MPEG-4, AVS deliver better video compression

By Jose Lau
VP of Asia
SigmaTel Corp.
E-mail: joselau.asia @sigmatel.com

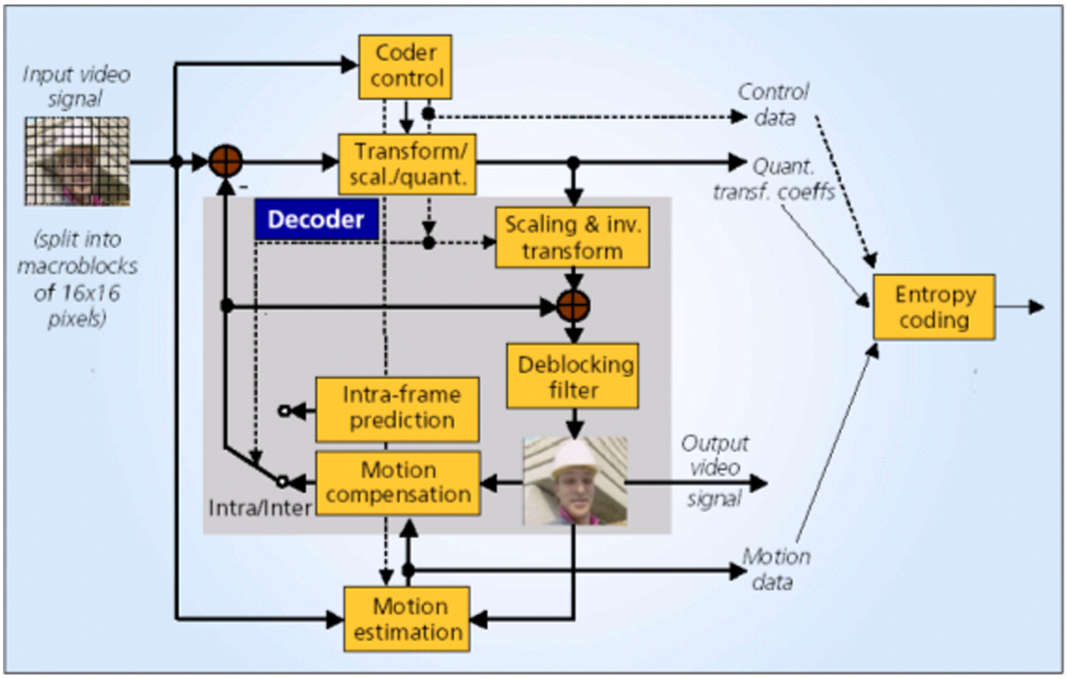

Figure 2: The video coding layer of H.264/AVC consists of a hybrid of temporal and spatial prediction

MPEG-4 is an ISO/IEC standard developed by the Moving Picture Expert Group. A successor to MPEG-1 and MPEG-2, it is now a standard under ISO/IEC-14496 vers 1 and 2. MPEG-4 aims to provide better compression, a more flexible format and more options to target on applications such as set-top boxes, Internet and mobile devices.

MPEG-4 standard is made up of three parts: system (part 1), visual (part 2) and audio (part 3). Delivery Multimedia Integration Framework (part 6) defines an interface between application and network/storage. Conformance (part 4) defines how to test an MPEG-4 implementation while part 5 gives a significant body of reference software that can be used to start implementing the standard. Part 7 defines an optimized video encoder modified from the reference software (part 5).

The new parts that have been added into MPEG-4 are:

- Part 8: Transport–defines how to map MPEG-4 streams onto IP transport.
- Part 9: Reference Hardware Description: Phase 1–hardware accelerators; Phase 2–optimized reference software integration through virtual socket
- Part 10: Advanced Video Coding
- Part 11: Scene Description
- Part 12: ISO Media File Format.
- Part 13 : IPMP Extensions.
- Part 14 : MP4 File Format
- Part 15 : AVC File Format
- Part 16 : AFX (Animation Framework eXtensions) and MuW (Multi-user Worlds).

While MPEG-4 part 10 advanced video coding/H.264 will not replace MPEG-4 part 2, the new standard involves existing techniques such as intra-prediction, integer transform, variable block size motion estimation/compensation and de-blocking filter.

It promises 50 percent more average bit rate reduction than existing standards of the same visual quality.

MPEG-4 deals with "media objects," a general term for visual and audio content. These media objects are used together to form the A/V scenes. As a toolbox capable of providing interactive and personalized media content, MPEG-4 compresses/decompresses other media objects such as text, graphics, speech, animation, 2D and 3D objects in addition to audiovisual data.

