode to aid in acquisition of the signal. Blind equalization models the multilevel signal as binary data signal plus noise, and the equalizer produces the error estimate by detecting the sign of the output signal and subtracting a (scaled) binary signal from the output to generate the error estimate.

To perform the LMS algorithm, the error estimate (produced using the training sequence, 8-level slicer, or the binary slicer) is multiplied by delayed copies of the signal. The delay depends upon which tap of the filter is being updated. This multiplication produces a cross-correction between the error signal and the data signal. The size of the correlation corresponds to the amplitude of the residual ghost present at the output of the equalizer and indicates how to adjust the tap to reduce the error at the output.

A block diagram of the equalizer is shown in Figure 24. The DC bias of the input signal is first removed by subtraction. The DC may be caused by circuit offsets, non-linearities, or shifts in the pilot caused by ghosts. The DC offset is tracked by measuring the DC value of the training signal.

The equalizer filter consists of two parts: a 64 tap feed forward transversal filter followed by a 192 tap decision feedback filter. The equalizer operates at the 10.762 MHz symbol rate (T-sampled equalizer).

The output of the forward filter and feedback filter are summed to produce the output. This output is sliced by either an 8-level slicer (15 level slicer when the comb filter is used) or a binary slicer depending upon whether the data eyes are open or not. (As said in the previous section on interference filtering, the comb filter does not close the data eyes but creates twice as many of the same magnitude). This sliced signal has the training signal and segment syncs reinserted since these are fixed patterns of the signal. The resultant signal is fed into the feedback filter, and subtracted from the output signal to produce the error estimate. The error estimate is correlated with the input signal (for the forward filter), or by the output signal (for the feedback filter). This correlation is scaled by a step size parameter, (, and used to adjust the value of the tap. The delay setting of the adjustable delays are controlled according to the index of the filter tap that is being adjusted.

6.2.3.8 Phase Tracking Loop

The phase tracking loop is an additional decision feedback loop which further tracks out phase noise which has not been removed by the IF PLL operating on the pilot. Thus, phase noise is tracked out by not just one loop, but two concatenated loops. Because the system is already frequency-locked to the pilot by the IF PLL (independent of the data), the phase tracking loop bandwidth is maximized for phase tracking by using a first order loop. Higher order loops, which are needed for frequency tracking, do not perform phase tracking as well as first order loops. Therefore, they are not used in the VSB system.

A block diagram of the phase tracking loop is shown in Fig 25. The output of the real equalizer operating on the I signal is first gain controlled by a multiplier and then fed into a filter which recreates an approximation of the Q signal. This is possible because of the VSB transmission method, where the I and Q components are related by a filter function which is almost a Hilbert transform. The complexity of this filter is minor since it is a finite impulse response (FIR) filter with fixed anti-symmetric coefficients and with every other coefficient equal to zero. In addition, many filter coefficients are related by powers of two, thus simplifying the hardware design.

These I and Q signals are then fed into a de-rotator (complex multiplier), which is used to remove the phase noise. The amount of de-rotation is controlled by decision feedback of the data taken from the output of the de-rotator. Since the phase tracker is operating on the 10.76 Msymbol/sec data, the bandwidth of the phase tracking loop is fairly large, approximately 60 KHz. The gain multiplier is also controlled with decision feedback.

6.2.3.9 Trellis Decoder (Modified 12-7-94)

To help protect the trellis decoder against short burst interference, such as impulse noise or NTSC cochannel interference, 12 symbol code intrasegment interleaving is employed in the transmitter. As shown in Fig 26, the receiver uses 12 trellis decoders in parallel, where each trellis decoder sees every 12th symbol. This code interleaving has all the same burst noise benefits of a 12 symbol interleaver, but also minimizes the resulting code expansion (and hardware) when the NTSC rejection comb filter is active.

The trellis decoder performs the task of slicing and convolutional decoding. It has two modes; one when the NTSC rejection filter is used to minimize NTSC cochannel, and the other when it is not used. This is illustrated in Fig 28. The insertion of the NTSC rejection filter is determined automatically (before the equalizer), with this information passed to the trellis decoder. When there is little or no NTSC cochannel interference, the NTSC rejection filter is not used, and an optimal trellis decoder is used to decode the 4-state trellis-encoded data. Serial bits are re-created in the same order in which they were created in the encoder.

