

## **2. Syntax Hierarchies and Encapsulation**

### **2.1 VC-1 Syntax Hierarchy in Bitstreams**

#### **WMV-9 and VC-1 Standards**

WMV-9 is a video codec developed by Microsoft. It is widely used for streaming media over the Internet due to the popularity of MS Windows Operating Systems. Since WMV-9 is a generic coder, many of its algorithms/tools can be used for a variety of applications under different operating conditions. Originally, three profiles were defined – Simple Profile, Main Profile and Complex Profile. However, Complex Profile was unofficially dropped. Consequently, WMV-9 focuses more on compression technology for progressive video up to Main Profile, while VC-1 has been developed for broadcast interlaced video as well as progressive video [srinivasan:WMV9, SMPTE:VC1].

VC-1 has three profiles – Simple (SP), Main (MP) and Advanced Profiles (AP). Simple and Main Profiles of VC-1 correspond to Simple and Main Profiles of WMV-9, respectively. Advanced Profile is mainly targeted for broadcast applications. Those two technologies are almost identical in philosophy and the tools they provide except where interlaced video is concerned. VC-1 is a pure video compression technology derived from Microsoft's proprietary WMV-9, and is expected to be deployed as a key engine in satellite TV, IP set-tops and HD-DVD/ Bluray-DVD recorders. HD-DVD and Bluray-DVD adopt only the Advanced Profile of VC-1.

#### **Key Compression Tools for WMV-9 Video**

Like all other MPEG standards, WMV-9 is based on motion compensated transform coding. Originally YUV4:2:0 and YUV4:1:1 were defined as input formats for progressive and interlaced video, respectively. Since interlaced video is no longer considered with WMV-9, 8-bit YUV4:2:0 is the only input format.

There is no fixed GOP structure in WMV-9. I, P, B, BI and Skipped P are defined as picture/frame types. I (Intra) frames do not have to be periodic. Rather, any reference can be either an I or P (Predicted) frame. Therefore, if there is no big scene change for a lengthy period of time, there could be only P frames as references.

The number of B frames (Bi-directionally predicted frames) between two reference frames can vary. Maximally, there could be seven B frames. BI frames are almost identical to I frames. If there are major continuous scene changes, some B frames may not capture similarities from two reference frames. In such a case, intra mode performance might be better than prediction mode performance. BI frame compression is a good choice for this case. Since BI frames are not used as reference, dropping those frames is possible under certain conditions such as lack of computation or bandwidth.

The last frame type is the Skipped P frame. If the total length of the data comprising a compressed frame is 8 bits, then this signals that the frame was coded as a non-coded P frame in an encoder.

A key compression tool in WMV-9 is the adaptive block size transform. Transform block size can change adaptively, while the block size for motion compensation is either 16x16 or 8x8. Note that this is quite the opposite to that of H.264. H.264 normally uses fixed size 4x4 (or 8x8 in High Profile) transform with various block sizes for motion compensation. There are four transform sizes – 8x8, 4x8, 8x4 and 4x4. The transforms are 16-bit transforms where both sums and products of all 16-bit values produce results with 16 bits – the inverse transform can be implemented in 16-bit fixed point arithmetic. Note that the transform approximates a DCT, and the norms of the basis function between transforms are identical to enable the same quantization scheme through various transform types.

There are three main options for Motion Compensation (MC): 1. Either half-pel or quarter-pel resolution MC can be used. 2. Either bi-cubic or bi-linear filter can be used for the interpolation. 3. Either 16x16 or 8x8 block size can be used. These are all combined into a single MC mode with MVMODE and MVMODE2 syntax elements to be represented at the Frame level. Note that combinations are allowed in a specific way that clearly prioritizes the performance of MC. There is a tool mode in Sequence layer called FASTUVMC for motion vector computation in

Chroma components. If this is on, computed Chroma MVs are all rounded to the half-pel domain. Thus, interpolation for quarter points is not necessary for Chroma data at decoders – this saves a lot of computation in software-based decoder implementations.

Quantization is generally defined with two parameters in video standards – Qp and Dead-zone. There are two choices for Dead-zone in WMV-9 – 3Qp and 5Qp. There are two levels where this can be described: 1. Sequence header has QUANT field for this description – 3Qp or 5Qp for entire sequence. 2. Explicit option is written in each Picture header, or Implicit option is to describe it through PQindex. In I frames, PQAUNT is applied to entire MBs. However, DQUANT is used to adaptively describe Qp in each MB in P/B frames. In addition, there are other options to use only two Qps for an entire frame depending on either boundary MB or non-boundary MB.

There are two techniques used in WMV-9 to reduce blocky effects around a transform boundary – Overlapped Transform (OLT) smoothing and In Loop deblocking Filtering (ILF). OLT is a unique and interesting technique to reduce blocky effect based on an accurately defined pre-/post-processing pair. The idea is that forward and inverse operations are defined in such a way that original data is recovered perfectly when operations are serially applied (forward and inverse). The forward transform exchanges information across boundary edges in adjacent blocks. The forward operation is performed before the main coding stage. Consider an example where one block preserves relatively good edge data, while the other block loses details of edge data. In this case, the blocky effect is very visible. At the decoder side, the inverse operation is required to exchange the edge data back again to obtain original data. By doing so, good quality and bad quality edges diffuse each other. Therefore, the blocky effect is significantly reduced.

On the other hand, ILF is a more or less heuristic way to reduce blocky effects. Blocky pattern is considered to be high frequency since abrupt value changes are happening around block edges. Considering that original data quality might also contain high frequency, relatively simple non-linear low pass filtering is applied about block edges in ILF. ILF is performed on I and P reference frames. Thus, the result of filtering affects the quality of pictures that use ILFed frames as references.

Entropy coding used in WMV-9 is a kind of Context-Adaptive VLC. Based on Qp, from which the coded quality can be guessed, a new set of VLC tables is introduced. Such examples include mid-rate VLC tables and high-rate VLC tables. In addition, based on MVs, another set of VLC tables is introduced. Such examples include low-motion DC differential tables and high-motion DC differential tables.

### **WMV-9 Video Specific Semantics and Syntax**

There are five levels of headers in WMV-9 video bitstream syntax – Sequence, Picture, Slice (not clearly defined in WMV-9), MB, and Block. Sequence header contains basic parameters such as profile, interlace, frame rate, bit rate, loop filter, overlap filter and some other global parameters. Picture header contains information about type of picture/ BFACTION/ PQindex/ MVMODE/ MVMODE2/ LumScale/ LumShift/ DQUANT related/ TTMBF/ TTFRM/ DCTACMBF/ DCTACFRM, etc. BFACTION data is relative temporal position of B that is factored into the computation of direct mode vectors. Note that the number of B frames inserted with geometrical position of B can be determined from this value. PQindex is interpreted for quantizer scale (QS) and quantizer types (3QP/5QP) in Implicit case, while quantizer types are explicitly defined in Sequence or Picture header in other cases. A combination of MVMODE and MVMODE2 represents a prioritized MC mode adopted for the current frame. LumScale/ LumShift are Intensity Compensation parameters. TTMBF is the flag that tells whether the additional field for Transform Type is in MB level or in Frame level. DCTACMBF is the flag that tells whether DCT AC Huffman table is defined in MB level or Frame level. TTFRM may be used to force a certain transform type in the frame level for P or B frames. DCTACFRM may be used to force a certain DCT AC Huffman table in the frame level. Slices are not clearly defined in WMV-9. When STARTCODE is set in the Sequence header, MB header contains SKIPMBBIT/ MVMODEBIT/ MVDATA/ TTMB, etc. SKIPMBBIT indicates whether the MB is “Skipped”; for MBs in P or B frames (i.e., P-MBs or B-MBs). This representation is extended to take Hybrid mode in WMV-9. MVMODEBIT is present in P-MBs in P frames if the picture is coded in “Mixed-MV” mode. If MVMODEBIT=0, the MB shall be coded in 1MV mode. If MVMODEBIT=1, the MB shall be coded in 4MV mode.

MVDATA tells whether the blocks are coded as Intra or Inter type. If they are coded as Inter 1MV, then MVDATA indicates MV differentials. If they are coded as Inter 4MV, then BLKMVDATA in each Block header indicates MV differentials. Block layer contains all transform coefficients. Sub-block pattern data is included in sub-blocks.

### **Simple and Main Profiles for VC-1 Video**

The Simple Profile (SP) and Main Profile (MP) of VC-1 are the same as those of WMV-9 in progressive compression. However, there is no Advanced Profile (AP) in WMV-9 to compare with the AP in VC-1. VC-1 AP mainly focuses on interlaced video compression technology. Legacy Complex Profile (CP) in WMV-9 can handle the interlaced video. However, the interlace tools cannot be compared directly in both standards. Even the input interlaced video formats are different -- the input format is YUV4:1:1 for WMV-9, but is YUV4:2:0 for VC-1. Note that only Field-based prediction and Frame-based prediction are selectively applied in each MB in the legacy CP of WMV-9. This tool is applied when the INTERLACE flag is turned on in the Sequence header in the legacy CP. The CP of WMV-9 was not considered for VC-1 interlaced video when the SP and the MP of WMV-9 were adopted in the VC-1 without change.

### **Advanced Profile for VC-1 Video**

Advanced Profile adds interlace tools to the Main Profile. The number of B frames is not fixed between two references in VC-1. New distance value comes into the Picture Structure at Entry Point Layer with REFDIST\_FLAG for interlaced video. REFDIST data indicates the number of pictures between the current picture and the reference one. Progressive-Picture/ Frame-Picture/ Field-Picture can be mixed in VC-1 AP. It is the encoder's job to construct each picture with a Frame Coding Mode (FCM).

The maximum number of references is two for a P Field-Picture. The references are specified in the Picture layer. If both references are used, the selection of reference information is described in MB level and Block level. The number of references is always four for a B Field-Picture — no

Picture layer selection is needed. So, the selection of reference is always in MB level and Block level. Note that one of the reference fields for a bottom field in a B frame is its top field itself.

A P Field-Picture has two MC modes (1MV with 16x16, 4MV with 8x8), while a B Field-Picture has three MC modes (1MV with 16x16 in forward or backward modes, 2MV with 16x16 in interpolative or direct modes, 4MV with 8x8 only in forward or backward modes). A P Frame-Picture has 4 MC modes (1MV Frame-based prediction with 16x16, 4MV Frame-based prediction with 8x8, 2MV Field-based prediction with 16x8 (each field), 4MV Field-based prediction with 8x8 (each field 16x8 divided to left/right 8x8)). A B Frame-Picture has four MC modes (1MV Frame-based prediction with 16x16 in forward or backward modes, 2MV Frame-based prediction with 16x16 in interpolative or direct modes, 2MV Field-based prediction with 16x8 (each field) in forward or backward modes, 4MV Field-based prediction with 16x8 (each field) in interpolative or direct modes). Once residual data is obtained after motion-estimation in encoders, a transform is applied on it.

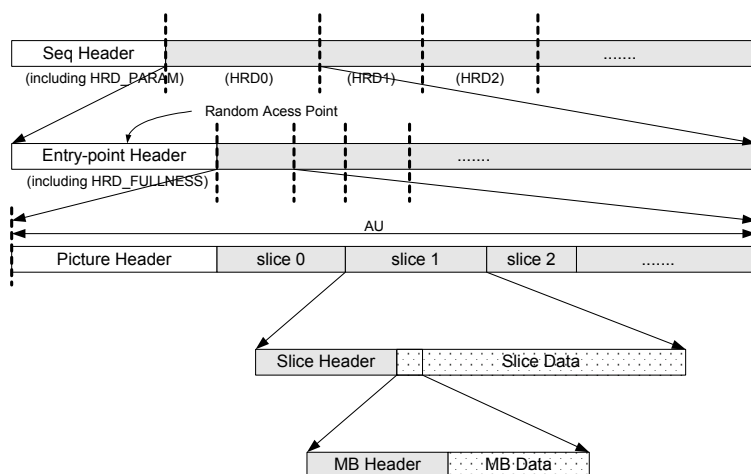
In Intra MBs or Intra Blocks, a transform is applied on original data. There are two transforms – Frame-transform and Field-transform. Frame-transform applies the transform on Frame-Picture data without any reordering, while Field-transform applies the transform on Frame-Picture data with sorted top/ bottom field data. Note that this option is only available in Frame-Pictures. Encoders decide which transform mode is applied in each MB. In the case of Intra MBs, the mode determined is written in FIELDTX. In Inter MBs, however, the mode is written in MBMODE.

Transform block size can change adaptively, while the size of motion compensation is one of 16x16/ 16x8/ 8x8 in VC-1 interlace video. There are four transform sizes as are in WMV-9 – 8x8, 4x8, 8x4 and 4x4.

The same two techniques are used in VC-1 to reduce blocky effects around transform boundaries – OLT smoothing and ILF. One important difference in the OLT technique between WMV-9 and VC-1 is to have the control even on MB level in I frame with CONDOVER and OVERFLAGS. The 128 level-shift is done on all the Intra MBs and Intra Blocks in VC-1, while the level-shift is performed only on Intra MBs and Intra Blocks that undergo OLT in WMV-9. In interlaced video, the OLT smoothing is applied only for vertical direction in Frame-Pictures, while it

is performed for both horizontal and vertical directions in Field-Pictures. Note that horizontal edge filtering might require top and bottom fields together as inputs in Frame-Pictures – this would make potential output filtered data blurry. That is why only vertical direction edges are OLT-filtered for Frame-Pictures in the VC-1 standard.

On the other hand, ILF filters both horizontal and vertical directions in Field-Pictures of interlaced video. In Frame-Pictures of interlaced video, however, horizontal and vertical ILFs are performed differently. ILF in vertical edges is the same as that of Field-Pictures, while ILF in horizontal edges is performed based on Field-based ILF filtering. In other words, only the same polarity data are considered in ILF filtering.



**Figure 2-1 Syntax Hierarchy for VC-1**

## VC-1 Video Specific Semantics and the Syntax

There are six levels of headers in VC-1 video bitstream syntax – Sequence, Entry Point, Picture, Slice, MB and Block. AP has explicit Sequence header, but SP/ MP don't have any Sequence header or Entry Point header in VC-1. The data necessary in the Sequence header should be provided by an external means. Sequence header contains basic parameters such as profile/ level, interlace, loop filter, max\_coded\_width,

max\_coded\_height and some other global parameters. This includes display related metadata and HRD parameters.

The Entry Point header is the random access point, and is present only in AP. It is used to signal control parameter changes in the decoding. Examples include broken\_link, closed\_entry, reldist\_flag, loopfilter, overlap, coded\_width, coded\_height and other global parameters until the next Entry Point header. It also contains HRD fullness information.

A key syntax layer is the Picture and it is the access unit (AU) in VC-1. A Picture is composed of many slices as shown in Figure 2-1. A Slice is composed of many MB rows/ MBs as explained in Chapter 1. An MB is composed of six 8x8 blocks as a coding unit.

The Picture header contains information about FCM/ TFF/ RFF/ RNDCTRL/ PQindex/ LumScale1~2 /LumShift1~2/ CONDOVER/ BFRACTION/ MVTAB/ CBPTAB/ MVTYPEMB [bitplane]/ FIELDTX [bitplane]/ ACPRED [bitplane]/ OVERFLAGS [bitplane]/ SKIPMB [bitplane]/ DIRECTMB [bitplane]/ FORWARDMB [bitplane]/ TRANSACFRM/ TRANSACFRM2/ TRANSDCTAB/ MVMODE/ MVMODE2/ 4MVSWITCH/ MBMODETAB/ IMVTAB/ ICBPTAB/ 2MVBPTAB/ 4MVBPTAB, etc. FCM is present only if INTERACE has the value 1, and it indicates whether the frame is coded as progressive/ Field-Picture/ Field-Frame. TFF and RFF are present as Top Field First and Repeat First Field flags respectively if PULLDOWN and INTERLACE are set to 1. RNDCTRL indicates the type of rounding used for the current frame. In P Field-Pictures, two intensity compensation parameters (LumScale and LumShift) are needed for top field and bottom field, respectively. CONDOVER is present only in I pictures and only when OVERLAP is on and PQUANT is less than or equal to 8. For SP and MP, PQUANT is the only condition for the operation of the OLT. Note that the OLT is performed only when PQUANT is larger and equal to 9. For AP, OLT can be even performed when PQUANT is less than or equal to 8 with introduction of CONDOVER syntax element. The syntax elements tagged with “[bitplane]” mean that there are corresponding syntax elements in MB header and the syntax elements in the Picture header are used instead with Raw mode turned off. For example, DIRECTMB syntax is just bitplane coding flags corresponding to DIRECTBBIT in each MB header. Bitplane coding is explained in Chapter 9. TRANSACFRM and TRANSACFRM2 are used to define AC Huffman coding tables selected for CbCr and Y, respectively.