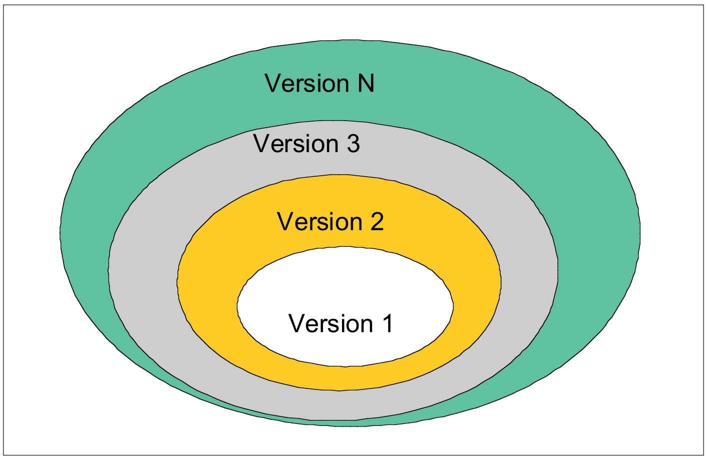

Figure 1: Ver 2 is a backward-compatible extension of ver 1, and ver 3 is a backward-compatible extension of ver 2, and so on.

To support the diversity of the future content market, MPEG-4 offers a variety of "profiles"–tool sets from the toolbox–that are useful for specific applications (e.g. in A/V coding, simple or advanced simple profile). Users need only implement the profiles that support the functionality required.

MPEG-4 versions

MPEG-4 ver 1 was approved by MPEG in December 1998 while ver 2 was approved in early 2000. After these two major vers, more tools were added in subsequent amendments that could be qualified as newer vers. However, it is more important to distinguish profiles. Existing tools and profiles from any ver are never replaced in subsequent vers; technology is always added to MPEG-4 in the form of new profiles. **Figure 1** depicts the relationship between the vers. Ver 2 is a backward-compatible extension of ver 1, and ver 3 is a backward-compatible extension of ver 2, and so on.

MPEG-4 profiles

MPEG-4 provides a large set of tools for the coding of A/V objects. To implement the standard effectively, subsets of the MPEG-4 system, Visual and Audio tool sets, which can be used for specific applications, have been identified. These subsets, called "profiles", limit the num-

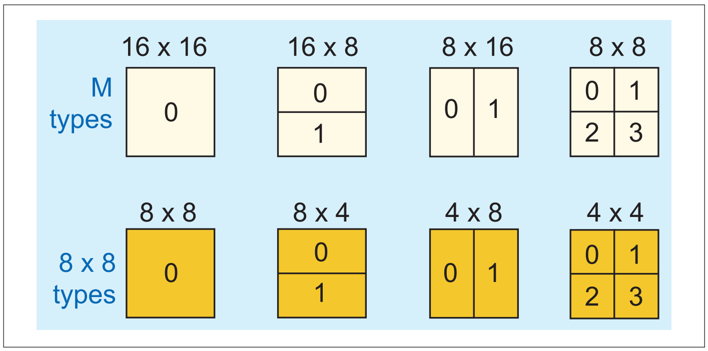

Figure 3: Figure shows the partitioning of a macroblock in H.264 P slices.

ber of tool sets that a decoder has to implement. For each profile, one or more levels have been set, minimizing the computational complexity. The approach is similar to MPEG-2, where the most well known profile/level combination is "Main Profile@ Main Level." A Profile@Level combination allows a codec builder to implement only the subset of the standard he needs, while interworking with other MPEG-4 devices built using the same combination, and check whether MPEG-4 devices comply with the standard (conformance testing).

Profiles exist for various types of media content (audio, visual and graphics) and for scene descriptions. MPEG does not prescribe or advise combinations of these profiles, but care has been taken that good matches exist between the different application areas.

The most commonly used MPEG profile is simple profile (SP), which was defined in ver 1 of MPEG-4. SP is defined for very low complexity decoders that provide efficient, error-resilient coding of rectangular video objects.

Advanced simple profile (ASP), on the other hand, was defined to provide significantly better compression efficiency than SP (as much as 50 percent improvement) as specified in ver 2 of MPEG-4. Several advanced compensation tools such as quarter-pixel motion estimation and compensation (QME/QMC), global motion estimation and compensation (GME/GMC), and B-frame are included in MPEG-4 ASP.