In the presence of significant NTSC cochannel interference, when the NTSC rejection filter (12 symbol, feedforward subtractive comb) is employed, a trellis decoder optimized for this partial response channel is used. This optimal code requires 8 states. This is necessary since the NTSC rejection filter, which has memory, represents another state machine seen at the input of the trellis decoder. In order to minimize the expansion of trellis states, two measures are taken: (1) special design of the trellis code, and (2) twelve-to-one interleaving of the trellis encoding. The interleaving, which corresponds exactly to the 12 symbol delay in the NTSC rejection filter, makes it so that each trellis decoder only sees a one-symbol delay NTSC rejection filter. By minimizing the delay stages seen by each trellis decoder, the expansion of states is also minimized. Only a 3.5 dB penalty in white noise performance is paid as the price for having good NTSC cochannel performance. The additional 0.5 dB beyond the 3 dB comb filter noise threshold degradation is due to the 12 symbol differential coding.

The presence of the segment sync character in the data stream passed through the comb filter presents a complication which must be dealt with because segment sync is not trellis encoded or pre-coded. Figure 27 shows the technique that has been used. It shows the receiver

processing that is performed when the comb filter is present in the receiver. The multiplexer in the Segment Sync Removal block is normally in the upper position. This presents data that has been filtered by the comb to the trellis decoder. However, because of the presence of the sync character in the data stream, the multiplexer selects its lower input during the four symbols that occur twelve symbols after the segment sync. The effect of this sync removal is to present to the trellis decoder a signal that consists of only the subtraction of two adjacent data symbols that come from the same trellis encoder, one transmitted before, and one after the segment sync. The interference introduced by the segment sync symbol is removed in this process, and the overall channel response seen by the trellis decoder is the single-delay partial response filter.

The complexity of the trellis decoder is dependent upon the number of states in the decoder trellis. Since the trellis decoder operates on an 8-state decoder trellis when the comb filter is active, this defines the amount of processing that is required of the trellis decoder. The decoder must perform an Add-Compare-Select (ACS) operation for each state of the decoder. This means that the decoder is performing 8 ACS operations per symbol time. When the comb filter is not activated, the decoder operates on a 4-state trellis. The decoder hardware can be constructed such that the same hardware that is decoding the 8-state comb filter trellis can also decode the 4-state trellis when the comb filter is disengaged. So there is no need for separate decoders for the two modes. The 8-state trellis decoder requires less than 5000 gates.

It should be noted that after the ATV transition period and NTSC is no longer being transmitted, the NTSC rejection filter and the 8-state trellis decoder can be eliminated from receivers.

It should be noted that after the ATV transition period and NTSC is no longer being transmitted, the NTSC rejection filter and the 8-state trellis decoder can be eliminated from receivers.

6.2.3.10 Data De-Interleaver (Modified 12-7-94)

The convolutional de-interleaver performs the exact inverse function of the transmitter convolutional interleaver. Its 1/6 data field depth, and intersegment "dispersion" properties allow noise bursts lasting about 193 SYMBOL 109 \f "Symbol" secs to be handled. Even strong NTSC cochannel signals passing through the NTSC rejection filter and creating short bursts due to NTSC vertical edges, are reliably handled due to the interleaving and R-S coding process. The de-interleaver use data field sync for synchronizing to the first data byte of the data field. The convolutional de-interleaver is shown in Figure 29.

6.2.3.11 Reed-Solomon Decoder

The trellis-decoded byte data is sent to the (207,187) t=10 R-S decoder, where it uses the 20 parity bytes to perform the byte-error correction on a segment-by-segment basis. Up to 10-byte errors/data segment are corrected by the R-S decoder. Any burst errors created by impulse noise, NTSC cochannel interference, or trellis-decoding errors, are greatly reduced by the combination of the interleaving and R-S error correction.

6.2.3.12 Data De-Randomizer

The de-randomizer accepts the error-corrected data bytes from the R-S decoder, and applies the same PRS randomizing code to the data. The PRS code is generated identically as in the transmitter, using the same PRS generator feedback and output taps. Since the PRS is locked to the reliably recovered Data Field Sync (and not some codeword embedded within the potentially noisy data), it is exactly synchronized with the data, and performs reliably.

6.2.3.13 Receiver Loop Acquisition Sequencing

The receiver incorporates a "universal reset" which initiates a number of "confidence counters" and "confidence flags" involved in the lock-up process. A universal reset occurs, for example, when tuning to another station or turning on the receiver.