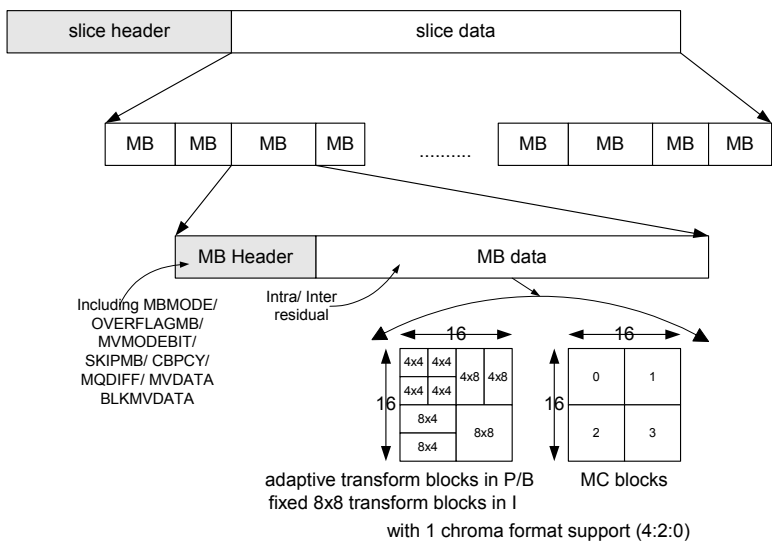


TRANSDCTAB is used to indicate whether the low motion table or the high motion table shall be used for DC value decoding. 4MVSWITCH shall be present in Interlace Frame P. If 4MVSWITCH=0, the MBs in the picture shall have only one or two MVs depending on the MB has been frame-coded or field-coded. If 4MVSWITCH=1, there shall be either one, two or four MVs per MB – this syntax shall not be present in Interlace Frame B. IMVTAB and ICBPTAB are syntax elements to select tables for Intensity Compensation. MBMODETAB is used to select Huffman tables for decoding MBMODE syntax element. The 2MVBPTAB and 4MVBPTAB syntax elements signal which one of four tables are used to decode 2MVBP and 4MVBP syntax elements, respectively.

The Slice header provides information about SLICE\_ADDR/ PIC\_HEADER\_FLAG. Slice Address is from 1 to 511, where the row address of the first MB in the slice is binary encoded. The picture header information is repeated in the slice header if the PIC\_HEADER\_FLAG is set to 1.

The MB header has TTMB/ OVERFLAGMB/ MVMODEBIT/ SKIPMBBIT/ FIELDTX/ CBPCY/ ACPRED/ MQDIFF/ ABSMQ/ MVDATA/ BLKMVDATA/ HYBRIDPRED/ MBVTYPE/ BMV1/ BMV2/ BMVTYPE/ DIRECTBBIT/ MBMODE/ 2MVBP/ 4MVBP/ MVSW, etc. MBMODE indicates whether Intra, Inter-1MV, Inter-4MV, CBP and MVDATA are present. OVERFLAGMB is present when CONDOVER has the binary value 11. OVERFLAGMB indicates whether to perform OLT within the block and neighboring blocks. MBVTYPE is BMV1 and BMV2 are used for the 1<sup>st</sup> MVDATA and the 2<sup>nd</sup> MVDATA in B-MBs, respectively. The decoding procedure for BMV1 and BMV2 shall be identical to the procedure for MVDATA. BMVTYPE indicates whether the MB uses forward, backward or interpolative prediction modes for B-MBs. Note that direct prediction mode is separately described in DIRECTBBIT syntax element for B-MBs. MBMODE is defined for Interlace Field P/B and Interlace Frame P/B MBs to indicates coding modes including FIELDTX and CBPCY. 2MVBP is defined for Interlace Frame P/B MBs to indicates which of two luma blocks contain non-zero MVD – this syntax element shall be present if the MBMODE syntax element indicates that the MB has two field MVs. 4MVBP is defined for Interlace Field P/B and Interlace Frame P/B MBs to indicates which of four luma blocks contain non-zero MVD – this

syntax element shall be present if the MBMODE syntax element indicates that the MB has two field MVs under the interpolative mode. MVSW shall be present in B-MBs if the MB is in field mode and BMVTYPE is forward or backward prediction mode. If MVSW=1, it shall indicate that the MV type and prediction type changes from forward to backward (or backward to forward) in going from the top to bottom field. If MVSW=0, the prediction type shall not change in going from the top to the bottom field. Other data can similarly be interpreted as those in WMV-9.



**Figure 2-2 Slice Syntax Hierarchy for VC-1**

There are many coding options for each MB as shown in Figure 2-2. There are four transforms shown to apply in each 8x8 block, where the same transform is specified to apply. For example, an 8x8 block cannot be partitioned into one 8x4 block and two 4x4 blocks. The choice among transform sizes is basically performed by the encoder to take advantage of statistics of input signals. Typically a long size transform is effective when the input signal has high correlation among localized spatial data, while a short size transform is effective when the input signal has relatively low

correlation. There are two MC modes in terms of size and the choice among MC sizes is made by the encoder to take advantage of statistics of input signals. Typically a large area is effective when a rigid object is moving (uniform motion in the area), while a small area is effective when tearing object (no object) and/or overlapped objects are moving (random motion in the area).

### **VC-1 Profiles/ Tools**

The Simple Profile targets low-rate internet streaming and low-complexity applications such as mobile communications, or play back of media in personal digital assistants. There are two levels in the profile – Low and Medium levels.

The Main Profile targets high-rate internet applications such as streaming, movie delivery via IP, or TV/ VOD over IP. This profile has three levels – Low, Medium and High.

The Advanced Profile targets broadcast applications, such as digital TV, HD-DVD for PC play back, or HDTV. It is the only profile that supports interlaced content. In addition, this profile contains the required syntax elements to transmit video bitstreams with encapsulations of generic systems, such as MPEG-2 Transport or Program Streams. This profile contains five levels – L0, L1, L2, L3 and L4.

Each class of bitstreams contains tool sets defined in Table 2-1 and parameters set defined in Table 2-2. Table 2-1 indicates the constraints on the algorithms or compression features for each of the profiles. Note that dynamic resolution change refers to scaling the coded picture size by a factor of two via the RESPIC syntax element in the MP.

Note that range adjustment refers to range reduction by a factor of two via the RANGEREDFRM syntax element in the MP, and range mapping by arbitrary factors via RANGE\_MAPY and RANGE\_MAPUV syntax elements in the AP.

**Table 2-1 VC-1 Profiles and Tools**

Tool Options	Simple Profile	Main Profile	Advanced Profile
Baseline Intra Frame Compression	x	x	x
Variable-sized Transform	x	x	x
16-bit Transform	x	x	x
Overlapped Transform	x	x	x
8x8 and 16x16 Motion Modes	x	x	x
Quarter-pixel Motion Compensation Y	x	x	x
Quarter-pixel Motion Compensation U, V		x	x
Start Codes		x	x
Extended Motion Vectors		x	x
Loop Filter		x	x
Dynamic Resolution Change		x	x
Adaptive MB Quantization		x	x
Bidirectional (B) Frames		x	x
Intensity Compensation		x	x
Range Adjustment		x	x
Interlace: Field/ Frame Coding Modes			x
Self Descriptive Fields/ Flags			x
GOP Layer/ Entry Points			x
Display Metadata (Pan/ Scan, Colorimetry, Aspect Ratio, Pulldown, Top Field First, Repeat First Field, etc.)			x

**Table 2-2 VC-1 Levels and Limitations**

Profile @Level	MB/s	MB/f	Examples	B	I	Rmax	Bmax	MV [H]x[V]
SP@LL	1,485	99	QCIF(176x144) @15fps			96	22	[-64,63 ¾] x [-32,31 ¾]
SP@ML	5,940	396	CIF(352x288) @15fps, 240x176 @30fps			384	77	[-64,63 ¾] x [-32,31 ¾]
MP@LL	7,200	396	QVGA(320x240) @24fps, CIF(352x288) @15fps	x		2,000	306	[-128,127 ¾] x [-64,63 ¾]
MP@ML	40,500	1,620	480p(720x480) @30fps, 576p(720x576) @25fps	x		10,000	611	[-512,511 ¾] x [-128,127 ¾]
MP@HL	245,760	8,192	1080p(1920x1080) @25/30fps	x		20,000	2,442	[-1024,1023 ¾] x [-256,255 ¾]
AP@L0	11,880	396	CIF(352x288) @25/30fps, SIF(352x240) @30fps	x		2,000	250	[-128,127 ¾] x [-64,63 ¾]
AP@L1	48,600	1,620	480i-SD, 704x480 @30fps, 576i-SD, 720x576 @25fps	x	x	10,000	1,250	[-512,511 ¾] x [-128,127 ¾]
AP@L2	110,400	3,680	480p, 704x480 @60fps, 720p, 1280x720 @25/30fps	x	x	20,000	2,500	[-512,511 ¾] x [-128,127 ¾]
AP@L3	245,760	8,192	1080i/p, 1920x1080 @25/30fps, 720p, 1280x720 @50/60fps, 2048x1024 @30fps	x	x	45,000	5,500	[-1024,1023 ¾] x [-256,255 ¾]
AP@L4	491,520	16,384	1080p, 1920x1080	x	x	135,000	16,500	[-1024,1023 ¾]

			@50/60fps, 2048x1536 @24fps, 2048x2048 @30fps					x [-256,255 3/4]
--	--	--	---	--	--	--	--	------------------

There are several levels for each of the profiles. Each level limits the video resolution, frame rate, HRD bit rate, HRD buffer requirements, and the motion vector range. These limitations are defined in Table 2-2. The column marked “B” denotes B frames and loop filter support, and “I” denotes interlace support. For Interlace, picture rate is described in “frames” per second. “Fields” per second is twice that value.

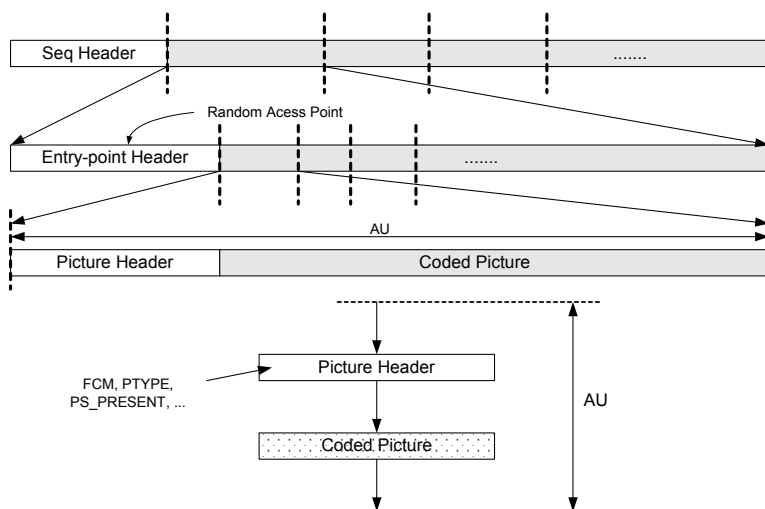
2.2 VC-1 Encapsulation in MPEG-2 Systems

Entry Point and Access Unit in VC-1

Entry Point headers are used in VC-1 Advanced Profile (AP) to support random access. An Access Unit (AU) contains all the data of a picture and padding bits that follow up to the next AU start code. Figure 2-3 shows the structure of the AUs in VC-1 bit stream. Coded picture data represents a video frame, regardless of whether the frame has been encoded as a progressive frame, interlace frame or interlace field mode. AUs start on a byte boundary and begin with either a Sequence start code, an Entry point start code or a Frame start code.

If the frame is not preceded by a Sequence start code, Sequence header or Entry Point header, the AU shall begin with a Frame start code. Otherwise, the AU shall start with the first byte of the first of these structures (excluding any stuffing bytes) before the Frame start code. An AU shall also include any user data start code and user data bytes at the sequence, entry point, frame or field level.

In the case of SP/ MP, an AU shall include all the coded data for a video frame, including the VC-1\_SPMP\_PESpacket\_PayloadFormatHeader( ) bytes that precedes it as well as any flushing bits present to ensure byte alignment that follows it up to, but not including, the start code of the next AU. The start of the next AU shall be the first byte in the next VC-1\_SPMP\_PESpacket\_PayloadFormatHeader( ) which is either a Sequence start code or a Frame start code.



**Figure 2-3 AU for VC-1**

## Encapsulation of VC-1 in PES

The carriage of VC-1 in MPEG-2 Systems is defined in SMPTE RP227 specification. VC-1 streams can be carried in either an MPEG-2 Transport Stream or an MPEG-2 Program stream [SMPTE:VC1systems].

MPEG-2 Systems require PES encapsulation of video data in order to generate PES as described in Chapter 1. The PES structures for MPEG-2 video and VC-1 video are largely similar. The VC-1 specific encapsulation requirements for PES are discussed in the section.

The PES of VC-1 is exactly the same as that of MPEG-2 video with the exception of VC-1 SP/ MP PES packet payload format header as shown in Figure 2-4. PES\_packet\_length field indicates VC-1 video. A value of “0” means that the semantics shall be extended to VC-1 video ESs. The extended fields contain all necessary parameters set at Sequence level for SP/ MP.

The stream\_id for VC-1 ESs shall be set to 0xfd to indicate the use of an extension mechanism mentioned in Chapter 1.

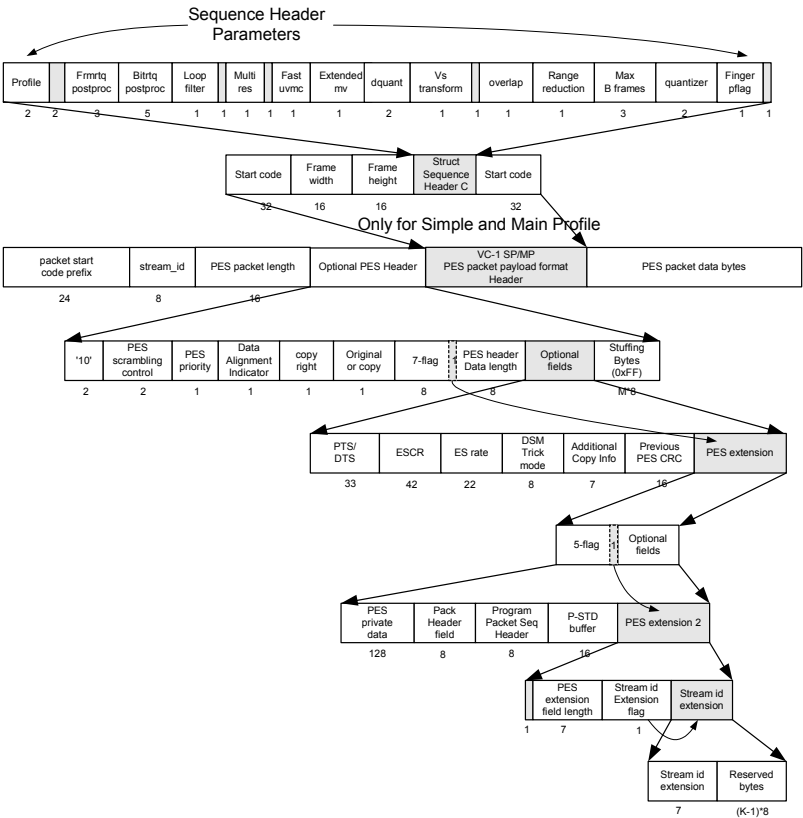


Figure 2-4 PES Syntax Diagram for VC-1

The data\_alignment\_indicator for VC-1 ESs shall follow the same semantics mentioned in Chapter 1, except that it applies to the Data Alignment sub-descriptor values defined in Table 2-5. In particular, the data\_alignment\_indicator in the PES header shall be set to “1” if there is Data Alignment sub-descriptor associated with the VC-1 ES in the PMT. If the data\_alignment\_indicator is set to “1” and there is no Data Alignment sub-descriptor associated with the VC-1 ES in the PMT, the default alignment type value shall be equal to 0x02. The data\_alignment\_indicator value “0” indicates that the alignment is unspecified. For SP/ MP, the value of the data\_alignment\_indicator field shall always be set to “1” and there shall not be any Data Alignment sub-descriptor associated with the VC-1 ES.