The simple profile only accepts objects of type Simple, and was created with low-complexity applications in mind. The first is mobile use of A/V services, and the second is putting very low-complexity video on the Internet. Small camera devices recording moving video to, say, a disk or memory chips, can make good use of this profile. It supports up to four objects in the scene with, at the lowest level, the maximum total surface of a QCIF picture. There are three levels for the simple profile, with bit rates ranging from 64Kbps to 384Kbps. The levels also define the maximum total surface for the objects and the amount of macroblocks per second that the decoder must be able to decode. Further, they define the size of various (hypothetical) buffers needed for decoding. While the maximum total object size is defined, the aspect ratio is not prescribed. This allows maximum creative freedom. It could be used, for instance, in a PC screen, where a very wide or very tall object could be created. Several smaller objects could also be made in various places on the screen, and not just confined to a typical QCIF area. The same philosophy is followed for restricting the complexity of the natural video objects in all the visual profiles

Simple profile techniques

1. VOP and P-VOP coding–A separable two-dimensional discrete cosine transform (DCT) defined in **Equation 1**, with u, v, x, y = 0, 1, 2, ... N-1 where x, y are spatial coordinates in the sample domain u, v are coordinates in the transform domain in **Equation 2**, which is applied to each 8 x 8 block inside a macroblock, either an intra frame or residual of a predicted block from P-VOP picture.

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} \quad \text{Eq. 1}$$

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} \\ 1 \end{cases} \quad \text{Eq. 2}$$

Transform DCT coefficients will then be passed to either H.263 or MPEG-2 quantizer according to the mode selected. Afterward, quantized DCT is zig-zag scanned and VLC encoded that is similar to the MPEG-2 standard.

2. 4-MV modes/unrestricted MV mode–In the advanced prediction mode, one/two/four vectors decision is indicated by the MCBPC codeword and field prediction flag for each macroblock. If one motion vector is transmitted for a certain macroblock, this is defined as four vectors with the same value as the motion vector. If four motion vectors are transmitted for the current macroblock, the information for the first motion vector is transmitted as the codeword MVD and the information for the three additional motion vectors is transmitted as the codewords MVD2-4.

An unrestricted motion estimation mode is used for VOP motion estimation. The technique is to improve the motion estimation techniques, especially for VOP-based coding schemes. For unrestricted motion compensation, the motion vectors are allowed to point outside the decoded area of a reference VOP. In the case of a rectangular VOP (nonarbitrary shape), the rectangle is extended in all four directions (left, top, right, and bottom) by 16pixels by repetitive padding.

2. ASP–ASP belongs to the group of streaming video profiles as defined in MPEG-4. It provides the capability to distribute single-layered frame-based video at a wide range of bit rates, targeting video on Internet. ASP combines high coding efficiency and reduced implementation complexity. With B-VOPs and advanced motion-compensation tools, such as quarter-pel motion compensation (QMC) and global motion estimation (GMC), its coding performance is clearly better that of the simple profile. Omitting arbitrarily-shaped objects, on the other hand, makes its implementation complexity much lower than

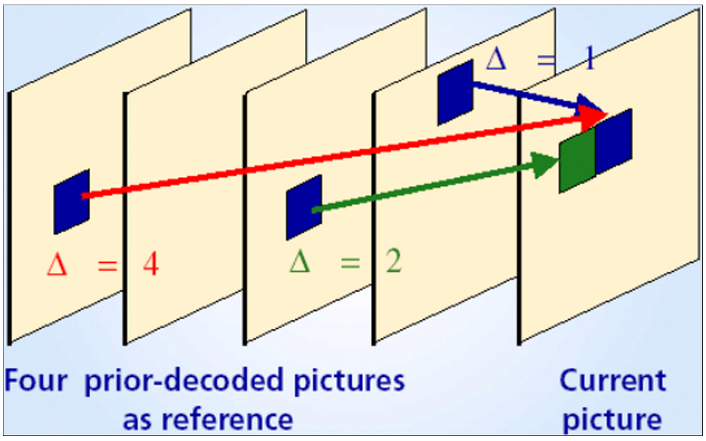

Figure 4: More than one prior-coded picture can be used as a reference for motion-compensation prediction.