The various loops within the VSB receiver acquire and lock-up sequentially, with "earlier" loops being independent from "later" loops. The order of loop acquisition is as follows:

* Tuner 1st LO synthesizer acquisition

* Non-coherent AGC reduces unlocked signal to within A/D range

* Carrier acquisition (FPLL)

* Data segment sync and clock acquisition

* Coherent AGC of signal (IF and RF gains properly set)

* Data field sync acquisition

* NTSC rejection filter insertion decision made

* Equalizer completes tap adjustment algorithm

* Trellis and R-S data decoding begin

Most of the loops mentioned above have confidence counters associated with them to insure proper operation. However, the build-up or let-down of confidence is not designed to be equal. The confidence counters buildup confidence quickly for quick acquisition times, but lose confidence slowly to maintain operation in noisy environments. The VSB receiver sync and clock recovery circuits will work in S/N conditions of 0 dB or less as well as in severe interference situations.

6.3. High Speed Cable Mode System Description (Modified 12-7-94)

The high data rate cable mode trades off transmission robustness (28.3 dB signal-to-noise threshold) for system data rate (43 Mbit/sec). Most parts of the cable mode VSB system are identical or similar to the terrestrial system. A pilot, data segment sync, and data field sync are all used to provide robust operation. The pilot in the cable mode also adds 0.3 dB to the data power. The symbol, segment, and field signals and rates are all the same, allowing either receiver to lock up on the other's transmitted signal. Also, the data frame definitions are identical. The primary difference is the number of transmitted levels (8 versus 16) and the use of trellis coding and NTSC interference rejection filtering in the terrestrial system.

The RF spectrum of the cable modem transmitter looks identical to the terrestrial system, as illustrated in Fig 30. Peak-to-average power ratio, shown in Fig 31, is very similar between the two VSB systems. The 16-VSB signal has slightly higher peak-to-average power ratio due to its 16- level random data. The error probability of 16-VSB, shown in Fig 32, is $3x10^{-6}$ BER at about 28.3 dB S/N with forward error correction provided by Reed-Solomon. Fig 33 illustrates a typical data segment, where the number of data levels is seen to be 16 due to the doubled data rate. Each portion of 828 data symbols represents 187 data bytes and 20 Reed-Solomon bytes followed by a second group of 187 data bytes and 20 Reed-Solomon bytes. This is the situation before convolutional interleaving. After interleaving, the data bytes and R-S bytes appear where the interleaver places them.

Fig 34 shows the block diagram of the transmitter. It is identical to the terrestrial VSB system except the trellis coding is replaced with a Mapper which converts data to multi-level symbols. The Mapper is shown in Figure 35. This conversion is performed by the trellis coding in the terrestrial VSB system. The receiver, shown in Fig 36, is identical to the VSB terrestrial receiver, except that the trellis decoder is replaced by a slicer, which translates the multi-level symbols into data. Instead of an 8-level slicer, a 16-level slicer is used. Also note that no NTSC interference rejection filter corresponding to the encoder trellis code concatenated with the filter.

The interleaver employed in the 16-VSB Cable mode is a 26 data segment intersegment convolutional byte interleaver. Interleaving is provided to a depth of about 1/12 of a data field (2ms deep). The system will tolerate a burst error of 96.6µsec.

6.4. Summary

The VSB transmission system provides high performance and low cost in both the terrestrial and cable modes. The terrestrial mode combines R-S and trellis coding with an NTSC rejection filter for maximum coverage area during and after the ATV transition period. Both the terrestrial and cable modes make use of a pilot, segment sync, and training signal for virtually no implementation loss. This means that theoretical performance is possible today without relying on future improvements. This performance is also achieved with minimal hardware complexity.

Factors contributing to low cost receivers are:

Simple Carrier Recovery (FPLL) using small pilot

Standard technology IF SAW filters

High immunity to phase noise (phase tracker) for inexpensive tuners

Symbol-spaced sampling for all data processing circuits

Single A/D converter (I channel only)

Simple Clock Recovery using data segment syncs

Real T-sampled Adaptive Equalizer operating at 10.76 MHz

6.5 Future Considerations

During the first round of ATV testing, the 4-VSB system demonstrated a bi-rate capability whereby some data was transmitted more robustly in exchange for a lower net data rate. As delivered for G-A testing, the VSB systems will be configured to provide all the data at equal robustness. Bi-rate data can be a future option if considered important by the Advisory Committee. Bi-rate transmission in the trellis-coded VSB transmission system will have the same data and robustness tradeoffs as in the 4-VSB system already evaluated by the advisory committee.