The PTS/ DTS are used in exactly the same manner as in MPEG-2. In particular, the values of the PTS/ DTS fields shall pertain to the first video AU that starts in the payload of the PES packet.

The `stream_id_extension` field for VC-1 ESs shall have any value in the range between 0x55 and 0x5f. The combination of `stream_id` and `stream_id_extension` unambiguously define the stream of PES packets carrying VC-1 video data.

A VC-1 access point is defined as follows:

- The first byte of a VC-1 Sequence header can be an access point if there is no Sequence start code preceding the Sequence header.
- The first byte of the Sequence start code can be an access point if a Sequence start code immediately precedes the Sequence header.

As explained in Chapter 1, the `discontinuity_indicator` is 1-bit field that indicates whether the discontinuity state is true for the current TS packet. After a continuity counter discontinues in a TS packet with VC-1 data, the first byte of ES data in a TS packet of the same PID shall be the first byte of a VC-1 access point or a VC-1 end-of-sequence start code followed by an access point.

All other data such as `random_access_point`, `elementary_stream_priority_indicator`, `splice_countdown`, `seamless_splice_flag`, `slice_type` are interpreted as explained in Chapter 1.

For VC-1 Simple or Main Profile ESs, a `VC-1_SPMP_PESpacket_PayloadFormatHeader( )` structure shall be present at the beginning of every AU. A `VC-1_SPMP_PESpacket_PayloadFormatHeader( )` shall always start with a `start_code` presenting a Sequence start code (value of “0x0000010f”) or a Frame start code (value of “0x0000010d”). The start code emulation prevention shall be applied to the bytes in the PES packet following the `start_code` field in the `VC-1_SPMP_PESpacket_PayloadFormatHeader( )` structure to protect the start code from occurring in any other locations in the PES packet payload. The start code emulation prevention mechanism is described in Annex E of SMPTE 421M.

The structure shown in Figure 2-5 below corresponds to the Sequence header for the VC-1 SP/ MP. There shall not be any VC-1\_SPMP\_PESpacket\_PayloadFormatHeader( ) structure at the beginning of any PES packet payload.

```
VC-1_SPMP_PESpacket_PayloadFormatHeader( ){
    start_code          (32 bit)      // bslbf
    if( start_code == 0x0000010f) {
        frame_width     (16 bit)      // uimbsf
        frame_height    (16 bit)      // uimbsf
        STRUCT_SEQUENCE_HEADER_C( ) (32 bit) //bslbf
        Start_code      (32 bit)      // 0x0000010d
    } else if (start_code == 0x0000010d) {
    }
}
```

Figure 2-5 Syntax for VC-1\_SPMP\_PESpacket\_PayloadFormatHeader( ) structure

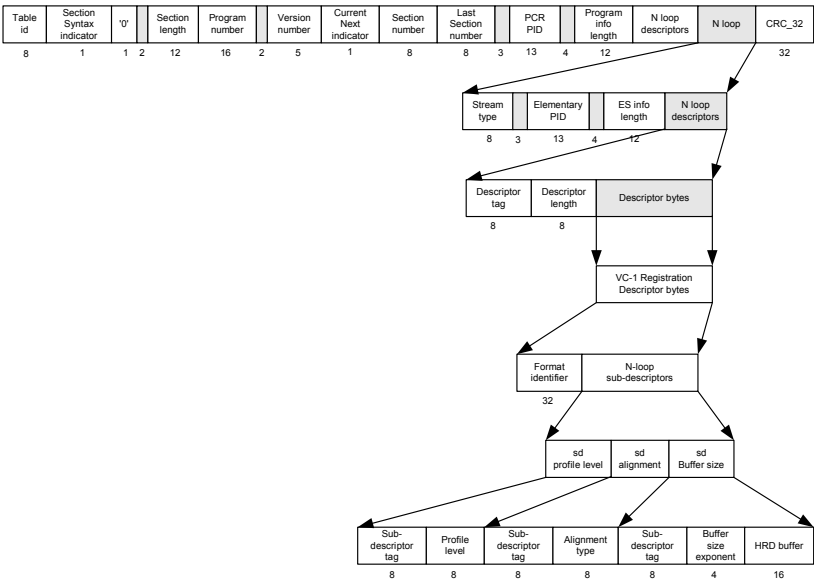


Figure 2-6 TS PM Section Diagram for VC-1

## Encapsulation of VC-1 in TS

In a TS, Programs are signaled using a collection of Tables transmitted cyclically and known as PSI (Program Specific Information). This was explained in detail in Chapter 1. Specifically, a Program Map Table (PMT) provides the Program details and specifies necessary information such as PID to find and decode the component ESs. Delivery of VC-1 ESs in MPEG-2 TSs shall be governed by a T-STD in MPEG-2 Systems provisions.

The Registration Descriptor in MPEG-2 Systems is designed to identify formats of “private” data uniquely and unambiguously. The `registration_descriptor()`, whose structure is originally defined for MPEG-2 Systems, resides in the inner descriptor loop of the MPEG-2 Program Element (PE) in the TS Program Map (PM) section corresponding to a VC-1 ES. The PM section is depicted in Figure 2-6. Figure 2-7 depicts the `registration_descriptor()`. The `format_identifier` is a number assigned/registered with the SC92 committee to eliminate any confusion for future private data.

```
registration_descriptor ( ){
    descriptor_tag          (8 bit)          // 0x05
    descriptor_length       (8 bit)          // uimsbf
    format_identifier       (32 bit)         // 0x56432d31
    for (i=0; i<K; i++) {
        sub-descriptor( )   (N*8)           //uimsbf
    }
}
```

**Figure 2-7 Syntax for Registration Descriptor**

The `sd_profile_level()` sub-descriptor signals the Profile and Level for the associated VC-1 ES. Syntax and semantics for this sub-descriptor are defined in Figure 2-8.

```
sd_profile_level ( ){
    subdescriptor_tag       (8 bit)          // 0x01
    profile_level           (8 bit)          // uimsbf
}
```

**Figure 2-8 Syntax for Profile and Level Sub-Descriptor**

The sub-descriptor( ) has a data structure as shown in Figure 2-8, Figure 2-9 and Figure 2-10. The first field of a sub-descriptor( ) shall be an 8-bit value known as the subdescriptor\_tag which identifies syntax and semantics for a particular sub-descriptor. Table 2-3 defines values of the subdescriptor\_tag.

**Table 2-3 Value for subdescriptor\_tag**

Assigned value	Description
0x00	Indicates a null sub-descriptor. A null sub-descriptor consists only of an 8-bit sub-descriptor_tag field
0x01	Profile and Level sub-descriptor defined for VC-1
0x02	Alignment sub-descriptor defined for VC-1
0x03	Buffer size sub-descriptor defines for VC-1
0x04~0xfe	SMPTE reserved for future applications
0xff	Indicates a null sub-descriptor. A null sub-descriptor consists only of an 8-bit sub-descriptor_tag field

An alignment sub-descriptor sd\_alignment( ) is used to define explicitly which type of alignment exists between the coded byte sequence and a PES packet. Syntax and semantics for this sub-descriptor are defined in Figure 2-9.

```
sd_alignment ( ){
    subdescriptor_tag      (8 bit)      // 0x02
    alignment_type         (8 bit)      // uimsbf
}
```

**Figure 2-9 Syntax for Profile and Level Sub-Descriptor**

```
sd_buffer_size( ){
    subdescriptor_tag      (8 bit)      // 0x02
    reserved               (8 bit)      // uimsbf
    buffer_size_exponent   (4 bit)      // uimsbf
    hrd_buffer              (16 bit)     // uimsbf
}
```

Figure 2-10 Syntax for Sd\_buffer\_size Sub-Descriptor

Table 2-4 Value for profile\_level field

Assigned value	Description
0x00	SMPTE reserved
0x11	Simple Profile, Low Level
0x12	Simple Profile, Medium Level
0x13~0x50	SMPTE reserved
0x51	Main Profile, Low Level
0x52	Main Profile, Medium Level
0x53	Main Profile, High Level
0x54~0x90	SMPTE reserved
0x91	Advanced Profile, Level L0
0x92	Advanced Profile, Level L1
0x93	Advanced Profile, Level L2
0x94	Advanced Profile, Level L3
0x95	Advanced Profile, Level L4
0x96~0xff	SMPTE reserved

**Table 2-5 Syntax for Alignment Sub-Descriptor**

Assigned value	Description
0x00	SMPTE reserved
0x01	Slice or Video AU
0x02	Video AU
0x03	Entry Point or Sequence
0x04	Sequence
0x05	Frame
0x06~0xff	SMPTE reserved

A buffer size sub-descriptor specifies the minimum size of the video ES buffer  $EB_n$  in TSs or  $B_n$  in PSs needed in the decoder to decode the ES conveyed in the MPEG-2 Program Element associated with this sub-descriptor. This descriptor is provided to allow receivers to check compatibility of their decoding capability against the decoding requirements for the ES. The buffer associated with the VC-1 Profile and Level shall be assumed if this sub-descriptor is not present and if no HRD parameters are specified in the Sequence header.

Conventionally defined descriptors in MPEG-2 Systems shall be used with VC-1 ES in certain cases. Such descriptors include Target Background Grid Descriptor, Video Window Descriptor, CA Descriptor, ISO 639 Language Descriptor, Multiplex Buffer Utilization Descriptor, Smoothing Buffer Descriptor, Copyright Descriptor, Maximum Bitrate Descriptor, Private Data Indicator Descriptor, IBP Descriptor, and STD Descriptor.

The size of buffer  $TB_n$ , known as  $TBS_n$ , shall be equal to 512 bytes, which is exactly the same as in MPEG-2 Systems. In addition, the buffer size MBS and rate  $R_x$  are similar as those of MPEG-2 Systems.

When there is no data in  $TB_n$  :

$$Rx_n = 0. \quad (2-1)$$

Otherwise,

$$Rx_n = 1.2 \times R_{\max} [profile, level]. \quad (2-2)$$

The multiplexing buffer size  $MBS_n$  is defined as follows:

$$MBS_n = BS_{mux} + BS_{oh} \quad (2-3)$$

where PES packet overhead buffering

$$BS_{oh} = (1 / 750) \text{sec} \times R_{\max} [profile, level], \quad (2-4)$$

and additional multiplex buffering

$$BS_{mux} = (0.004 \text{sec}) \times R_{\max} [profile, level]. \quad (2-5)$$

Note that the transfer data from MB to EB shall be governed by a leak method only. Use of a leak method shall be signaled via one of the following two methods: 1. There is no MPEG-2 STD Descriptor present in the inner descriptor loop of the MPEG-2 PE in the TS\_program\_map\_section corresponding to the VC-1 ES. 2. An MPEG-2 STD Descriptor is present in the inner descriptor loop of the MPEG-2 PE in the TS\_program\_map\_section corresponding to the VC-1 ES and the leak\_valid flag has the value of “1.”

The leak method transfers data from MB to EB using the leak rate  $Rbx$  as:

$$Rbx_n = R_{\max} [profile, level] . \quad (2-6)$$

The default size  $EBS_n$  of the ES buffer shall be the  $VBV_n[profile, level]$  associated with the profile and level of the VC-1 ES. The profile and level may be defined by the field profile\_level in the sd\_profile\_level( ) sub-descriptor. With an assumption that the incoming service delivery rate  $R$  is known through ES\_rate field in PES header, a receiver may opt to use a smaller ES buffer for its internal representation of the HRD. However, the size of ES buffer shall always be equal to the

minimum buffer value  $B_{\min}$  specified by the Generalized Hypothetical Reference Decoder for rate  $R$  as follows (in units of bits):

$$EBS_n[k] = (hrd\_buffer[k] + 1) \times 2^{(buffer\_size\_exponent+4)} \quad (2-7)$$

and the associated rate  $R[k]$  may be computed from the  $hrd\_rate[k]$  and the  $bit\_rate\_exponent$  fields as follows (bits/sec):

$$R[k] = (hrd\_rate[k] + 1) \times 2^{(bit\_rate\_exponent+6)} \quad (2-8)$$

### Encapsulation of VC-1 in PS

Delivery of VC-1 Elementary Streams in MPEG-2 PSs shall be governed by a P-STD in MPEG-2 Systems provisions. The stream type value 0xea and the use of registration descriptor and sub-descriptors defined in the previous section shall also be applicable to the carriage of a VC-1 ES in an MPEG-2 PS. The only difference is that in the case of an MPEG-2 PS, the structures where these fields are used is the PSM-PES as opposed to the PM section in TSs. Headers and payload formats shall be identical to the format described in MPEG-2 Systems. PSM-PES for VC-1 is depicted in Figure 2-11.

The input buffers  $B_n$  in the P-STD shall not overflow. Furthermore, they shall not underflow except where the value of  $HRD\_PARAM\_FLAG$  field in the Sequence header of the VC-1 ES is equal to "0", in which case the codec operates in variable delay mode as described in Chapter 3. Data enters the P-STD at the rate specified by the value of the field  $program\_mux\_rate$  in the Pack header. The PES packet data bytes from VC-1 ES number  $n$  are passed to the input  $B_n$ . Unless specified by the  $P\_STD\_buffer\_scale$  and  $P\_STD\_buffer\_size$  field in the PES header, the input buffer sizes  $B_n$  are equal to the sum of  $vbv\_max[profile, level]$  value of the VC-1 ES and the value  $BS_{add}$  defined in ISO 13818-1 as:

$$BS_{add} \leq \text{Max}[6 \times 1024, R_{v\max} \times 0.001] \text{ bytes} \quad (2-9)$$

With an assumption that the incoming service delivery rate  $R$  is known through  $ES\_rate$  field in PES header, a receiver may opt to use a smaller ES buffer for its internal representation of the HRD. However, the



size of ES buffer shall always be equal to the minimum buffer value  $B_{\min}$  specified by the Generalized Hypothetical Reference Decoder for rate  $R$  as follows (in units of bits):

$$BS_n[k] = (hrd\_buffer[k] + 1) \times 2^{(buffer\_size\_exponent + 4)} \tag{2-10}$$

and the associated rate  $R[k]$  may be computed from the  $hrd\_rate[k]$  and the  $bit\_rate\_exponent$  fields as follows (bits/sec):

$$R[k] = (hrd\_rate[k] + 1) \times 2^{(bit\_rate\_exponent + 6)} \tag{2-11}$$

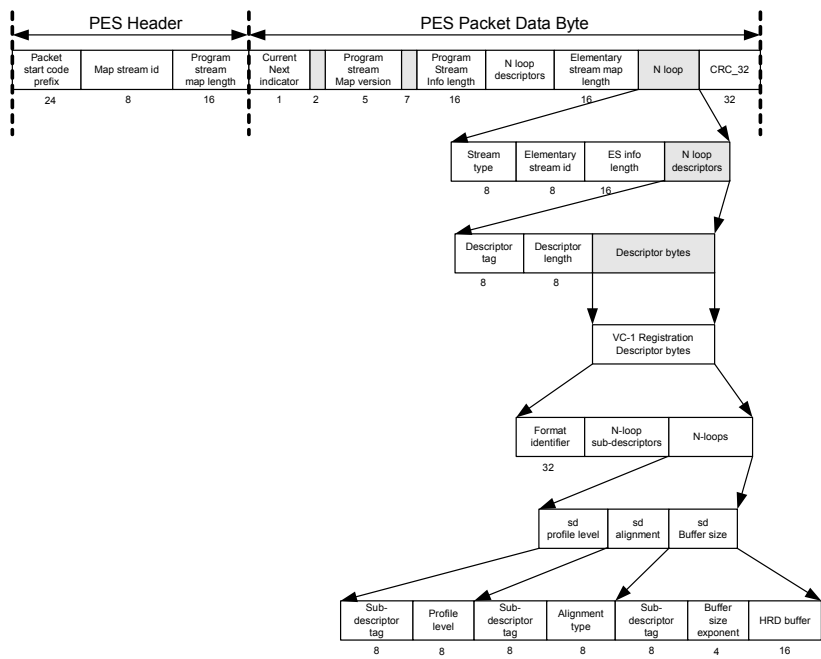


Figure 2-11 PSM-PES Diagram for VC-1

## 2.3 H.264 Syntax Hierarchy in Bitstreams

### H.264 Standard

MPEG-4 Part 10 or H.264 video compression standard was developed to enhance compression performance over the current de facto standard MPEG-2 that was developed about 10 years ago primarily for digital TV systems with interlaced video coding. H.264 is known to achieve a significant improvement in rate-distortion efficiency compared with existing standards, and is designed for broadcast TV over cable/DSL/satellite, IP set-tops and HD-DVD/ Bluray-DVD recorders. HD-DVD adopts only High Profile of H.264, while Bluray-DVD uses both Main Profile and High Profile [JVT:H.264, richardson:H.264].