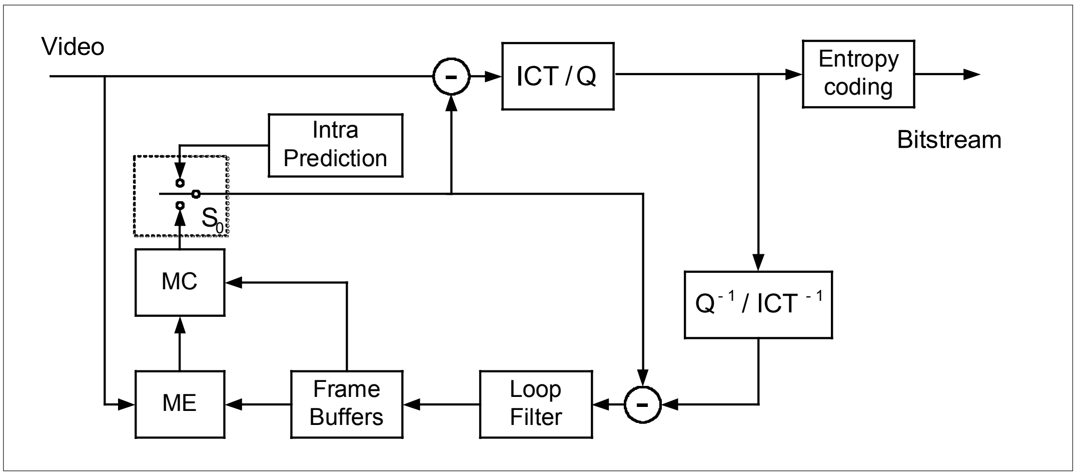

Figure 5: Since they use the same modules, the architecture of AVS looks like that of an H.264 encoder.

that of the advanced coding efficiency profile or even the core profile.

Six levels (L0-L5) have been defined for ASP, with maximum bit rates ranging from 128Kbps (L0, L1) to 8Mbps (L5), and resolutions from QCIF (176 x 144 for L0, L1) to ITU-R 601 (720 x 576 for L5). L4 and L4 additionally include support for interlaced video coding.

ASP techniques

The main goal of QMC is to enhance the resolution of the motion compensation scheme with only small syntactical and computational overhead, leading to more accurate motion description and less prediction error to be coded. QMC will only be applied to the luminance pixels while chrominance pixels will be compensated in half-pixel mode using the appropriate vector scaling. The quarter sample values are found by bilinear interpolation between the surrounding half and integer samples, respectively.

Macroblocks in B-VOPs can be coded either using H.263 such as B-block coding or by MPEG-1 such as B-picture macroblock coding. The main difference lies in the amount of motion vector and quantization-related overhead needed. The H.263-like B-block coding is referred to as direct prediction, besides which the forward, backward and interpolated prediction modes of MPEG-2 B-pictures are supported.

Forward coding mode uses forward motion compensation in the same manner as in MPEG-1/2. The only difference is that a VOP, instead of a picture, is used for prediction. Only one motion vector in half sample units is employed for a 16 x 16 macroblock being coded. Chrominance vectors are derived by scaling of luminance vectors as in MPEG-1/2. Backward coding mode uses backward motion compensation in the same manner as in MPEG-1/2.

GMC's main objective is to encode the global motion in a VOP using a small number of parameters. The core technologies of GMC coding are global motion estimation and prediction, trajectory coding and texture coding for prediction errors.

GMC coding supports the following five transformation models for the warping process: stationary, translational, isotropic, affine and perspective.

Stationary transformation:

$$x' = x,$$
$$y' = y.$$

Translational transformation:

$$x' = x + c,$$
$$y' = y + f.$$

Isotropical transformaton:

$$x' = a x + b y + c,$$
$$y' = d x + e y + f.$$

Affine transformation:

$$x' = a x + b y + c,$$
$$y' = d x + e y + f.$$

Perspective transformation:

$$x' = (a x + b y + c)/(g x + h y + 1)$$
$$y' = (d x + e y + f)/(g x + h y + 1),$$

The video coding layer of H.264/AVC is similar in spirit to other standards such as MPEG-2. It consists of a hybrid of temporal and spatial prediction, combined with transform coding. **Figure 2** shows a block diagram of the video coding layer for a macroblock in H.264/AVC.

In summary, the picture is split into blocks. The first picture of a sequence or a random access point is usually "intra" coded (i.e. coded without using information from other pictures). Each sample of a block in an intra frame is predicted using neighboring samples of previously coded blocks. The encoding process chooses which and how neighboring samples are used for intra prediction, which is simultaneously conducted at the encoder and decoder using the transmitted intra prediction side information.

The remaining pictures of a sequence or between random access points are "inter" coded, which uses prediction (motion compensation) from other previously decoded pictures. The encoding process for inter prediction (motion estimation) consists of choosing motion data, comprising the reference picture and a spatial displacement applied to all samples of the block. The motion data which are transmitted as side information are used by the encoder and decoder to simultaneously provide the inter prediction signal.

The residual of the prediction (either intra or inter), which is the difference between the original and the predicted block, is transformed. The transform coefficients are scaled and quantized, after which they are entropy-coded and transmitted together with the side information for either intraframe or interframe prediction.

The encoder contains the decoder to enable prediction for the next blocks or picture. The quantized transform coefficients, therefore, are inverse-scaled and inverse-transformed in the same way as the decoder side, resulting in the decoded prediction residual. This is added to the prediction, the outcome of which is fed into a deblocking filter.