The cable mode has the capability of being flexible in that the data rate can be traded for a lower white noise threshold, all under headend control. A given 6 MHz RF channel on the cable can be set to have a lower data rate by sending fewer bits/symbol, thus gaining white noise performance. The receiver knows what multi-level data is being transmitted by reading a binary code (always transmitted 2-level during the data field sync). This provides a cable operator with system flexibility as channels are added to the cable system.

6.6 References

[1] Zenith [September 1991] Digital Spectrum Compatible HDTV: Technical Details. Monograph published by Zenith Electronics Corp & AT&T Bell Laboratories. FCC ACATS Document SS/WP1-0193.

[2] AT&T and Zenith Electronics Corporation, Oct. 26, 1992, DSC-HDTV System Improvements, submitted to Technical Sub-Group of the ATV Special Panel on Proposed ATV System Improvements.

[3] Certification Presentation, Systems Subcommittee, Working Party 1, Washington, D.C., November 6, 1991.

[4] VSB Transmission System Technical Details, Dec. 17, 1993, by Zenith Electronics Corp., submitted to Transmission Expert Group of the Technical Sub-Group of the ATV Special Panel.

TABLE 6.1 - VSB Parameters

Parameter	*Terrestrial Mode*	*High Data Rate Cable Mode*
Channel Bandwidth	6 MHz	6 MHz
Excess Bandwidth	11.5%	11.5%
Symbol Rate	10.76 MSPS	10.76 MSPS
Bits per Symbol	3	4
Trellis FEC	2/3 rate	None
Reed-Solomon FEC	T=10 (207,187)	T=10 (207,187)
Segment Length	832 Symbols	832 Symbols
Segment Sync	4 symbols per segment	4 symbols per segment
Frame Sync	1 per 313 segments	1 per 313 segments
Payload Data Rate*	19.39 Mb/s	38.78 Mb/s
NTSC Co-Channel Rejection	NTSC Rejection Filter in receiver	N/A
Pilot Power Contribution	0.3 dB	0.3 dB
C/N Threshold	14.9 dB	28.3 dB

*This is calculated as follows:

S_R = Symbol Rate - (4.5/286) 684 = 10.762237 --- M symbols/second

Segment Rate = S_R/832 = 12.93538 --- k segments/second

Payload Segment Rate = segment rate x 312/313 = 12.89405 --- k segments/second

Payload Bit Rate = Payload Segment Rate x 188 bytes/segment x 8 bits/byte = 19.39265845 --- Mb/s for terrestrial and 38.78531690 --- Mb/s for cable.

Chapter 7

GRAND ALLIANCE SYSTEM SUMMARY

7.1 Specification Tables

The following tables document features supported in the system description of the Grand Alliance HDTV specification. While the initial hardware prototype may not completely exercise all of the capabilities of the system standard, it is anticipated that future implementations will benefit from the flexibility provided by the standard.

SPECIFICATION TABLES

Video Parameter	**Format 1**	**Format 2**
Active Pixels	1280 (H) x 720 (V)	1920 (H) x 1080 (V)
Total Samples	1600 (H) x 787.5 (V)	2200 (H) x 1125 (V)
Frame Rate	60 Hz Progressive 30 Hz Progressive 20 Hz Progressive	60 Hz Interlaced / 30 Hz Progressive / 24 Hz Progressive
Chrominance Sampling	4:2:0	
Aspect Ratio	16:9	
Data Rate	Selected Fixed Rate, 10-45 Mbits / sec / or Variable	
Colorimetry	SMPTE 240 M	
Picture Coding Types	Intra Coded (I) / Predictive Coded (P) / Bidirectionally Predictive Coded (B)	
Video Refresh	I-Picture / Progressive	
Picture Structure	Frame / Field	
Coefficient Scan Pattern	Zig-Zag / Alternate Zig Zag	
DCT Modes	Frame	Frame / Field (60 Hz only)
Motion Compensation Modes	Frame	Frame / Field (60 Hz only) / Dual Prime (60 Hz only)

P-Frame Motion Vector Range	Horizontal: Unlimited by syntax Vertical: -128,+127.5
B-Frame Motion Vector Range (forward and backward)	Horizontal: Unlimited by syntax Vertical: -128,+127.5
Motion Vector Precision	1/2 Pixel
DC Coefficient Precision	8 / 9/ 10 bits
Rate Control	Modified TM5 with Forward Analyzer
Film Mode Processing	Automated 3:2 Pulldown Detection and Coding
Maximum VBV Buffer Size	8 Mbits
Intra / Inter Quantization	Downloadable Matrices (Scene Dependent)
VLC Coding	Separate Intra and Inter Runlength / Amplitude Codebooks
Error Concealment	Motion Compensated Frame Holding (Slice Level)