### Key Compression Tools for H.264 Video

H.264 is based on motion compensated transform coding. Unlike VC-1, YUV4:4:4, YUV4:2:2 and YUV4:2:0 are defined as input formats for H.264. For YUV4:4:4, a Residual Color Transform (RTC) tool was originally designed to get a better compression efficiency due to color space change in color representation. After some reconsideration and confusion over its value, the RCT and the “High 4:4:4” Profile have actually been removed from the latest standard. The original High 4:4:4 Profile has been replaced by the “High 4:4:4 Predictive” Profile that has following properties relative to the prior High 4:4:4 Profile [sullivan:new, lee:improved]:

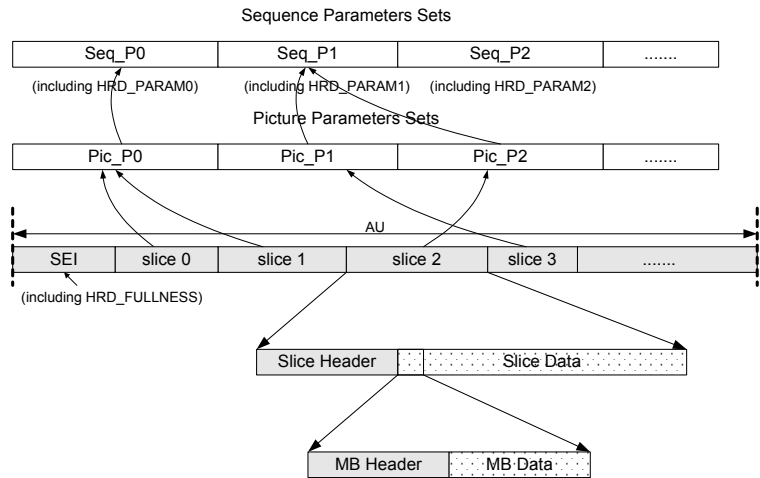
- Increased bit depth (upto 14 bits per sample)
- No RTC
- Improved transform-bypass lossless coding
- Processing of Chroma channels in a similar way to the way Luma channels are processed (i.e., in terms of MC, interpolation and Intra prediction)
- Two modes of operation – the “Common” and “Separate” modes control

The key concept in H.264 is the Slice. The information of two higher layers over Slice is Sequence parameters set and Picture parameters set. Note that those are parameters set to be used either directly or indirectly by

each Slice as shown in Figure 2-12. Sequence parameter ID ranges from 0 to 31 while Picture parameter ID ranges from 0 to 255.

There is no GOP structure since referencing order is decoupled from coding order in H.264. But there is an imaginary Picture structure that is composed of one or more Slices. Here, the term “imaginary” means that there is no Picture layer in the bitstream structure, but a picture is generated through the Slice decoding process. The characteristic of the picture is declared with `primary_pic_type`, where the value indicates what kind of slices are in the primary coded picture. Note that many different types of Slices can constitute a single picture. For example, a primary coded picture can be composed of I, P and B Slices. There are 10 Slice types – two I (Intra), two P (Predicted), two B (Bi-predictive), two SP (Switching P) and two SI (Switching I) as shown in Table 1-8. Each coding mode slice has two different numbers assigned to it since a picture in H.264 can be composed of multiple slices with different slice types. An example includes I slices, P slices, B slices mixed to constitute a single picture.

In H.264, the geometrical structure of Slices is very generic. There are six pre-defined Slice geometrical structures—Interleaved, Dispersed, Foreground and Background, Box-out, Raster scan and Wipe as discussed in Chapter 1. There is a 7th option to define any kind of Slice shape explicitly. Note that the geometrical structure of a Slice can help visual recovery of lost data. For example, interleaving of two slices improves visual perceptual when one of them is lost as shown in Chapter 1.



**Figure 2-12 Syntax Hierarchy for H.264**

Multiple reference pictures can be used in motion compensation. The syntax element, `num_ref_frames`, specifies the maximum total number of short-term and long-term reference frames, and it ranges from 0 to 16. In addition, B frames can be a reference in H.264. References are stored in a DPB (Decoded Picture Buffer) in both the encoder and the decoder. The encoder and the decoder maintain a list of previously coded pictures – reference picture list0 for P Slices and reference picture list0 and list1 for B Slices. These two lists can contain short-term and long-term pictures, where either list0 or list1 can be used for past coded pictures, while the other one can be used for future coded pictures. There is pre-defined index management called “sliding window” memory control. For example, a coded picture is reconstructed by the encoder and marked as a short-term picture that is identified with its `PicNum` for list0. However, the reference list0 can be also controlled with “adaptive” memory control in a customized manner by an encoder. For example, if a certain pattern is continuously used as a background video pattern during 10 minutes in a video, an encoder may want to keep it as a long-term picture reference. That kind of encoder decision about list management can be written in the bitstreams with two syntax elements – reference picture list reordering syntax and decoded reference picture marking syntax. These could be thought of as on the fly memory control commands in the bitstreams.

Also, an encoder can send an IDR (Instantaneous Decoder Refresh) coded picture made up of I- or SI-Slices to clear all DPB contents. This means that all subsequent transmitted slices can be decoded without reference to any frame decoded prior to the IDR picture.

A key compression tool in H.264 is adaptive size MC (motion compensation) with a small 4x4 transform. Note that the 8x8 transform has been recently introduced. The size of MC is described in two levels – MB partition and MB sub-partition. The MB partition size can be broken down into sizes of 16x16, 16x8, 8x16 and 8x8. If 8x8 size is chosen in MB partition, each MB sub-partition (8x8 block) can be broken into sizes of 8x8, 8x4, 4x8 or 4x4. The color components, Cb and Cr, have the same, half, or quarter the size of luma components based on video formats such as YUV4:4:4, YUV4:2:2 or YUV4:2:0, but each chroma block is partitioned in the same way as the luma component. Note that the resolution of MVs in luma is quarter-pel, while that of derived MVs in chroma is 1/8-pel. A specific 6 tap-FIR filter and bi-linear filter are used for interpolation. Also, MVs can point past picture boundaries with conceptually extended padding.

The 4x4 transform is a novel integer transform with the following two aspects: 1. It is an integer transform, where all operations can be carried out with integer arithmetic. 2. Mismatch between encoders and decoders doesn't occur. The 4x4 transform de-correlates well for high frequency patterns. As a fact, small size transform performs well in random-looking areas such as residual data. Consequently, it reduces the “ringing” artifact. Note that directional spatial prediction for Intra coding is introduced in H.264 to eliminate redundancy resulting in random-looking areas, even in I Slices. Directional spatial prediction has a couple of options to best de-correlate intra block data based on previously-decoded parts of the current pictures. Hierarchical transform is applied on DC values of 4x4 transforms in 16x16 Intra mode MB and chroma blocks, since R-D characteristics with longer basis transform are better than that of shorter basis transform over a smooth area. There are improvements in skipped mode and direct mode in motion compensation over existing standards in syntax representation to suppress space for MV data.

The weighted prediction is extended to handle fading scene scenarios by providing weights and offsets. The direct mode uses bi-directional prediction with derived MVs based on the MV of the co-located MB in the subsequent picture reference one as shown in Figure 6-4. Note that a

default weight based on geometrical division can be overridden with explicit weights to improve prediction performance. One important difference between H.264 and VC-1 in terms of direct mode technique is that actual blending for a predictor happens based on weights in H.264, while pixel-wise average for a predictor is obtained in VC-1 with weights only being considered for MV computation. Multi-hypothesis mode is worthwhile to note since a joint estimation for a predictor is possible through it. While bi-directional prediction type only allows a linear combination of a forward/ backward prediction pair, any combination of references is possible (i.e., backward/ backward combination or forward/ forward combination).

ILF is used to reduce blocking artifacts. H.264 ILF dynamically adapts the length of FIR filter tap. The decision of filter type and taps is dependent on how serious the blocky effect is; the standard provides ways to measure it. Based on the measurement, 5/4/3 tap filters are applied on horizontal and vertical block boundaries. The handling of 4x4 transform block coefficients is quite different compared with other standards. 3D RLC or 2D RLC is not used, but Token, Sign, Level, run\_before groups are coded in the inverse scanning order (the last to the first coefficient).

H.264 standard suggests two different types of entropy coding – CA-VLC and CA-BAC. In CA-VLC, Huffman (VLC) tables are chosen based on neighboring contexts or historical contexts. Examples are following: the choice of Token VLC tables is dependent on the number of non-zero coefficients in upper and left-hand previously coded blocks. The choice of Level VLC computation rule is triggered by the present Level value. If a present Level value is over a certain threshold, new VLC computation rules for Level are used to compress Level symbols.

When CA-BAC is selected for entropy coding, the following four consecutive stages follow: 1. Binarization, 2. Context-model selection, 3. Arithmetic encoding, 4. Probability update. Generally speaking, CA-BAC outperforms CA-VLC. The reason for this is that arithmetic coding is block entropy coding (the block size is the size of overall input bits until another initialization), while Huffman (VLC) coding is scalar entropy coding. An additional benefit for using arithmetic coding is to dynamically adapt the probability model as the probability of symbols change w. r. t. time. There can be many adaptation methods. One is to use accumulation of Symbols up to present for the model probability. However, this method doesn't capture local statistics, but only considers the long-term average.

An excellent method to capture local statistics is to use neighboring contexts as is done in CA-BAC. Note that the context models are initialized depending on the initial value of the Qp. The Binarization method and Context-model selection for each Syntactic element are specified in the standard. A multiplication-free method for the arithmetic coding is used in H.264 based on a small set of representative values of range and probability of symbols. The range is quantized into four distinct values and the probability of symbols is quantized into 64 distinct values. This allows a pre-computed table approach in the arithmetic coding, thus making it multiplication-free.

There are many coding options for each MB as shown in Figure 2-13. There are two transforms choices in each 8x8 block, where the same transform is required to apply – either 4x4 and 8x8. Note that 8x8 transform is only defined for High Profile. The choice between transform sizes is basically determined by the encoder to take advantage of statistics of input signals in High Profile. There are many MC modes in terms of size where the partition is a combination of sub-pattern of a 16x16 block and a 8x8 block as shown in the Figure 2-13. The size can vary from 4x4 to 16x16, thus meaning that the number of MVs can vary from 32 to 1 for a MB. The choice among MC sizes is basically made by the encoder to take advantage of statistics of the input signals. Typically, a large area is effective when a rigid object is moving (uniform motion in the area), while a small area is so when tearing object (no object) and/or overlapped objects are moving (random motion in the area).

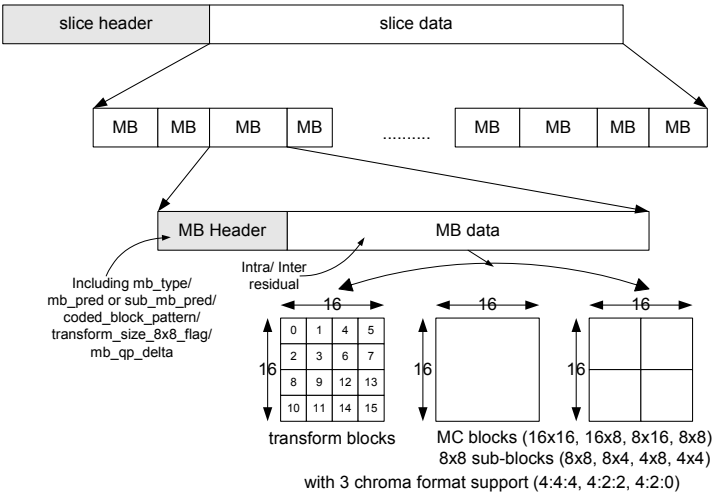


Figure 2-13 Slice Hierarchy for H.264

H.264 Video Specific Semantics and the Syntax

There are five levels of information in the H.264 video bitstream syntax – Sequence Parameter Set (SPS), Picture Parameter Set (PPS), Slice level, MB and Block. SPS contains basic parameters such as profile and level data, seq\_parameter\_set\_id (identifies sequence parameter set that is referred to), MaxFrameNum, picture order related parameters, num\_ref\_frames (maximum total number of short-term and long-term reference frames), frame\_mbs\_only\_flag, direct\_8x8\_inference\_flag (specifies the method used in the derivation process of luma MVs for B\_Skip, B\_Direct\_16x16, B\_Direct\_8x8), etc.

PPS contains information about pic\_paramter\_set\_id (identifies the picture parameter set that is referred to in the Slice header), seq\_parameter\_set\_id, entropy\_coding\_mode\_flag (either CA-BAC or CA-VLC), slice definition related parameters, maximum reference index data for reference list0 or list1, weighted\_pred\_flag (whether weighted prediction is applied to P and SP Slices), weighted\_bipred\_idc (weighted bi-prediction mode applied to B Slices), Qp related data, deblocking\_filter\_control\_present\_flag (syntax element controlling the characteristics of the deblocking filter is present in the Slice header), etc .



In the Slice header, there is information about slice attributes, extra display order related parameters, `direct_spatial_mv_pred_flag`, override information about reference index for `list0` and `list1`, deblocking filter related data, etc. Slice data contains `mb_skip` related data and MB layer information. The MB layer contains `mb_type`, `coded_block_pattern`, `mb_qp_delta`. Based on `mb_type` information, `mb_pred` or `sub_mb_pred` data are put into the bitstream. Then, residual data is appended. The `mb_pred` and `sub_mb_pred` information contain reference picture indices and MV data. Note that each 8x8 block area must refer to a single reference picture even though different MVs can be used for different 4x4 blocks in a single 8x8 block. There are two residual data syntax flows defined – one for CA-VLC and the other for CA-BAC.

## **H.264 Profiles/ Tools**

Baseline Profile targets low-rate Internet video conferencing and low-complexity applications such as mobile communications or play back of media in personal digital assistants.

The Main Profile targets broadcast applications such as digital TV and local video storage for PC play back. Bluray-DVD can contain Main Profile contents.

The Streaming Profile targets Internet applications such as streaming, movie delivery via IP, or TV/ VOD over IP.

The High Profile targets high quality applications such as studio editing, HD-DVD, Bluray-DVD, or DVB (digital video broadcast for European TV).

The High 10 Profile targets future Consumer Electronics products supporting for up to 10 bits per sample of decoded picture precision.

The High 4:2:2 Profile targets professional applications that use interlaced video supporting for the 4:2:2 Chroma sub-sampling format and 10 bits per sample of decoded picture precision.

The High 4:4:4 Predictive Profile targets professional applications supporting for the 4:4:4 Chroma sub-sampling format and 14 bits per sample of decoded picture precision.

Each class of bitstreams contains tool sets defined in Table 2-6 and Table 2-7 and parameters set defined in Table 2-8.

**Table 2-6 H.264 Profiles and Tools in Original H.264**

Tool Options	Baseline Profile	Main Profile	Extended Profile
I and P Slices	x	x	x
CA-VLC	x	x	x
CA-BAC		x	
B Slices		x	x
Interlaced Coding (PAFF, MBAFF)		x	x
Enh. Err. Resil. (FMO, ASO, RS)	x		x
Further Enh. Err. Resil. (DP)			x
SI and SP Slices			x

**Table 2-7 H.264 Profiles and Tools in New H.264 Profiles/Amendment**

Tool Options	High	High 10	High 4:2:2	High 4:4:4 Predictive
Main Profile Tools	x	x	x	x
4:2:0 Chroma Format	x	x	x	x
8 bit Sample Bit Depth	x	x	x	x
8x8 vs. 4x4 Transform Adaptivity	x	x	x	x
Quantization Scaling Matrices	x	x	x	x
Separate Cb and Cr QP Control	x	x	x	x
Monochrome Video Format	x	x	x	x
9 and 10 bit Sample Bit Depth		x	x	x
4:2:2 Chroma Format			x	x
14 bit Sample Bit Depth				x

4:4:4 Chroma Format				x
Improved Transform-bypass Lossless coding				x
3 Separate Color Plan coding				x

Additionally, the standard now contains four all-Intra Profiles, which are defined as simple subsets of other corresponding Profiles. These are for professional applications. The reason to define these profiles is to reduce decoder complexity by restricting the coding mode and the entropy coding method. This direction was taken based on the fact that H.264 Intra coding many times outperforms the state of the art in still image coding schemes.