H.264/AVC

Each macroblock can be transmitted in one of several coding types depending on the slice-coding type. In all slice-coding types, two classes of intra coding types are supported, which are denoted as intra-4 x 4 and intra-16 x 16. In contrast to previous video coding standards where prediction is conducted in the transform domain, prediction in H.264/AVC is always conducted in the spatial domain by referring to neighboring samples of already coded blocks.

When using the intra-4 x 4 mode, each 4 x 4 block of the luma component uses one of nine prediction modes. Beside DC prediction, eight directional prediction modes are specified. When using the intra-16 x 16 mode, which is well suited for smooth image areas, a uniform prediction is performed for the whole luma component of a macroblock. Four prediction modes are supported. The chroma samples of a macroblock are always predicted using a similar prediction technique as for the luma component in intra-16 x 16 macroblocks. Intra prediction across slice boundaries is not allowed in order to keep all slices independent of each other.

In addition to the intra macroblock coding types, various predictive or motion-compensated coding types are specified for P-slice macroblocks. Each P-type macroblock corresponds to a specific partitioning of the macroblock into fixed-size

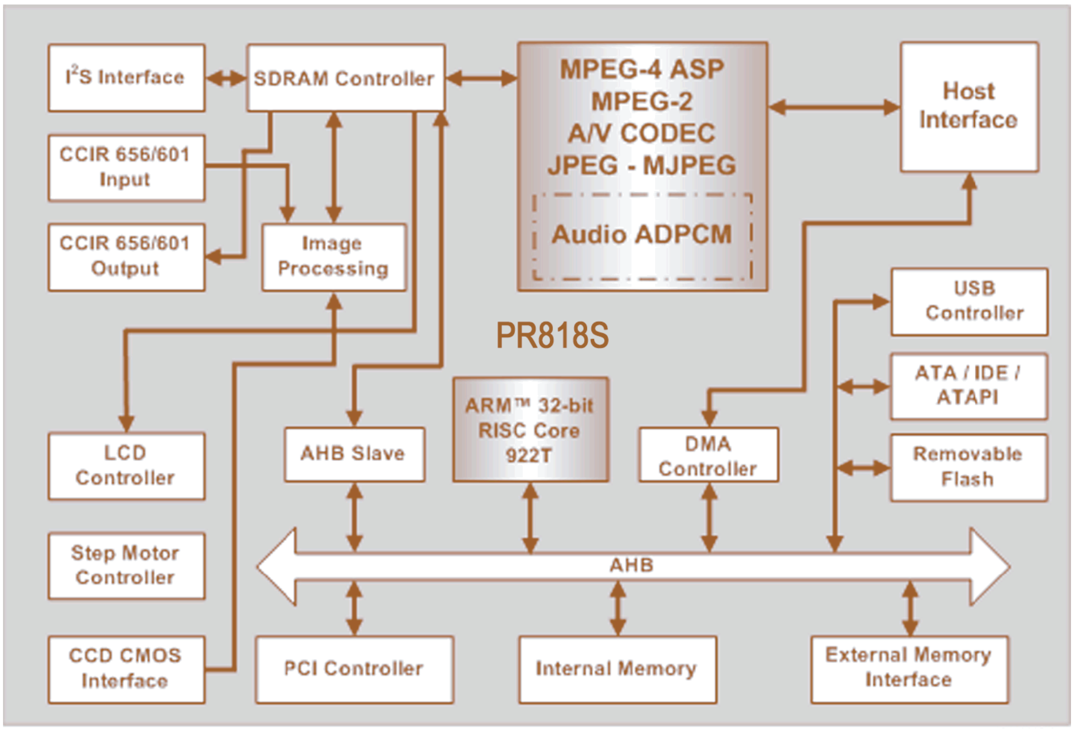

Figure 6: The GoldenREEL family of silicon codecs optimizes full-range Quality Motion Video while overcoming the low bit rates and power constraints of ultra-small form factor products.

blocks used for motion description. Partitions with luma block sizes of 16 x 16, 16 x 8, 8 x 16 and 8 x 8 samples are supported by the syntax corresponding to the inter-16 x 16, inter-16 x 8, inter-8 x 16 and inter-8 x 8 P macroblock types, respectively. In cases where inter-8 x 8 macroblock mode is chosen, one additional syntax element for each 8 x 8 sub-macroblock is transmitted. This syntax element specifies if the corresponding sub-macroblock is coded using motion-compensated prediction with luma block sizes of 8 x 8, 8 x 4, 4 x 8 or 4 x 4 samples. **Figure 3** illustrates the partitioning of a macroblock in H.264 P slices.

The accuracy of motion compensation is a quarter of a sample distance. From motion vector points to non-integer sample position, the prediction signals are obtained by using interpolation at the sub-sample positions. The prediction values at half-sample positions are obtained by applying a one-dimensional 6-tap FIR filter. Prediction values at quarter-sample positions are generated by averaging samples at the integer- and half-sample positions. The prediction values for the chroma components are always obtained by bi-linear interpolation.