SPECIFICATION TABLES

Transmission Parameter	**Terrestrial Mode**	**High Data Rate Cable Mode**
Channel Bandwidth	6 MHz	6 MHz
Excess Bandwidth	11.5%	11.5%
Symbol Rate	10.76 MSPS	10.76 MSPS
Bits per Symbol	3	4
Trellis FEC	2/3 rate	None
Reed-Solomon FEC	T=10 (207, 187)	T=10 (207, 187)
Segment Length	832 Symbols	832 Symbols
Segment Sync	4 Symbols per segment	4 Symbols per segment
Frame Sync	1 per 313 segments	1 per 313 segments
Payload Data Rate	19.3 Mb/s	38.6 Mb/s
NTSC Co-Channel Rejection	NTSC Rejection Filter in receiver	N/A
Pilot Power Contribution	0.3 dB	0.3 dB
C/N Threshold	14.9 dB	28.3 dB

Audio Parameter	
Number of Channels	5.1
Audio Bandwidth	20-20KHz
Sampling Frequency	48 KHz
Dynamic Range	100 dB
Compressed Data Rate	384 KBits / sec.

Transport Parameter	

Multiplex Technique	MPEG-2 Systems Layer
Packet Size	188 Bytes
Packet Header	4 Bytes including sync
Number of services Conditional Access Error Handling Prioritization	 Payload scrambled on service basis 4-bit continuity counter 1 bit / packet
System Multiplex	Multiple program capability described in PSI stream

Chapter 8

PROTOTYPE HARDWARE IMPLEMENTATION

8.1 Overview

To demonstrate and verify the performance of the proposed standard, the member companies of the Grand Alliance, have combined their efforts to produce a hardware prototype for testing at the Advanced Television Test Center later this year. This prototype implementation will demonstrate excellent picture and sound quality, a flexible, packet-based transport, and transmission performance superior to previously tested systems. It is impractical, however, to incorporate all the flexibility and extensibility permitted by the standard into any one implementation. This chapter outlines the prototype design considerations and elements of the standard that will not be fully exercised in the initial implementation.

8.2 Video Compression Subsystem

The Video Compression subsystem implements a high level, main profile MPEG-2 compression algorithm. The MPEG-2 syntax offers an extremely powerful and flexible toolkit of compression techniques to cover a variety of multimedia communication capabilities. The prototype hardware will demonstrate excellent picture quality at the available bit rates, however it will not utilize all of the features available in the MPEG-2 syntax. Additionally, there are some features of the MPEG-2 algorithm that will be demonstrated with some restricted capabilities.

8.2.1 Picture Formats Supported

The prototype hardware will be capable of compressing pictures in two input formats: 1920 x 1080 interlaced pictures at 60 fields per second and 1280 x 720 progressive pictures at 60 frames per second. The sequences will be coded in the format in which they are received. The compression decoder will be capable of recognizing, decoding and displaying pictures coded in either the progressive or interlace input format. Additionally, the decoder will have the capability to convert interlaced coded formats to a progressive display format. A similar capability will demonstrate conversion from progressive to interlaced formats.

The prototype will employ lookahead circuitry to detect and identify sequences that were originally sourced at 24 or 30 Hz progressive formats, and will take advantage of this redundancy by only coding the necessary frames. the hardware is not capable of directly receiving pictures whose input format and timebase are 24 frames per second. Information will be passed in the coded bitstream to allow the decoder to reconstruct and display the sequences at the appropriate input format.

8.2.2 Motion Compensation

While the MPEG-2 syntax allows very extensive motion search range, the prototype will restrict its search range to +/- 128 pixels horizontally and +/- 32 pixels vertically for forward Predicted (P) frames, and +/- 64 pixels horizontally and +/- 32 pixels vertically in each temporal direction for Bi-directional predicted (B) frames. The motion vectors will be calculated and coded with 1/2 pixel accuracy. The prototype hardware will not implement concealment motion vectors.

The specification of the GA compression system includes the adaptive use of field, frame and dual prime prediction techniques for interlaced video. Since the dual prime technique is primarily useful for low-delay applications, it will not be included in the prototype hardware, in the interest of time.

8.2.3 Coding

The prototype implements high-level, main profile of the MPEG-2 video coding standard. While the standard allows for variable bit-rate applications, the prototype will employ a selectable constant bit rate algorithm. The rate will be set through comminication with a computer console, and will allow the ability to show different allocations for video and data services.