The High 10 Intra Profile is defined when the High 10 Profile is constrained to all-Intra use.

The High 4:2:2 Intra Profile is defined when the High 4:2:2 Profile is constrained to all-Intra use.

The High 4:4:4 Intra Profile is defined when the High 4:4:4 Profile is constrained to all-Intra use.

The CA-VLC 4:4:4 Intra Profile is defined when the High 4:4:4 Predictive Profile is constrained to all-Intra use and to CA-VLC.

**Table 2-8 H.264 Levels and Limitations**

Level Number	MB/s	MB/f	DPBmax (1024 bytes for 4:2:0)	Rmax (1000 bps-VCL, 1200 bps-NAL)	CPBmax (1000 bps-VCL, 1200 bps-NAL)	MV [Vertical]	CRmin	Max number of MVs per 2 consecutive MBs (MaxMVSPer2Mb)
1	1,485	99	148.5	64	175	[-64,63 ¾]	2	-
1b	1,485	99	148.5	128	350	[-64,63 ¾]	2	-
1.1	3,000	396	337.5	192	500	[-128,127 ¾]	2	-
1.2	6,000	396	891.0	384	1,000	[-128,127 ¾]	2	-

1.3	11,880	396	891.0	768	2,000	$[-128,127\frac{3}{4}]$	2	-
2	11,880	396	891.0	2,000	2,000	$[-128,127\frac{3}{4}]$	2	-
2.1	19,800	792	1,782.0	4,000	4,000	$[-256,255\frac{3}{4}]$	2	-
2.2	20,250	1,620	3,037.5	4,000	4,000	$[-256,255\frac{3}{4}]$	2	-
3	40,500	1,620	3,037.5	10,000	10,000	$[-256,255\frac{3}{4}]$	2	32
3.1	108,000	3,600	6,750.0	14,000	14,000	$[-512,511\frac{3}{4}]$	4	16
3.2	216,000	5,120	7,680.0	20,000	20,000	$[-512,511\frac{3}{4}]$	4	16
4	245,760	8,192	12,288.0	20,000	25,000	$[-512,511\frac{3}{4}]$	4	16
4.1	245,760	8,192	12,288.0	50,000	62,500	$[-512,511\frac{3}{4}]$	2	16
4.2	522,240	8,704	13,056.0	50,000	62,500	$[-512,511\frac{3}{4}]$	2	16
5	589,824	22,080	41,400.0	135,000	135,000	$[-512,511\frac{3}{4}]$	2	16
5.1	983,040	36,864	69,120.0	240,000	240,000	$[-512,511\frac{3}{4}]$	2	16

There are several levels for each of the profiles. Each level limits the video resolution, frame rate, HRD bit rate, HRD buffer requirements, and the motion vector range. These limitations are defined in Table 2-8.

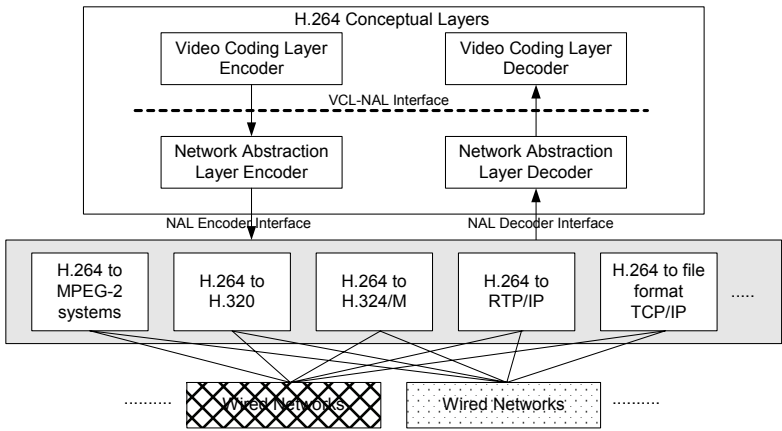
2.4 H.264 Encapsulation in MPEG-2 Systems

Amendment 3 of ITU-T Recommendation H.222.0| ISO/IEC 13818-1:2000/Amd.3:2004 recommends H.264 bitstream encoding provisions that define a minimum set of rules for the carriage of an H.264 elementary stream in an MPEG-2 Transport Stream with additional intention to provide a generic means of carrying an H.264/ AVC video elementary stream in an MPEG-2 Program Stream as used by the DVD Forum [ISO:MPEG2systems.amd]. This section discusses H.264 bitstream encapsulation in MPEG-2 Systems.

NAL and VCL

In H.264, the Network Abstraction Layer (NAL) is designed to be self-contained for extensive encapsulation methods so that the coding

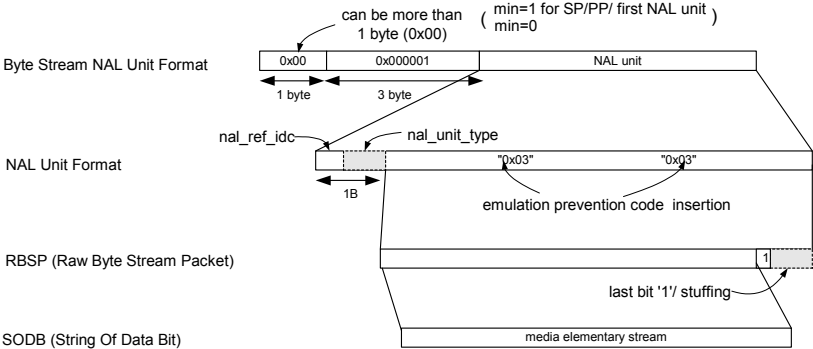
layer can be separated from delivery/ system encapsulation mechanism as shown in Figure 2-14.



**Figure 2-14 Network Abstraction Layer and Interface**

To this end, NAL and Video Coding Layer (VCL) layers are devised. VCL is designed to efficiently represent the video content, while NAL encapsulates the VCL representation of video with header information in such a way that a variety of transport layers and storage media can easily adopt compressed contents. A NAL unit specifies both byte-stream and packet-based formats. Byte-stream format as shown in Figure 2-15 is used for bitstream-like representation as in MPEG-2, while packet-based format targets applications with coded data carried in an internet-like transport protocol. NAL unit header 1B is composed of 1-bit “forbidden\_zero\_bit,” 2-bit “nal\_ref\_idc” and 5-bit “nal\_unit\_type.” The forbidden\_zero\_bit indicates whether the NAL unit has errors – “1” means to be in error. The nal\_ref\_idc implies whether the NAL unit is disposable. The nal\_unit\_type provides a peeking function into the payload about NAL unit types such as SEI message, Parameter Sets, VCL data, etc. Packet-based systems can employ NAL units directly.

NAL units are classified into VCL NAL and non-VCL NAL units. The VCL NAL units contain the data that represents the values of the samples in the video pictures, and the non-VCL NAL units contain additional information such as timing information.

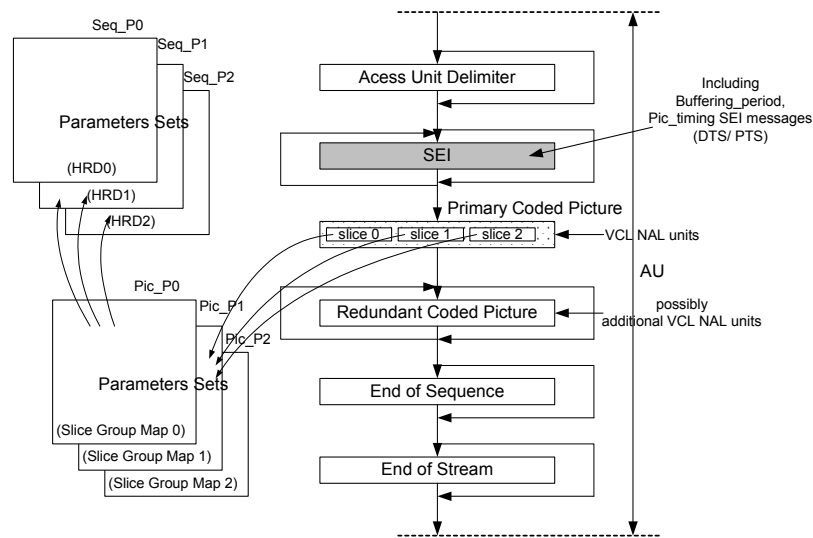


**Figure 2-15 NAL Unit Syntax for MPEG-2 Systems**

**Access Unit and SEI in H.264**

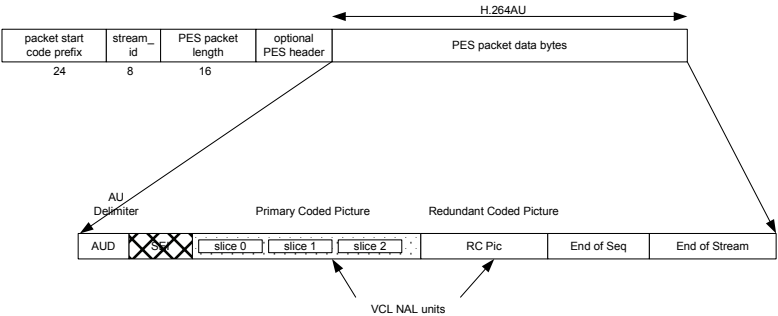
A set of NAL units in a specified form as in Figure 2-16 is referred to as an AU in H.264. The decoding of each AU results in one decoded picture. Each AU contains a set of VCL NAL units that together compose a Primary Coded Picture (PCP). It is mandatory to be pre-fixed with an AU delimiter to aid in locating the start of the AU.

Supplemental Enhancement Information (SEI) containing data such as picture timing information may also precede the PCP. The PCP consists of a set of VCL NAL units containing slices or slice data partitions that represent the samples of the video picture. The Redundant Coded Picture (RCP) is composed of additional VCL NAL units that may contain redundant representations of areas of the same video picture. Decoders are not required to decode redundant coded pictures when they are present. Decoders can decode them when any problems occur at PCP decoding.



**Figure 2-16 Access Unit in H.264**

If the coded picture is the last picture of the coded video sequence (a sequence of pictures that is independently decodable with one sequence parameter set), an End of Sequence NAL unit may be present to indicate the end of the sequence. Note that in order to activate a different SPS, one must start a new coded video sequence. In other words, one movie can be broken down to a couple of segments each of which is defined as a separate coded sequence. If the coded picture is the last coded picture in the entire NAL unit stream, and End of Stream NAL unit may be present to indicate that the stream is ending. Figure 2-17 depicts PES payload format for H.264.



**Figure 2-17 PES Syntax Diagram for H.264**

Annex D of H.264 describes SEI. Among the different types are Buffering\_period( payloadSize) and Pic\_timing( payloadSize) that contain timing information of pictures. Figure 2-18 defines Buffering\_period SEI message syntax, while Figure 2-19 defines Pic\_timing SEI message syntax.

The explanation about Buffering\_period SEI message syntax as shown in Figure 2-18 is as follows: Seq\_parameter\_set\_id specifies the SPS that contains the sequence HRD attributes. The value of Seq\_parameter\_set\_id is equal to the value of Seq\_parameter\_set\_id in the PPS referenced by the PCP associated with the Buffering\_period SEI message.

Initial\_cpb\_removal\_delay[ SchedSelIdx] specifies the delay for the SchedSelIdx-th Coded Picture Buffer (CPB) between the time of arrival in the CPB of the first bit of the coded data associated with the AU of the Buffering period SEI message and the time of removal from the CPB of the coded data associated with the same AU. It is for the first buffering period after HRD initialization. The syntax element has a length in bits given by initial\_cpb\_removal\_delay\_length\_minus1 + 1. It is in units of a 90 kHz clock (a.k.a., counter tick).

The initial\_cpb\_removal\_delay\_offset[ SchedSelIdx] is used for the SchedSelIdx-th CPB in combination with the Cpb\_removal\_delay to specify the initial delivery time of coded AU to the CPB. The syntax element has a length in bits given by initial\_cpb\_removal\_delay\_length\_minus1 + 1. It is in units of a 90 kHz clock. This syntax element is not used by decoders and is needed only for



delivery scheduling at Hypothetical Stream Scheduler (HSS) specified in Annex C.

```

Buffering_period( payloadSize ){
    Seq_parameter_set_id
    If( NalHrdBpPresentFlag){
        For( SchedSelIdx=0; SchedSelIdx <=
            Cpb_cnt_minus1; SchedSelIdx ++){
            Initial_cpb_removal_delay[SchedSelIdx]
            Initial_cpb_removal_delay_offset[SchedSelIdx]
        }
    }
    If( VclHrdBpPresentFlag){
        For( SchedSelIdx=0; SchedSelIdx <=
            Cpb_cnt_minus1; SchedSelIdx ++){
            Initial_cpb_removal_delay[SchedSelIdx]
            Initial_cpb_removal_delay_offset[SchedSelIdx]
        }
    }
}

```

**Figure 2-18 Buffering\_period SEI Message Syntax**

The explanation about Pic\_timing SEI message syntax as shown in Figure 2-19 is as follows: Cpb\_removal\_delay specifies how many clock ticks to wait after removal from the CPB of the AU associated with the most recent buffering period SEI message before removing from the buffer the AU data associated with the picture timing SEI message. The syntax element has a length in bits given by `cpb_removal_delay_length_minus1 + 1`. This value is also used to calculate the earliest possible time of arrival of AU data into the CPB for the HSS specified in Annex C. The clock tick is defined in VUI as:

$$t_c = \frac{\text{num\_units\_in\_tick}}{\text{time\_scale}} \quad (2-12)$$

For example,  $t_c = 1/29.97$  (sec) when `num_units_in_tick=1001` and `time_scale = 30000`. During the period of  $t_c$ , the Time Stamp counter increases 3003 based on a 90kHz clock. This setting is most likely used with `fixed_frame_rate_flag=0`.

```

Pic_timing( payloadSize ){
    If( CpbDpbDelaysPresentFlag ){
        Cpb_removal_delay
        Dpb_output_delay
    }
    If( pic_struct_present_flag ){
        Pic_struct
        For( I=0; I<NumClockTS; I++){
            Clock_timestamp_flag[I]
            If( clock_timestamp_flag[I]){
                Ct_type
                Nuit_field_based_flag
                Counting_type
                Full_timestamp_flag
                Discontinuity_flag
                Cnt_dropped_flag
                N_frames
            }
            If( full_timestamp_flag ){
                Seconds_value
                Minutes_value
                Hours_value
            } else {
                seconds_flag
                if( seconds_flag ){
                    seconds_value
                    minutes_flag
                    if( minutes_flag ){
                        minutes_value
                        hours_flag
                        if( hours_flag ){
                            hours_value
                        }
                    }
                }
            }
        }
    }
    If( time_offset_length>0 )
        Time_offset
}
}
}

```

Figure 2-19 Pic\_timing SEI Message Syntax

For example,  $t_c = 1 / 59.94$  (sec) when num\_units\_in\_tick=1001 and time\_scale = 60000. This setting is most likely used with fixed\_frame\_rate\_flag=1 that covers certain broadcast video (i.e., Pic\_struct=0, 3, 4, 5, 6 etc. in Table 2-9). Note that fixed\_frame\_rate\_flag=1 scenarios are built on the assumption that  $1/t_c$  is twice the frame rate.

Dpb\_output\_delay specifies how many clock ticks to wait after removal of an AU from CPB before the decoded picture can be output from the DPB. This value is used to compute the DPB output time of the picture.

Pic\_struct indicates whether a picture should be displayed as a frame or one or more fields according to Table 2-9 below. Frame doubling (7) indicates that the frame should be displayed two times consecutively, and frame tripling (8) indicates that the frame should be displayed three times consecutively.