The H.264/AVC syntax generally allows unrestricted motion vectors, i.e. motion vectors can point outside the image area. In this case, the reference frame is extended beyond the image boundaries by repeating the edge pixels before interpolation. The motion vector components are differentially coded using either median or directional prediction from neighboring blocks. No motion vector component prediction takes place across slice boundaries.

Similar to previous video coding standards, H.264/AVC also uses transform coding of the prediction residual. However, in H.264/AVC, the transformation is applied to 4 x 4 blocks, and instead of a 4 x 4 discrete cosine transform (DCT), a separable integer transform–with basically the same properties as a 4 x 4 DCT–is used. The transform matrix is given as **Equation 3**.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & -2 & -1 \end{bmatrix} \quad \text{Eq. 3}$$

Since the inverse transform is defined by exact integer operations, inverse-transform mismatches are avoided. An additional 2 x 2 transform is applied to the four DC coefficients of each chroma component. If a macroblock is coded in intra-16 x 16 mode, a similar 4 x 4 transform is performed for the 4 x 4 DC coefficients of the luma signal.

For the quantization of transform coefficients, H.264/AVC uses scalar quantization. One of 52 quantizers is selected for each macroblock by the quantization parameter (QP). The quantizers are arranged so that there is an increase of approximately 12.5 percent in the quantization step size when incrementing the QP by one. The quantized transform coefficients of a block are generally scanned in a zig-zag fashion and transmitted using entropy coding methods.

All transforms in H.264/AVC can be implemented using only additions to and bit-shifting operations on the 16bit integer values.

For transmitting the quantized transform coefficients, a more sophisticated method called context-adaptive variable length coding (CAVLC) is employed. In this scheme, VLC tables for various syntax elements are switched, depending on already-transmitted syntax elements. Since the VLC tables are well designed to match the corresponding conditioned statistics, the entropy coding performance is improved compared with schemes using a single VLC table.

The efficiency of entropy coding can be improved further if context-adaptive binary arithmetic coding (CABAC) is used. On the one hand, the use of arithmetic coding allows the assignment of a non-integer number of bits to each symbol of an alphabet, which is extremely beneficial for symbol probabilities much greater than 0.5. On the other hand, the use of adaptive codes permits adaptation to non-stationary symbol statistics. Another important

Tool	Simple profile (SP)	Advanced simple profile (ASP)
I-VOP	Yes	Yes
P-VOP	Yes	Yes
B-VOP		Yes
AC/DC prediction	Yes	Yes
4 MV/Unrestricted MV	Yes	Yes
Error resilience	Yes	Yes
H.263/MPEG-2 quantizer		Yes
Global MC		Yes
Quarter-Pel MC		Yes
Interlace		Level 4 and up

Table 1: The table summarizes the tools available to SP and ASP.