Each video frame is coded in one of three modes: Intra-coding (MPEG I-picures), Predictive coding (MPEG P-pictures) or bi-directional predictive coding (MPEG B-pictures). The period of I-pictures (Group-of-pictures size, N) and the distance in number of frames between two anchor (I or P) pictures (known as M), can be programmed.

In the above mentioned mode of operation, I-pictures provide refresh. The codec will allow an alternate way of refreshing, which is achieved through progressive refresh. In this mode, parts of P-pictures (I-slices) are refreshed progressively. I-frames will be sent periodically when using progressive refresh mode to facilitate editing.

The chroma samples will be subsampled by a factor of two horizontally and vertically (4:2:0), relative to the luma sampling grid. The prototype will only use the standard coefficient scan pattern, not the alternate zig-zag. The DC DCT coefficients may be represented with a precision of either 8, 9, or 10 bits in the GA system. However the prototype will only support the use of 8 or 9 bits.

The GA specification includes the option of using field-structure pictures when processing interlaced video, however this option will not be used in the prototype. The maximum number of coded bits per frame will be 8 Mbits.

In addition to the features listed above, there are a number of extensions to the syntax supported in the GA system that are not implemented in the prototype. Pan and scan and picture display extensions will not be implemented. Also absent from the prototype will be spatial and temporal scalability extensions, that are not part of the main profile.

8.3 Audio Compression Subsystem

The prototype implementation of the audio compression subsystem is expected to include all features in the system description. Further details of the Audio compression prototype will be forthcoming.

8.4 Transport Subsystem

The transport subsystem is responsible for multiplexing the variety of services into a single bit stream for transmission. This subsystem also has the responsibility for managing and delivering synchronization information between the encoders and decoders. The GA transport system provides an extrordinary degree of flexibility. The transport bit stream could describe a large number of programs, each potentially consisting of a large number of individual services. This flexibility is intended for a broad scope of services and to allow for future growth.

The prototype hardware will implement all the MPEG-2 syntax elements that would allow this high degree of flexibility, but obviously will be limited in its ability to simultaneously exercise this flexibility. Nominally, the prototype will carry a single program of HDTV services. The prototype will, however, be capable of delivering two independent programs simultaneously. Each program can be comprised of up to five independent services (typically, one video, two audio, and two ancillary data services). The ancillary data services can be either synchronous or asynchronous with the video and audio services. For an 18.8 Mbps transmitted data stream, the transport decoder will be able to recognize all five of these services, but will simultaneously decode only three services (one video, one audio, and one data).

The capability to assign a higher priority to certain transport packets is a capability of the system that will not be used in the prototype hardware. The splice countdown function will also be absent in this first hardware implementation, however the system will allow for the discontinuity flag to signal the decoder of switching that has occured in the compressed bit stream domain.

To demonstrate the ability of the transport layer to support conditional access, the prototype will include the capability to scramble the payload of individual packets on a "service by service" basis. Packets whose payloads include adaptation headers, will not be scrambled. As a demonstration, this scrambling circuitry is based on the DES electronic codebook encryption algorithm. Key management for the decryption of these services can also be demonstrated as an example implementation.

8.5 Transmission Subsystem

The prototype implementation of the transmission subsystem is expected to include all features in the system description. Further details of the Transmission prototype will be forthcoming.

Chapter 9

PROJECTED PROTOTYPE PERFORMANCE

9.1. Projected Prototype Performance (Modified 12-7-94)

The following tables detail the typical performance anticipated for the prototype implementation, when tested at the ATTC. The tables identify tests that have been adopted by SS/WP-2 for system testing and evaluation. For each of these tests, a target specification has been entered. These target specifications have been developed by the Grand Alliance Specialist Group in cooperation with the ACATS Expert Groups, and detail the projected performance of the Grand Alliance prototype system. Comments may be entered to explain differences between the target specification and the measured value.