**Table 2-9 Interpretation of Pic\_struct**

Value	Indicated display of picture	Restrictions	NumClockTS
0	Frame	Field_pic_flag shall be 0	1
1	Top field	Field_pic_flag shall be 1, Bottom_pic_flag shall be 0	1
2	Bottom field	Field_pic_flag shall be 1, Bottom_pic_flag shall be 1	1
3	Top field, bottom field, in that order	Field_pic_flag shall be 0	2
4	Bottom field, top field, in that order	Field_pic_flag shall be 0	2
5	Top field, bottom field, top field repeat, in that order	Field_pic_flag shall be 0	3

6	Bottom field, top field, bottom field repeat, in that order	Field_pic_flag shall be 0	3
7	Frame doubling	Field_pic_flag shall be 0, Fixed_frame_rate_flag shall be 1	2
8	Frame tripling	Field_pic_flag shall be 0, Fixed_frame_rate_flag shall be 1	3
9..15	reserved		

**Table 2-10 Definition of counting\_type Values**

Counting_type value	Interpretation
0	No dropping of N_frames count values and no use of time_offset
1	No dropping of N_frames count values
2	Dropping of individual zero values of N_frames count
3	Dropping of individual MaxFPS-1 values of N_frames count
4	Dropping of two lowest (value 0 and 1) N_frames counts when Seconds_value is equal to 0 and Minutes_value is not an integer multiple of 10.
5	Dropping of unspecified individual N_frames count values
6	Dropping of unspecified numbers of unspecified N_frames count values
7..31	reserved

NumClockTS is determined by Pic\_struct and it specifies that there are up to NumClockTS sets of Time Stamp information for a picture, as specified by clock\_timestamp\_flag[I] for each set. The sets of Time Stamp information apply to the field(s) or the frame(s) associated with the picture by Pic\_struct. The contents of the Time Stamp syntax elements indicate a time of origin, capture, or alternative ideal display. This indicated time is computed as:

$$\text{ClockTimeStamp} = ((\text{Hours\_value} \times 60 + \text{Minutes\_value}) \times 60 + \text{Second\_value}) \times \text{time\_scale} + \text{N\_frames} \times (\text{num\_units\_in\_tick} \times (1 + \text{Nuit\_field\_based\_flag})) + \text{Time\_offset} \quad (2-13)$$

in units of clock ticks of a clock with clock frequency equal to time\_scale Hz, relative to some unspecified point in time for which ClockTimeStamp is equal to 0. For example, with num\_units\_in\_tick=1001 and time\_scale=30000:

$$\text{ClockTimeStamp} = ((\text{Hours\_value} \times 60 + \text{Minutes\_value}) \times 60 + \text{Second\_value}) \times 30000 + \text{N\_frames} \times (1001 \times (1 + \text{Nuit\_field\_based\_flag})) + \text{Time\_offset}. \quad (2-14)$$

Output order and DPB output timing are not affected by the value of ClockTimeStamp.

Ct\_type indicates the scan type (interlaced or progressive) of the source material. Counting\_type specifies the method of dropping values of the N\_frames as specified in Table 2-10, while Cnt\_dropped\_flag specifies the skipping of one or more values of N\_frames using the counting method specified by Counting\_type.

Discontinuity\_flag indicates whether the difference between the current value of ClockTimeStamp and the value of ClockTimeStamp counted from the previous ClockTimeStamp in output order should or should not be interpreted as the time difference between the times of origin or capture of the associated frames or fields.

Seconds\_value/ Minutes\_value/ Hours\_value/ Time\_offset/ N\_frames, etc. are used to compute ClockTimeStamp.

## HRD Parameters in H.264

Most of important parameters used to parse SEI messages are in HRD parameters as shown in Figure 2-20.

```

Hrd_parameters( ){
    Cpb_cnt_minus1
    Bit_rate_scale
    Cpb_size_scale
    For(SchedSelIdx=0;    SchedSelIdx    <=
Cpb_cnt_minus1; SchedSelIdx ++){
        Bit_rate_value_minus1[SchedSelIdx]
        Cpb_size_value_minus1[SchedSelIdx]
        Cbr_flag[SchedSelIdx]
    }
    Initial_cpb_removal_delay_length_minus1
    Cpb_removal_delay_length_minus1
    Dpb_output_delay_length_minus1
    Time_offset_length
}

```

**Figure 2-20 HRD Parameters Syntax**

Cpb\_cnt\_minus1+1 specifies the number of alternative CPB specifications in the bitstream. When low\_delay\_hrd\_flag is equal to 1, cpb\_cnt\_minus1 shall be equal to 0. Bit\_rate\_scale and Bit\_rate\_value\_minus1[SchedSelIdx] specify the maximum input bit rate (the bit rate in bits per sec.) for the SchedSelIdx-th CPB with:

$$\text{BitRate}[\text{SchedSelIdx}] = (\text{bit\_rate\_value\_minus1}[\text{SchedSelIdx}] + 1) \times 2^{(6 + \text{bit\_rate\_scale})} \quad (2-15)$$

When the bit\_rate\_minus1[SchedSelIdx] is not present, BitRate[SchedSelIdx] shall be inferred to be equal to  $1000 \times \text{MaxBR}$  for VCL HRD parameters and to  $1200 \times \text{MaxBR}$  for NAL HRD parameters, respectively.

The CpbSize in bits is given by:

$$\text{CpbSize}[\text{SchedSelIdx}] = (\text{cpb\_size\_value\_minus1}[\text{SchedSelIdx}] + 1) \times 2^{(4 + \text{cpb\_size\_scale})} \quad (2-16)$$

When the `cpb_size_value_minus1` [`SchedSelIdx`] is not present, `CpbSize` [`SchedSelIdx`] shall be inferred to be equal to  $1000 \times \text{MaxCPB}$  for VCL HRD and to  $1200 \times \text{MaxCPB}$  for NAL HRD parameters, respectively.

### Derivation of DTS/ PTS in H.264

In MPEG-2 video, certain important information such as DTS/ PTS is not indicated in the ES. Utilizing ESs without explicit timing information might cause potential problems at random access instances in certain systems. Therefore, MPEG-2 PES becomes a kind of inevitable encapsulator for all applications. Once the time point of decoding and displaying the AUs in the PES is explicitly written at the Time Stamps in MPEG-2 Systems, there is no confusion for decoder and display processor to initiate actions.

Unfortunately, the same argument can be applied for H.264 ESs. Potential problems at random access instances can occur for H.264 ESs since there is no absolute timing information in the NAL units of H.264. To resolve this issue, H.264 can take advantage of already-well-established MPEG-2 Systems as an encapsulator as defined in the ITU-T H.222.0 | ISO/IEC 13818-1:2000/Amd.3 document. In such cases, an encoder's packetizer should be able to extract DTS/ PTS information from the NAL layer to generate the PES packet of MPEG-2 Systems with timing information. Once a PES encapsulated H.264 bitstream is generated, any decoder can process it with correct synchronization actions.

When H.264 bitstreams are not encapsulated in MPEG-2 Systems, DTS/ PTS information might be or might not be explicitly described in any header of the encapsulator. When the timing data is not explicitly written in forms of DTS/ PTS in such an encapsulator, the DTS/ PTS can be derived from timing related information in NAL layer of H.264.

Two key items of information to extract DTS/ PTS are `Buffering_period` (payloadSize) and `Pic_timing` (payloadSize) SEI messages. The AU with a buffering period SEI message that initializes the CPB is referred to as AU-0.

### DTS Derivation

The nominal removal time  $t_{r,nom}(0)$  and its DTS(0) of the AU-0 from the CPB are specified by:

$$\begin{aligned} t_{r,nom}(0) &= Initial\_cpb\_removal\_delay[SchedSelIdx]/90000 \text{ and} \\ DTS(0) &= Initial\_cpb\_removal\_delay[SchedSelIdx] \end{aligned} \quad (2-17)$$

Typically, the HRD is initialized at the beginning of a buffering period in the stream when a random access occurs. For the first AU of a buffering period that does not initialize the HRD, the nominal removal time  $t_{r,nom}(n)$  and its DTS(n) of the AU from the CPB are specified by:

$$\begin{aligned} t_{r,nom}(n) &= t_{r,nom}(n_b) + t_c \times cpb\_removal\_delay(n) \text{ and} \\ DTS(n) &= DTS(n_b) + \frac{num\_units\_in\_tick}{time\_scale} \times 90000 \times \\ &cpb\_removal\_delay(n) \end{aligned} \quad (2-18)$$

where  $t_{r,nom}(n_b)$  is the nominal removal time of the first picture of the previous buffering period and  $cpb\_removal\_delay(n)$  is specified in the picture timing SEI message associated with AU-n.

When an AU-n is the first AU of a buffering period,  $n_b$  and  $DTS(n_b)$  are set equal to n and DTS(n) at the removal time of AU-n.

The nominal removal time  $t_{r,nom}(n)$  and its DTS(n) of an AU-n that is not the first AU of a buffering period are given by:

$$\begin{aligned} t_{r,nom}(n) &= t_{r,nom}(n_b) + t_c \times cpb\_removal\_delay(n) \text{ and} \\ DTS(n) &= DTS(n_b) + \frac{num\_units\_in\_tick}{time\_scale} \times 90000 \times \\ &cpb\_removal\_delay(n) \end{aligned} \quad (2-19)$$

where  $t_{r,nom}(n_b)$  is the nominal removal time of the first picture of the current buffering period and  $cpb\_removal\_delay(n)$  is specified in the picture timing SEI message associated with AU-n.



Finally, the removal time and its DTS(n) of AU-n are specified as follows:

- If `low_delay_hrd_flag` is equal to 0 or nominal removal time  $t_{r,nom}(n) \geq$  final arrival time  $t_{af}(n)$ , the removal time  $t_r(n)$  and its DTS(n) of AU-n are specified by:

$$\begin{aligned}
 t_r(n) &= t_{r,nom}(n) \text{ and} \\
 DTS(n) &= DTS(n_b) + \frac{\text{num\_units\_in\_tick}}{\text{time\_scale}} \times 90000 \times \\
 &\quad \text{cpb\_removal\_delay}(n) \\
 &= DTS(0) + \frac{\text{num\_units\_in\_tick}}{\text{time\_scale}} \times 90000 \times \\
 &\quad (n_b \cdot \Delta_{cpb} + \text{cpb\_removal\_delay}(n)) \\
 &= \text{Initial\_cpb\_removal\_delay}[\text{SchedSelIdx}] + \\
 &\quad \frac{\text{num\_units\_in\_tick}}{\text{time\_scale}} \times 90000 \times \\
 &\quad (n_b \cdot \Delta_{cpb} + \text{cpb\_removal\_delay}(n)) \tag{2-20}
 \end{aligned}$$

Note that DTS(n) in Equation (2-20) shall be rounded to the closest integer value prior to its insertion in a PES header since it is a Time Stamp.

For example, when  $t_c = 1/29.97$  (sec) (i.e.,  $\Delta_{cpb} = 1$ ) with `num_units_in_tick`=1001 and `time_scale`=30000, DTS(n) is as follows:

$$\begin{aligned}
 DTS(n) &= DTS(n_b) + 3003 \times \text{cpb\_removal\_delay}(n) \\
 &= DTS(0) + 3003 \times n_b + 3003 \times \text{cpb\_removal\_delay}(n) \\
 &= \text{Initial\_cpb\_removal\_delay}[\text{SchedSelIdx}] + 3003 \times \\
 &\quad (n_b + \text{cpb\_removal\_delay}(n)) \tag{2-21}
 \end{aligned}$$

- Otherwise (`low_delay_hrd_flag` is equal to 1 and  $t_{r,nom}(n) < t_{af}(n)$ ), the removal time and its DTS(n) of AU-n are specified by:

$$\begin{aligned}
t_r(n) &= t_{r,nom}(n) + t_c \times Ceil((t_{af}(n) - t_{r,nom}(n)) \div t_c) \text{ and} \\
DTS(n) &= DTS(n_b) + \frac{num\_units\_in\_tick}{time\_scale} \times 90000 \times \\
&\quad (cpb\_removal\_delay(n) + Ceil((t_{af}(n) - t_{r,nom}(n)) \div t_c)) \\
&= DTS(0) + \frac{num\_units\_in\_tick}{time\_scale} \times 90000 \times \\
&\quad (n_b \cdot \Delta_{cpb} + cpb\_removal\_delay(n) + Ceil((t_{af}(n) - t_{r,nom}(n)) \div t_c)) \\
&= Initial\_cpb\_removal\_delay[SchedSelIdx] + \\
&\quad \frac{num\_units\_in\_tick}{time\_scale} \times 90000 \times (n_b \cdot \Delta_{cpb} + cpb\_removal\_delay(n) + \\
&\quad Ceil((t_{af}(n) - t_{r,nom}(n)) \div t_c))
\end{aligned} \tag{2-22}$$

For example, when  $t_c = 1/29.97$  (sec) with  $num\_units\_in\_tick=1001$  and  $time\_scale = 30000$ ,  $DTS(n)$  is as follows:

$$\begin{aligned}
DTS(n) &= DTS(n_b) + 3003 \times (cpb\_removal\_delay(n) + \\
&\quad Ceil((t_{af}(n) - t_{r,nom}(n)) \div t_c)) \\
&= DTS(0) + 3003 \times (n_b + cpb\_removal\_delay(n) + \\
&\quad Ceil((t_{af}(n) - t_{r,nom}(n)) \div t_c)) \\
&= Initial\_cpb\_removal\_delay[SchedSelIdx] + 3003 \times \\
&\quad (n_b + cpb\_removal\_delay(n) + Ceil((t_{af}(n) - t_{r,nom}(n)) \div t_c))
\end{aligned} \tag{2-23}$$

This case indicates that the size of AU-n is so large that it prevents removal at the nominal removal time.

## PTS Derivation

Picture n is decoded and its DPB output time  $t_{o,dpb}(n)$  and PTS(n) are derived by:

$$t_{o,dpb}(n) = t_r(n) + t_c \times dpb\_output\_delay(n) \text{ and}$$

$$PTS(n) = DTS(n) + \frac{num\_units\_in\_tick}{time\_scale} \times 90000 \times dpb\_output\_delay(n) \quad (2-24)$$

The output time of the current picture and its PTS(n) are specified as follows:

- If  $t_{o,dpb}(n) = t_r(n)$ , the current picture is output.

$$t_r(n) = t_{r,nom}(n) \text{ and}$$

$$PTS(n) = DTS(n) + \frac{num\_units\_in\_tick}{time\_scale} \times 90000 \times dpb\_output\_delay(n) \quad (2-25)$$

Note that PTS(n) in Equation (2-25) shall be rounded to the closest integer value prior to its insertion in a PES header since it is a Time Stamp.

For example, when  $t_c = 1/29.97$  (sec) with num\_units\_in\_tick=1001 and time\_scale = 30000, PTS(n) is as follows:

$$PTS(n) = DTS(n) + 3003 \times dpb\_output\_delay(n) \quad (2-26)$$

- Otherwise ( $t_{o,dpb}(n) > t_r(n)$ ), the current picture is output later and will be stored in the DPB. And, the stored picture is output at time  $t_{o,dpb}(n)$  unless indicated not to be output by the decoding or inference of no\_output\_of\_prior\_pics\_flag equal to 1 at a time that precedes  $t_{o,dpb}(n)$ .

### Artificial Generation of PTS For Special Pic\_struct Type

If a picture has a special structure such as frame doubling, the same content of decoded frame would be used again at a later time. If necessary, the PTSs can be artificially generated. ClockTimeStamp formula in a

previous subsection may be used to compute multiple PTS for a specific `Pic_struct`.

### **Constraints of Byte-Stream NAL Unit Format for MPEG-2 Systems**

An H.264 stream is an element of an ISO/IEC 13818-1 program as defined by the PMT in a TS and the PSM in a PS. The `stream_id` and `stream_type` are defined as shown in Table 2-11 and Table 2-12 for H.264. H.264 is also called “AVC video stream” in this context. The carriage and buffer management of AVC video streams is defined using existing parameters from international standards such as PTS and DTS as well as information present within a H.264 video stream.

Carriage of H.264 streams in MPEG-2 Systems defines accurate mapping between STD parameters and HRD parameters that may be present in an AVC video stream. When an H.264 stream is carried in MPEG-2 Systems, coded H.264 bitstreams shall be contained in PES packets. The coded data shall comply with the byte-stream NAL unit format as shown in Figure 2-15.

Extra constraints on byte-stream NAL unit format are as follows:

- Each AVC AU shall contain an AU delimiter NAL Unit.
- Each byte-stream NAL unit that carries the AU delimiter shall contain exactly one zero-byte syntax element.
- All SPS and PPS necessary for decoding the AVC video stream shall be present within the AVC video stream.
- Each AVC video sequence that contains `hrd_parameters()` with the `low_delay_hrd_flag` set to “1” shall carry VUI parameters where the `timing_info_present_flag` shall be set to “1.”