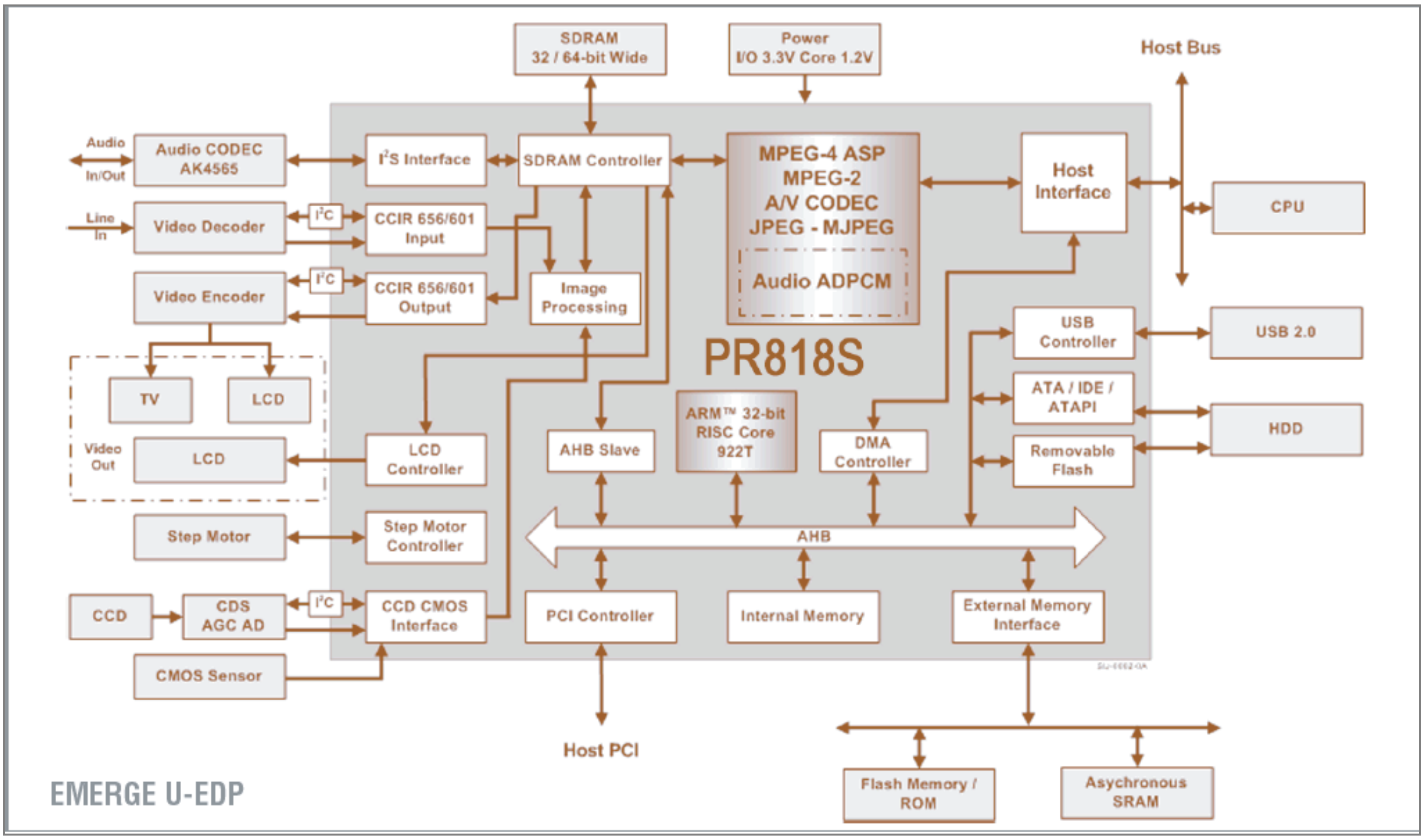

Figure 7: Emerge, the Universal Evaluation Development Platform (U-EDP), delivers a complete solution designed to help OEMs reduce time-to-market

property of CABAC is its context modeling. The statistics of already-coded syntax elements are used to estimate the conditional probabilities. These conditional probabilities are used for switching several estimated probability models. In H.264/AVC, the arithmetic coding core engine and its associated probability estimation are specified as multiplication-free low-complexity methods, using only shifts and table look-ups. Compared to CAVLC, CABAC typically provides a reduction in bit rate of between 10 and 15 percent when coding TV signals of the same quality.

H.264/AVC supports multi-picture motion-compensated prediction. That is, more than one prior-coded picture can be used as a reference for motion-compensation prediction as shown in **Figure 4**.

Both the encoder and decoder have to store the reference pictures used for interpicture prediction in a multipicture buffer.

In addition to the motion-compensated macroblock modes, P-slice macroblock can also be coded in the so-called SKIP mode. For this mode, neither a quantized prediction error signal nor a motion vector or reference index parameter has to be transmitted. In this case, the reconstructed signal is obtained using the motion vector identical to the motion vector for the 16 x 16 block. However, if special conditions hold, a zero motion vector is used instead.

Bit-rate savings relative to existing video coding standards, such as MPEG-2, MPEG-4 ASP and H.263 HLP, can be reduced by a factor of between 2.25 and 2.5 when using H.264/AVC coding for TV or HD video (at broadcast and DVD quality).

AVS video standard

AVS was drafted by the A/V Coding Standard Working Group of China, formed in June 2002 by China's Ministry of Information Industry, with the mandate to establish a national standard for compression, manipulation and digital rights management in multimedia systems.

The AVS working group completed the first ver of the AVS video standard in December 2003. This ver caters mainly to high-definition, high-quality digital broadcasting, digital storage media and other related applications.

Figure 5 depicts the block diagram of an AVS video encoder. The architectures of AVS H.264 encoders look similar as they use the same modules. However, the AVS video codec differs from H.274 in terms of target applications, backward compatibility with MPEG-2, and decoding complexity.

Since each input macroblock needs to be predicted in AVS, the switch as shown in **Figure 5** is used to select the correct prediction method for inter and intra macroblock. The intra predictions are derived from the neighboring pixels in the left and top blocks. The unit size of spatial prediction is 8 x 8 because of 8 x 8 integer transform. The inter predictions are derived from the decoded frames or fields. Four types of block sizes, i.e. 16 x 16, 16 x 8, 8 x 16 and 8 x 8, are supported in AVS. In general, the smaller blocks are seldom used in coding high-resolution video. The precision of the motion vector in inter block is quarter pixel.