Transmission Tests			
Test	Target Specification	Measured Value	Comments
CO-A/N	< 36.5 dB		
CO-N/A	< 3.5 dB		
CO-A/A	< 16.6 dB		
UP-A/N	< -12.5 dB		
UP-N/A	< -43 dB		
UP-A/A	< -37.5 dB		
LO-A/N	< -14.5 dB		
LO-N/A	< -41.5 dB		
LO-A/A	< -37.5 dB		
N-2 Taboo A/N	< -23.5 dB		
N+2 Taboo A/N	< -28.5 dB		
N+4 Taboo A/N	< -22.5 dB		
N+14 Taboo A/N	< -32.5 dB		
N+15 Taboo A/N	< -22.5 dB		
N-8 Taboo A/N	< -25.5 dB		
N-7 Taboo A/N	< -28.5 dB		
N+7 Taboo A/N	< -29.5 dB		
N+8 Taboo A/N	< -36.5 dB		
N-2 Taboo N/A	< -53 dB		
N-2 Taboo A/A	< -53 dB		
N+2 Taboo N/A	< -53 dB		
N+2 Taboo A/A	< -53 dB		
N-3 Taboo N/A	< -53 dB		
N-3 Taboo A/A	< -53 dB		
N+3 Taboo N/A	< -53 dB		
N+3 Taboo A/A	< -53 dB		
Random Noise in Presence of Multipath	< 3.5 dB		

Discrete Frequency Tests			
Test	Target Specification	Measured Value	Comments
Discrete Frequencies (25)	< -39.5 dB adj. ch. < 12.75 dB in band		

Power Measurement Tests			
Test	Target Specification	Measured Value	Comments
Peak/Average Power (99.9% probability)	< 6.95 dB		

Video — Objective Tests			
Test	Target Specification	Measured Value	Comments
Static Resolution, Luma, H/V/D, 1080x1920	860/700/1100 lph		
Static Resolution , Chroma, H/V/D, 1080x1920	430/350/550 lph		
Dynamic Resolution, Camera, Luma, H/V/D, 1080x1920	690/560/885 lph		
Dynamic Resolution, Camera, Chroma, H/V/D, 1080x1920	345/280/440 lph		
Static Resolution, Luma, H/V/D, 1080x1440	650/700/955 lph		
Static Resolution, Chroma, H/V/D, 1080x1440	325/350/475 lph		
Dynamic Resolution, Camera, Luma, H/V/D, 1080x1440	520/560/765 lph		
Dynamic Resolution, Camera, Chroma, H/V/D, 1080x1440	260/280/380 lph		
Video-Audio Latency (1080-lines)	< 15 msec		
Video-Captioning Latency (1080-lines)	< 100 msec		
Static Resolution, Luma, H/V/D, 720x1280	580/650/870 lph		
Static Resolution , Chroma, H/V/D, 720x1280	290/325/435 lph		
Dynamic Resolution, Camera, Luma, H/V/D, 720x1280	460/520/695 lph		
Dynamic Resolution, Camera, Chroma, H/V/D, 720x1280	230/260/345 lph		
Video-Audio Latency (720-lines)	< 15 msec		
Video-Captioning Latency (720-lines)	< 100 msec		

Video Quality Tests — Non-Expert Observers			
Test	Target Specification[10]	Measured Value	Comments
Quality, Basic Material (1080-lines)	0.3 Grade below reference[11]		
Quality, Noise & Cuts (1080-lines)	1.0 Grade below reference[12]		
Quality, Graphics & NII (1080-lines)	1.0 Grade below reference[13]		
Quality, 24 fps Film (1080-lines)	0.25 Grade below reference[14]		
Quality, Video/Auxiliary Data Tradeoff (1080-lines)	Video 1.0 Grade/Mb below reference[3] Film 0.5 Grade/Mb below reference[3]		
Quality, Receiver Conversion, 720-lines transmission, 1080-lines display	1.0 Grade below reference		
Quality, Basic Material (720-lines)	0.3 Grade below reference[2]		
Quality, Noise & Cuts (720-lines)	1.0 Grade below reference[3]		
Quality, Graphics & NII (720-lines)	1.0 Grade below reference[4]		
Quality, 24 fps Film (720-lines)	0.25 Grade below reference[5]		
Quality, Video/Auxiliary Data Tradeoff (720-lines)	Video 1.0 Grade/Mb below reference[3] Film 0.5 Grade/Mb below reference[3]		
Quality, Receiver Conversion, 1080-lines transmission, 720-lines display	1.0 Grade below reference		

[10] Grade is the average over all sequences tested, not the maximum.

[11] This specification is based on average performance of the best performing digital system during round-one testing. Round-two testing will use the more difficult test material from round-one testing and newer, potentially more difficult material. Accordingly, comparison of the measured value with the target specification should be tempered by consideration of the difficulty of the test material used.

[12] Quantitative data do not exist for the previous systems. This is a suggested target specification.

[13] Valid data do not exist from tests of the previous systems. This is a suggested target specification.

[14]Comparision of the measured value with the target specification should be tempered by consideration of the selected test material.