### **Encapsulation of H.264 in MPEG-2 Systems**

H.264 video is carried in PES packets in the payload using one of 16 `stream_id` values assigned to video as shown in Table 2-11, while

signaling an H.264 video stream by means of the assigned stream\_type value in the PMT or PSM as shown in Table 2-12.

**Table 2-11 Stream\_id Assignment (MPEG IS-Amd3: Table 2-18)**

Stream_id	Stream coding
1011 1100	Program_stream_map
110x xxxx	ISO/IEC 13818-3 or ISO/IEC 11172-3 or ISO/IEC 13818-7 or ISO/IEC 14496-3 audio stream number x xxxx
1110 xxxx	ITU-T Rec. H.262  ISO/IEC 13818-2, ISO/IEC 11172-2, ISO/IEC 14496-2 or ITU-T Rec. H.264   ISO/IEC 14496-10 video stream number xxxx
.....	.....
1111 1111	Program_stream_directory

**Table 2-12 Stream\_type Assignment (MPEG IS-Amd3: Table 2-29)**

Value	Description
0x00	ITU-T ISO/IEC Reserved
0x01	ISO/IEC 11172-2 Video
0x02	ITU-T Rec. H.262  ISO/IEC 13818-2 or ISO/IEC 11172-2 video stream
0x03	ISO/IEC 11172-3 Audio
.....	.....
0x1b	AVC video stream as defined in ITU-T Rec. H.264  ISO/IEC 14496-10 video
0x1C~0x7e	ITU-T Rec. H.222.0  ISO/IEC 13818-1 Reserved
0x7f	IPMP stream
0x80~0ff	User Private

**Table 2-13 PD and PED Examples (MPEG IS-Amd3: Table 2-39)**

Descriptor_tag	TS	PS	Identification
.....			.....
2	x	x	Video_stream_descriptor
3	x	x	Audio_stream_descriptor
9	x	x	CA_descriptor
10	x	x	ISO_639_language_descriptor
35	x		MultiplexeBuffer_descriptor
40	x	x	AVC video descriptor
42	x	x	AVC timing and HRD descriptor
.....			.....

The highest level that may occur in an H.264 video stream as well as a profile that the entire stream conforms to should be signaled using the AVC video descriptor as shown in Table 2-13. If an AVC video descriptor is associated with an H.264 video stream, then this descriptor shall be conveyed in the descriptor loop for the respective elementary stream entry in the PMT or PSM.

The AVC video descriptor provides basic information for identifying coding parameters of the associated H.264 stream, such as on profile and level parameters included in the SPS of a H.264 stream. The AVC video descriptor also signals the presence of AVC still pictures and the presence of AVC 24-hour pictures in the H.264 video stream. If the descriptor is not included in the PMT or PSM for a H.264 video stream, such a H.264 video stream shall not contain AVC still pictures or AVC 24-hour pictures. The syntax for AVC video descriptor is provided in Figure 2-21.

```

AVC_video_descriptor( ){
    descriptor_tag          (8 bit)          // uimsbf
    descriptor_length       (8 bit)          // uimsbf
    profile_idc             (8 bit)          // uimsbf
    constraint_set0_flag    (1 bit)          // bslbf
    constraint_set1_flag    (1 bit)          // bslbf
    constraint_set2_flag    (1 bit)          // bslbf
    AVC_compatible_flags    (5 bit)          // bslbf
    level_idc               (8 bit)          // uimsbf
    AVC_still_present       (1 bit)          // bslbf
    AVC_24_hour_picture_flag(1 bit)          // bslbf
    reserved                (6 bit)          // bslbf
}

```

**Figure 2-21 Syntax for AVC Video Descriptor**

Most of the semantics are exactly same to the those in SPS. Here, `AVC_still_present` indicates that the AVC video stream may include AVC still pictures, while `AVC_24_hour_picture_flag` indicates that the associated AVC video stream may contain AVC 24-hour pictures. Note that the definition of AVC still picture is to be an AVC still picture that consists of an AVC AU containing an IDR picture, preceded by SPS and PPS NAL units that carry sufficient information to correctly decode the IDR picture. Preceding an AVC still picture, there shall be another AVC still picture or an End of Sequence NAL unit terminating a preceding coded video sequence. An AVC still picture is repeatedly displayed until the PTS of the next AU. And, the definition of AVC 24-hour picture is an AVC AU with a presentation time that is more than 24 hours in the future. The AVC AU- $n$  has a presentation time that is more than 24 hours in the future if the difference between the initial arrival time  $t_{ai}(n)$  and the DPB output time  $t_{o,dpb}(n)$  is more than 24 hours.

The AVC timing and HRD descriptor provides timing and HRD parameters of the associated H.264 video stream. For each AVC video stream carried in MPEG-2 Systems, the AVC timing and HRD descriptor shall be included in the PMT or PSM. The H.264 bitstream can carry VUI parameters with the `timing_info_present_flag` set to “1”

- for each IDR picture.
- and for each picture that is associated with a recovery point SEI message.

Absence of the AVC timing and HRD descriptor in the PMT for a H.264 video stream signals usage of the leak method in the T-STD, but such usage can also be signaled by the `hrd_management_valid_flag` set to “0” in the AVC timing and HRD descriptor. If this transfer rate is used in the T-STD for the transfer between  $MB_n$  to  $EB_n$ , the AVC timing and HRD descriptor with the `hrd_management_valid_flag` set to “1” shall be included in the PMT for the H.264 video stream.

```

AVC_timing_and_HRD_descriptor( ){
    descriptor_tag          (8 bit)          // uimsbf
    descriptor_length       (8 bit)          // uimsbf
    hrd_management_valid_flag(1 bit)         // bslbf
    reserved                (6 bit)         // bslbf
    picture_and_timing_info_present (1 bit)  // bslbf
    if( picture_and_timing_info_present) {
        90kHz_flag         (1 bit)         // bslbf
        reserved           (7 bit)         // bslbf
        if( 90kHz_flag=='0'){
            N                (32 bit)       // uimsbf
            K                (32 bit)       // uimsbf
        }
        num_units_in_tick   (32 bit)       // uimsbf
    }
    fixed_frame_rate_flag   (1 bit)         // bslbf
    temporal_poc_flag       (1 bit)         // bslbf
    picture_to_display_convesion_flag(1 bit) // bslbf
    reserved                (5 bit)         // bslbf
}

```

**Figure 2-22 Syntax for AVC Timing and HRD Descriptor**

When the AVC timing and HRD descriptor is associated to a H.264 video stream carried in a TS, the following applies: If the `hrd_management_valid_flag` is set to “1,” Buffering Period SEI and Picture Timing SEI messages shall be present in the associated H.264 video stream. These Buffering Period SEI messages shall carry coded `initial_cpb_removal_delay` and `initial_cpb_removal_delay_offset` values for the NAL HRD. If the `hrd_management_valid_flag` is set to “1,” the transfer of each byte from  $MB_n$  to  $EB_n$  in the T-STD shall be according to the delivery schedule for that byte into the CPB in the NAL HRD.



When the `hrd_management_valid_flag` is set to “0,” the leak method shall be used for the transfer from  $MB_n$  to  $EB_n$  in the T-STD.

The `90kHz_flag`, when set to “1,” indicates that the frequency of the AVC time base is 90 kHz. For a H.264 video stream the frequency of the AVC time base is defined by the AVC parameter `time_scale` in VUI parameters. The relationship between the AVC `time_scale` and the STC shall be defined by the parameters N and K in this descriptor as follows:

$$time\_scale = \frac{(N \times system\_clock\_frequency)}{K} \quad (2-27)$$

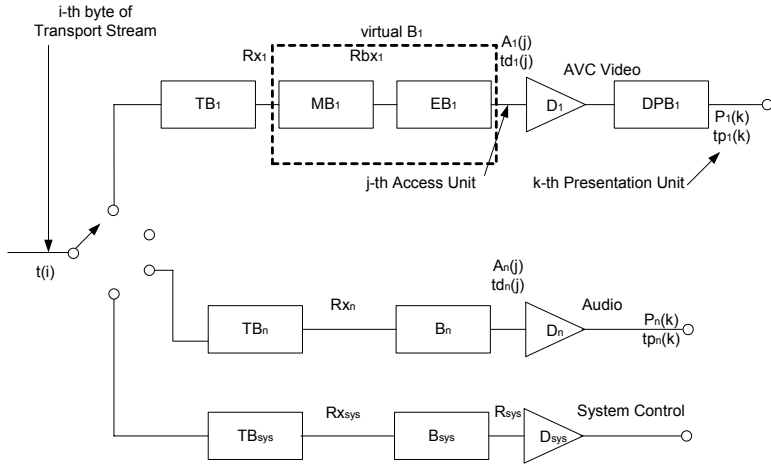
where `time_scale` denotes the exact frequency of the AVC time base with K larger than or equal to N.

When the `temporal_poc_flag` is set to “1” and the `fixed_frame_rate_flag` is set to “1,” the associated H.264 video stream shall carry Picture Order Count (POC) information (`PicOrderCnt`). When the `temporal_poc_flag` is set to “0,” no information is conveyed regarding any potential relationship between the POC information in the H.264 video stream and time.

For PES packetization, no specific data alignment constraints apply. For synchronization and STD management, PTSs and DTSs are encoded in the header of the PES packet that carries the H.264 video ESs.

### Extended T-STD

The T-STD model is the same as that of MPEG-2 T-STD except extended DPB as depicted in Figure 2-23. Carriage of a H.264 bitstream over MPEG-2 Systems does not impact the size of buffer  $DPB_n$ . The size of  $DPB_n$  is actually defined in H.264 standard for decoding a H.264 bitstream in the STD. A decoded H.264 AU enters  $DPB_n$  instantaneously upon decoding of the H.264 AU, hence at the CPB removal time of the H.264 AU. A decoded H.264 AU is presented at the DPB output time.



**Figure 2-23 T-STD Model Extension for H.264**

If the H.264 video stream provides insufficient information to determine the CPB removal time and the DPB output time of H.264 AU, then these time instances shall be determined in the STD model from PTS and DTS as follows:

- The CPB removal time of H.264 AU- $n$  is the instant in time indicated by  $DTS(n)$  where  $DTS(n)$  is the DTS value of the AU- $n$ .
- The DPB output time of H.264 AU- $n$  is the instant in time indicated by  $PTS(n)$  where  $PTS(n)$  is the PTS value of the AU- $n$ .

The output rate  $R_x$  is defined for T-STD based on data types as follows:

$$R_{x_n} = \text{bit\_rate for video data,} \quad (2-28)$$

where  $\text{bit\_rate}$  is the bit rate  $\text{BitRate}[\text{cpb\_cnt\_minus1}]$  of data flow into the CPB for the byte-stream format signaled in the NAL  $\text{hrd\_parameters}()$  carried in VUI parameters in the H.264 bitstream. If

NAL `hrd_parameters()` are not present, the `bit_rate` shall be the bitrate  $1200 \times \text{maxBR}[\text{level}]$  defined in Annex A of H.264 standard.

The purpose of “B” or “virtual B” (video case) is to eliminate packet multiplexer jitter. However, virtual B is broken down into two units of buffers for video – MB and EB. Resource usage schedule should be synchronized between an encoder and a decoder in terms of HRD for video. Since EB is nothing but a CPB in the HRD buffer, the size of  $EB_n$  (a.k.a.,  $EBS_n$ ) is  $cpb\_size$ . In other words,

$$\blacksquare \quad EBS_n = cpb\_size. \quad (2-29)$$

If NAL `hrd_parameters()` are not present, then the  $cpb\_size$  shall be the size  $1200 \times \text{maxCPB}$  defined in Annex A of H.264 standard.

The size  $MBS_n$  of Buffer  $MB_n$  is defined as follows:

$$\blacksquare \quad MBS_n = BS_{mux} + BS_{oh} + 1200 \times \text{MaxCPB}[\text{level}] - cpb\_size \quad (2-30)$$

where PES packet overhead buffering  $BS_{oh} = (1/750)\text{sec} \times \max\{1200 \times \text{MaxBR}[\text{level}], 2000000\text{bits/sec}\}$  (2-31)

and additional multiplex buffering  $BS_{mux} = 0.004\text{sec} \times \max\{1200 \times \text{MaxBR}[\text{level}], 2000000\text{bits/sec}\}$ , (2-32)

where  $\text{MaxCPB}[\text{level}]$  and  $\text{MaxBR}[\text{level}]$  are defined for the byte-stream format in H.264 standard.

If the `AVC_timing_and_HRD_descriptor` is present with the `hrd_management_valid_flag` set to “1,” the transfer of data from  $MB_n$  to buffer  $EB_n$  shall follow the HRD defined scheme for data arrival in the CPB as defined in Annex C in H.264 standard. Otherwise, the leak method shall be used to transfer data from  $MB_n$  to  $EB_n$  as follows:

$$\blacksquare \quad Rbx_n = 1200 \times \max \text{BR}[\text{level}]. \quad (2-33)$$

### Extended P-STD

The P-STD model is the same as that of MPEG-2 P-STD except with extended DPB as depicted in Figure 2-24. Carriage of a H.264 bitstream over MPEG-2 Systems does not impact the size of buffer  $DPB_n$ . For each H.264 video stream  $n$ , the size  $BS_n$  of buffer  $B_n$  in the P-STD is defined by the P-STD\_buffer\_size field in the PES packet header. Buffer  $DPB_n$  shall be managed in exactly the same way as in the extended T-STD aforementioned. The H.264 AU enters buffer  $B_n$ . At the time  $td_n(j)$ , H.264 AU  $A_n(j)$  is decoded and instantaneously removed from  $B_n$ . The decoding time  $td_n(j)$  is specified by the DTS or by the CPB removal time, derived from information in the H.264 video stream.

PS shall be constructed so that the following conditions for buffer management are satisfied:

- $B_n$  shall not overflow.
- $B_n$  shall not underflow, except when VUI parameters are present for the H.264 video sequence with the low\_delay\_hrd\_flag set to “1” or when trick\_mode status is true. Underflow of  $B_n$  occurs for H.264 AU  $A_n(j)$  when one or more bytes of  $A_n(j)$  are not present in  $B_n$  at the decoding time  $td_n(j)$ .

### DTS/ PTS Carriage in PES Packets for AVC Pictures

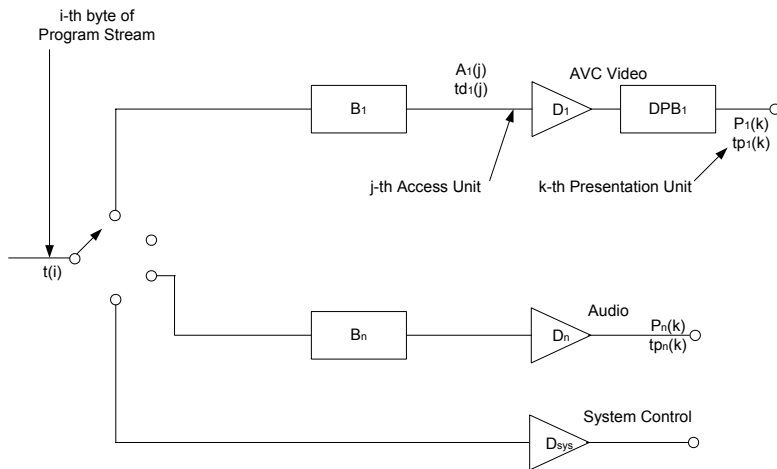
ITU-T H.222.0 | ISO/IEC 13818-1:2000/Amd.3 describes the extension of MPEG-2 Systems to encapsulate H.264.

When H.264 video bitstreams are encapsulated in MPEG-2 Systems, DTS/ PTS information are explicitly described in PES headers. However, there is no explicit DTS/ PTS value in the PES header for AVC 24-hour pictures or AVC still pictures. For such H.264 AU, decoders shall infer the PTS through HRD parameters in H.264 video streams, as mentioned in previous subsections. Therefore, each H.264 video stream that contains one or more AVC 24-hour pictures

- shall either carry picture timing SEI messages with coded values of `Cpb_removal_delay` and `Dpb_output_delay`.
- or shall carry VUI parameters with the `fixed_frame_rate_flag` set to “1” and shall carry POC information.

If a DTS is present in the PES packet header, it shall refer to the first H.264 AU that commences in this PES packet. An H.264 AU commences in a PES packet if the first byte of the H.264 AU is present in the PES packet.

If a PTS is present in the PES packet header, it shall refer to the first H.264 AU that commences in this PES packet. An H.264 AU commences in a PES packet if the first byte of the H.264 AU is present in the PES packet.



**Figure 2-24 P-STD Model Extension for H.264**

## 2.5 Comparisons between VC-1 and H.264

VC-1 is an emerging video standard primarily developed by Microsoft, while H.264 is an emerging video standard developed by the MPEG community under the auspice of the ISO [srinivasan:WMV9]. The VC-1 standard has been adopted for HD-DVD and Bluray-DVD. Its strength is its computational efficiency – currently low powered Pentium can decode HD video without any SOC support. If power consumption were not a concern, dedicated SOC might not be needed for VC-1 decoding. In fact, VC-1 (i.e., just another name of WMV-9) decoders are widely used as they are part of the Windows Media Player. A key weakness, however, is access to VC-1 encoder technology. There is limited activity on open VC-1 encoder development since no public reference implementation is available for encoding algorithms.

On the other hand, H.264 received more industry attention and support. The H.264 standard has also been adopted for HD-DVD and Bluray-DVD. Compared with the VC-1, H.264 technology has been completely open through the development process of the standard. Therefore, key players in the CE industry have contributed to the development and have taken the lead in manufacturing and promoting H.264. Especially, broadcasting industry sees more opportunities in H.264. H.264 is believed to perform better than the VC-1 for profiles above High Profile in terms of compression ratio.

### Tool Comparison and Complexity

Table 2-14 shows tools/features comparison between VC-1 and H.264. The details are covered in Chapters 4 - 9. There are many more Intra Prediction options devised in H.264 compared with VC-1. For MV accuracy, similar interpolation and resolutions are used. There are many more MC options devised in H.264 compared with VC-1. However, there are many more Transform options devised in H.264 compared with VC-1. In summary, the focus of the VC-1 tools is on simplicity with emphasis on compression ratio, while the focus of the H.264 tools is on rate-distortion performance with many viable options in terms of compression ratio.

**Table 2-14 Tools/ Features Comparison**

Feature	WMV-9/ VC-1 Main	VC-1 Advanced	H.264 Main	H.264 High
Picture type	I, P, B, BI, Skipped P	I, P, B, BI, Skipped P	N/A	N/A
Intra prediction	3 DC/ AC modes in frequency domain	3 DC/ AC modes in frequency domain	9(4x4) + 4(16x16) in pixel domain	9(4x4) + 9(8x8) + 4(16x16) in pixel domain
MV accuracy	¼ pel (4 tap bi-cubic or bi-linear)	¼ pel (4 tap bi-cubic or bi-linear)	¼ pel (6+2 tap)	¼ pel (6+2 tap)
MC block size	16x16, 8x8	16x16, 8x8	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4	16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4
Num reference frames	2	2	16 (limited by DPB <sub>max</sub> )	16 (limited by DPB <sub>max</sub> )
Trasform (I– Integer TX )	8x8 (I), 8x4(I), 4x8(I), 4x4(I)	8x8 (I), 8x4(I), 4x8(I), 4x4(I)	4x4(I)	4x4(I), 8x8(I)
Transform scaling matrices	N/A	N/A	N/A	8 adjustable (per Seq or Pic)
Entropy coding	VLC, kind of CA-VLC	VLC, kind of CA- VLC	CA-VLC, CA-BAC	CA-VLC, CA- BAC
Interlace	N/A	MPEG-2 like	PAFF, MBAFF	PAFF, MBAFF
In-loop deblocking filter (# taps, pels modified, pel compared)	6, 2, 8  (non-linear filtering)	6, 2, 8  (non-linear filtering)	5 (max), 6, 8  (linear filtering)	5 (max), 6, 8  (linear filtering)

VC-1 is more complex to decode than MPEG-2 and H.264 is more complex to decode than VC-1. The comparison based on computation measure is shown in Table 2-15. Even VC-1 Main Profile (MP) decoding takes much lower computation than that of H.264 Baseline Profile (BP) as shown in Table 2-15. This implies that low power/ low cost implementation with minimal efforts is possible with VC-1.

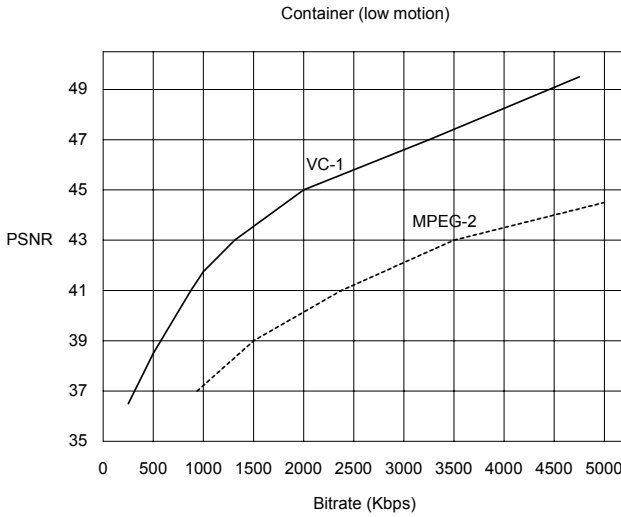
**Table 2-15 Complexity Comparison**

Sequence	Millions of ARM cycles/ second	
	VC-1 Main	H.264/ AVC Baseline (Optimized code)
Foreman	27	38
News	17	22
Container	19	24
Slient	18	25
Glasgow	25	30
Average	21.2	27.8

**Objective Tests**

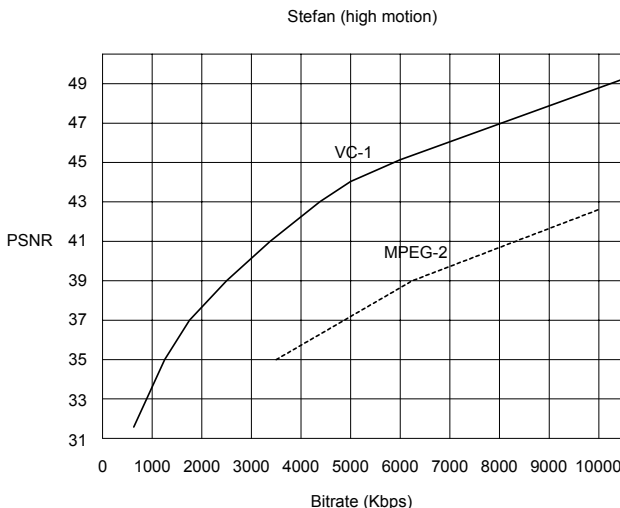
In VC-1, low motion video and high motion video distinctively use different Huffman tables based on Qp. Therefore, tests are divided into two categories to compare VC-1 MP and MPEG-2. Figure 2-25 shows general characteristics of PSNR and bitrate for slow moving video contents. In contrast, Figure 2-26 shows general characteristics of PSNR and bitrate for fast moving video contents. Given a bitrate, the quality of VC-1 coded bitstreams is significantly better than that of MPEG-2 coded bitstreams. The performance of VC-1 for slow moving scenes is actually much better than that for fast moving scenes.





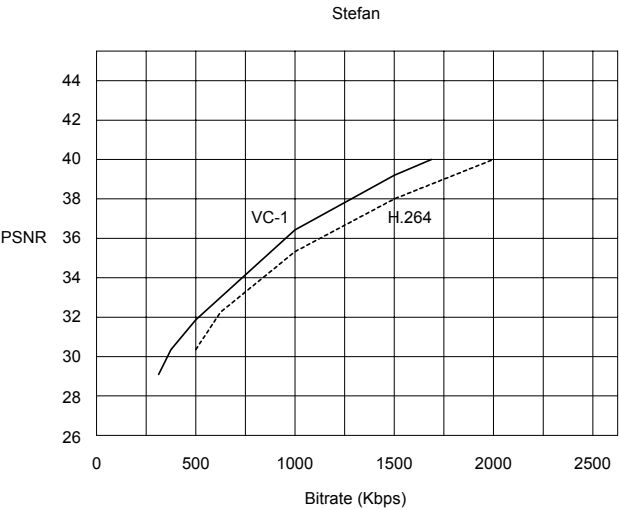
**Figure 2-25 MPEG-2 vs. VC-1 MP (WMV-9) for Low Motion Sequence**

For example, for slow moving scenes as shown in Figure 2-25, the quality of MPEG-2 coded at 2500 Kbps is same as the quality of VC-1 video coded at about 800 Kbps. In other words, the performance of VC-1 is 3 times better than that of MPEG-2 in terms of compression performance. For fast moving scenes as shown in Figure 2-26, the quality of MPEG-2 video at 6300 Kbps is same as the quality of VC-1 video at 2500 Kbps. In other words, the performance of VC-1 is 2.5 times better than that of MPEG-2 in terms of compression performance. Even though compression performance varies based on characteristics of input video and profiles/tools, it is safe to say that the performance of VC-1 is more than 2 times better than that of MPEG-2.



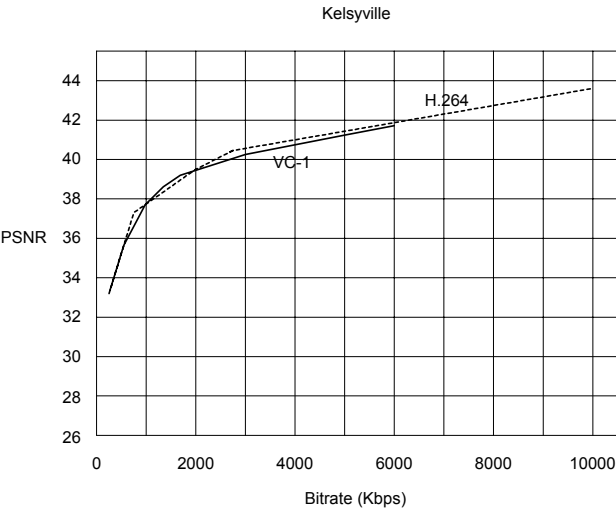
**Figure 2-26 MPEG-2 vs. VC-1 MP (WMV-9) for High Motion Sequence**

In VC-1 and H.264, inter prediction options are significantly different. Apart from various coding modes in MC block size and interpolation options, H.264 can utilize many more reference pictures compared to a maximum of two allowed by VC-1. To compare coding modes fairly, the same number of reference pictures should be at least allocated for both standards. Figure 2-27 illustrates results of VC-1 MP and H.264 BP. Since B slices are not allowed in H.264 BP, only one reference picture (with P slices) is used to compare coding options in Figure 2-27. For VC-1 MP, only one reference picture is used with the picture type order of I, P, P, P, ... The implication of the result shown in Figure 2-27 is that the performance of H.264 is not significantly better than that of VC-1 without multi-picture MCP option. In fact, the performance of VC-1 shows better than that of H.264 without multi-picture MCP option as shown in Figure 2-27. For example, the quality at 1500 Kbps with H.264 is achieved at about 1300 Kbps with VC-1. In other words, the performance of VC-1 is 1.2 times better than that of H.264 in terms of compression performance without multi-picture MCP option on. The compression performance varies based on characteristics of input video and profiles/tools, Figure 2-27 shows only one case of results with a specific input video.



**Figure 2-27 VC-1 MP (WMV-9) vs. H.264 BP with 1 Reference Picture**

Figure 2-28 illustrates results of VC-1 AP and H.264 High Profile (HP). Compared with Figure 2-27, the results of Figure 2-28 are obtained with multi-picture MCP tool on. The performance of H.264 is better than that of VC-1 as shown in Figure 2-28. However, the performance of H.264 is not significantly better than that of VC-1 even with multi-picture MCP option. For example, the quality at 6000 Kbps with VC-1 is achieved at about 5000 Kbps with H.264. In other words, the performance of H.264 is 1.2 times better than that of VC-1 in terms of compression performance with all reasonable options turned on. The compression performance varies based on characteristics of input video and profiles/tools, Figure 2-28 shows only one case of results with a specific input video.



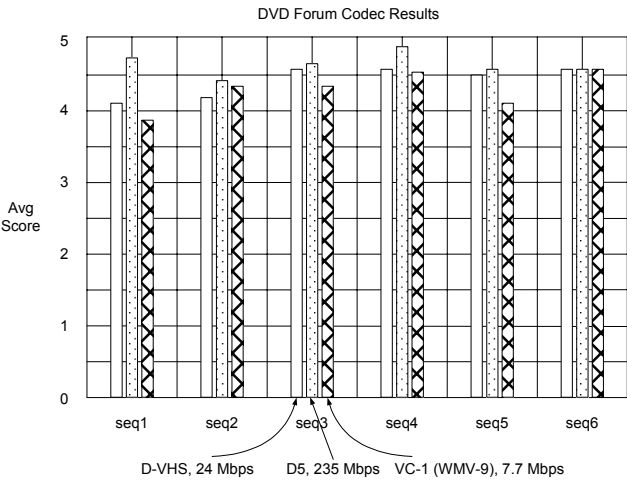
**Figure 2-28 VC-1 Advanced vs. H.264 High**

PSNR is generally taken as a reasonably reliable measurement for distortion. Based on PSNR, presented results imply that the performance of VC-1 is in the similar rate distortion ballpark as that of H.264, if not the same. The results are shown similarly in the MP and the HP of H.264 compared to the MP and the AP of VC-1.

**Subjective Tests**

Objective measures provide just one kind of measurement. In many cases, subjective measures provide more reliable results than objective measures since objective measures do not properly represent the characteristics of human visual systems.

Figure 2-29 shows results of the DVD Forum Codec comparison tests. The Forum tested the performance of multiple video codecs (e.g., MPEG-2, MPEG-4 ASP, H.264, VC-1 (WMV-9), etc.) in six film clips of time length 90s and resolution 1920x1080. Industry reference D-VHS (MPEG-2 at 24 Mbps) and original D5 master (nearly lossless compression at 235 Mbps) are included to compare with VC-1 as shown in the Figure 2-29. The subjective test scores for VC-1 with D-VHS and D5 are presented in Figure 2-29. VC-1 shows strong subjective test results as presented in Figure 2-29.

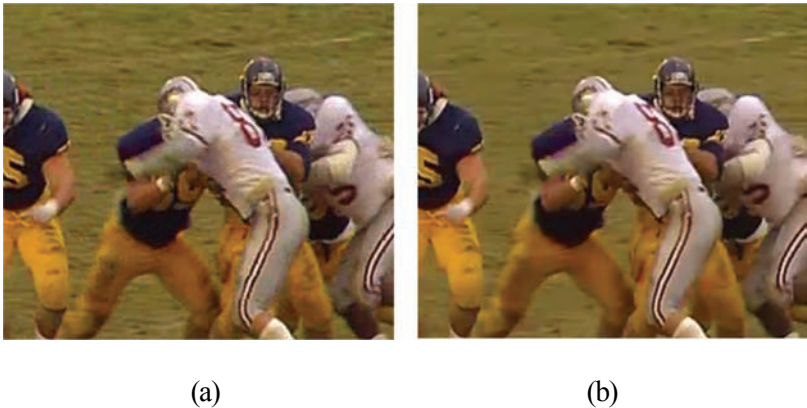


**Figure 2-29 DVD Forum Codec Results**

Figure 2-30 and 2-31 illustrate results of subjective comparison tests for VC-1 and H.264. Generally, they look equally good to bare human eyes with similar target bitrates when they are set at a higher operational bitrate in the allowed range in a specific profile/level combination. On the other hand, H.264 looks better with similar target bitrates when they are set at a medium or lower operational bitrate in the allowed range in a specific profile/level. Microsoft claims that generally VC-1 and H.264 show measures in the similar performance ballpark, but VC-1 has the advantage of being simpler in terms of implementation. For example, H.264 multi-picture MCP tool imposes harsh conditions for memory accesses on SOC implementation, while the VC-1 memory accesses patterns are similar to that of MPEG-2. Both VC-1 and MPEG-2 have a maximum of two reference pictures.



**Figure 2-30 Comparison of 704x576 video coded at 1.4 Mbps, the lower end of the bitrate for profile/level (a) VC-1 (b) H.264**



**Figure 2-31 Comparison of 352x28 video coded at 917 Kbps, the higher end of the bitrate for profile/level (a) VC-1 (b) H.264**

The VC-1 and H.264 Video Compression Standards for  
Broadband Video Services

Lee, J.-B.; Kalva, H.

2008, XVII, 496 p., Hardcover

ISBN: 978-0-387-71042-6