Digital Specific Tests — Expert Observers			
Test	Target Specification	Measured Value	Comments
Threshold Characteristics for Random Noise - Video	< 15.6 dB		
Threshold Characteristics for Random Noise - Audio	< 15.6 dB		
Threshold Characteristics for Random Noise (Audio + Video)	Audio usable at or beyond video POU		
Free Form Viewing (1080-lines)	As good as or better than the best previous digital system[5]		
Quality, Scene Cuts (1080-lines)	As good as or better than the best previous digital system[15]		
Noise in Video Source (1080-lines)	As good as or better than the best previous digital system[6]		
Video Coder Overload (1080-lines)	As good as or better than the best previous digital system[6]		
Motion Compensation Overload (1080-lines)	As good as or better than the best previous digital system[6]		
Quality, Video/Auxiliary Data Tradeoff (1080-lines)	See note[16]		
Concatenation Quality (1080-lines)	As good as or better than the best previous digital system[6]		
Free Form Viewing (720-lines)	As good as or better than the best previous digital system[5]		
Quality, Scene Cuts (720-lines)	As good as or better than the best previous digital system[6]		
Noise in Video Source (720-lines)	As good as or better than the best previous digital system[6]		
Motion Compensation Overload (720-lines)	As good as or better than the best previous digital system[6]		
Quality, Video/Auxiliary Data Tradeoff (720-lines)	See note[7]		

[15] In testing the previous systems, qualitative observations were made by expert observers. No quantitative data were taken. The System Specific / Digital Specific Task Force should record quantitative comparisons with the reference and with the previous digital systems.

[16] Quantitative data do not exist for the previous systems. The suggested target specification, using the 10-grade scale employed by the expert observers, is 2 (of 10) grades per Mb below reference for film and 4 (of 10) grades per Mb below reference for video.

Concatenation Quality (720-lines)	As good as or better than the best previous digital system[6]		

ATV Subjective Audio & Long Form Entertainment Tests			
Test	Target Specification	Measured Value	Comments
ATV Multichannel Audio	Subjectively as good as or better than the Grand Alliance / Audio Experts Group tests		
Long Form Entertainment Program	EO&C, no noticeable impairments		

Audio — Objective Tests			
Test	Target Specification	Measured Value	Comments
Frequency Response, Main Channels (10 Hz - 20 kHz)	± 0.25 dB		
Frequency Response, Subwoofer Channel (10 Hz - 120 Hz)	± 0.50 dB		
Dynamic Range	> 90 dB		
THD (at nominal test level)	< 0.1 %		
THD + N (at nominal test level)	< 0.1 %		
IM Distortion (at nominal test level)	< 0.1 %		

Interoperability & Packetization Tests			
Test	Target Specification	Measured Value	Comments
Header/Descriptor Robustness	Demonstration only		
Switching between Compressed Data Streams	Demonstration only		
Simulation of ATM Network Transmission	Demonstration only		
Transport Interoperability with Computer Networks	Demonstration only		

Cable Television Tests			
Test	Target Specification	Measured Value	Comments
Composite Second Order Distortion	< 25 dB		
Composite Triple Beat Distortion	< 37 dB		
Phase Noise	< 81 dB		
Residual FM	> 6.5 kHz		
Fiber Optic Tests	> 4.5 %		
Channel Change / Channel Acquisition	< 0.7 sec		
Threshold Characteristics for Random Noise - Data	< 15.6 dB		
Local Oscillator Instability	± 89 kHz		
Dynamic Multipath - Acquisition Time in the Presence of Multipath and Noise	< 0.75 sec		
Dynamic Multipath - Simulate Tower Sway	< 9.5 dB		
Burst Error Correction	> 169 µsec @ 10 Hz > 1.05 kHz @ 20 µsec		

Cable Television Tests with High Data Rate Transmission			
Test	Target Specification	Measured Value	Comments
Composite Second Order Distortion	< 38 dB		
Composite Triple Beat Distortion	< 49 dB		
Phase Noise	< 87 dB		
Residual FM	> 4.0 kHz		
Fiber Optic Tests	> 4.0 %		
Channel Change / Channel Acquisition	< 0.7 sec		
Threshold Characteristics for Random Noise - Data	< 28.85 dB		
Local Oscillator Instability	± 89 kHz		
Dynamic Multipath - Acquisition Time in the Presence of Multipath and Noise	< 0.75 sec		
Burst Error Correction	> 129 µsec @ 10 Hz > 1.45 kHz @ 20 µsec		