The prediction residues are transformed with 8 x 8 integer transform. The scanning order for progressive blocks is still zigzag, similar to that used in MPEG-2. However, a new scanning order is defined for interlacing blocks. AVS uses an adaptive VLC coding technique. There are four different types of Exp-Golomb codebook corresponding to different distributions. Some mapping tables are defined to map coded symbol into a special codebook and its elements.

The sum of prediction and current reconstructed error image forms the reconstructed reference. AVS uses the deblocking filter in the motion compensation loop. The deblocking process directly acts on the reconstructed reference, first across vertical edges and then across horizontal edges. Different image regions and different bit rates need different smoothes. Thus, the deblocking filter is automatically adjusted in AVS, depending on the blocks' activities and QP parameters.

Since MPEG-2 codec and system are extensively deployed in current broadcast systems, the syntax structure of AVS is specially designed to be similar to that of MPEG-

2. This similarity enables the easy application of AVS to the MPEG-2 system.

Currently, AVS supports YUV 4:2:0 and YUV 4:2:2 sampling structure and 8bit sample precision. The 2bit unsigned integer for the chroma format leaves space for other sequence formats such as YUV 4:4:4 or RGB 4:4:4.

AVS techniques

Entropy coding

AVS uses kth-order Exp-Golomb codebook (k=0, 1, 2, 3) for entropy coding. CBP, macroblock coding mode, motion vectors etc. are decoded with 0th -order Exp-Golomp codebook. The quantized transform coefficients are decoded with one of the four codebooks. First, an initial codebook is selected for the first non-zero quantized coefficient. The absolute value of the current decoded coefficient then decides which codebook will be used for the next non-zero quantized coefficient. To efficiently map the coded symbols to the elements of Exp-Golomp codebooks, 19 mapping tables are defined in AVS.

Because of the regularization of the Exp-Golomp codebooks, the AVS decoder does not need to store these codebooks. The syntax element can be decoded with a simple parse followed by an optional look-up table. While the 19 mapping tables only occupy less than 2Kbytes, they enable a strong adaptation to different distributions and high coding performance.

Transform and quantization

AVS uses an 8 x 8 integer transform. To reduce the rounding errors in the dequantization and inverse transform, both are regarded as a process. The operation can be completed within 16bits.

Intraframe prediction

AVS uses intra-frame prediction to improve the performance of intra-coded macroblocks. The intra-frame prediction in AVS is conducted for each 8 x 8 luma/chroma block in the spatial domain. Compared with AVC/H.264's nine intra prediction modes for 4 x 4 luma block, four modes for 16 x 16 luma block and four modes for 4 x 4 chroma block, AVS defines five modes for 8 x 8 luma block and four modes for 8 x 8 chroma block.

Reference pictures

In previous video coding standards such as MPEG-2, bi-directionally predictively-coded pictures (B pictures) use one previous picture and/or one future picture as references. Although predictively-coded pictures (P pictures) use only one previous picture to predict the current picture, the actual reference buffer size in a decoder has to be twice the picture size. AVS fully uses the reference buffer in P buffer coding, where P pictures can use two previous adjacent I/P pictures as references. It improves the coding efficiency and maintains the reference buffer size.

Symmetrical mode for B picture

In the existing coding standards, a macroblock in B pictures can be coded with one of four modes (direct, forward, backward and interpolation). Both forward and backward motion vectors are coded in the interpolation mode. AVS adopts a symmetrical mode to replace the interpolation mode. In this mode, only the forward motion vectors are coded and the backward motion vectors are derived from the correlation between backward and forward. Therefore, at most, one directional motion vector is coded in B macroblocks of AVS.

Weight prediction

Weight prediction can significantly improve coding efficiency, especially for scene transition and illumination change. A simple linear model is used in weighted prediction. The model parameters are coded in the predictive picture header. But, each macroblock is free to decide whether or not to use the weighted prediction.

De-blocking filter

Block-based video coding often produces blocking artifacts, especially at low bit rates. AVS defines an adaptive in-loop deblocking filter to improve the decoded visual quality. The filtering is applied to the boundaries of luma and chroma blocks except for the boundaries of picture or slice. The filtering strength is dependent on macroblock coding type, quantization step, motion vectors and difference among blocks.

Coder	MPEG-4 ASP	H.263 HLP	MPEG-2
H.264/AVC	38.62%	48.80%	64.46%
MPEG-4 ASP		16.65%	42.95%
H.263 HLP			30.61%

Table 2: The table summarizes the average bit-rate savings relative to other video coding standards.

Interlacing coding

For an input interlacing sequence, a picture can be coded as either one frame or two fields (a top field and bottom field). Only the picture-level adaptation between frame and field is allowed in the current ver. When it is coded as two fields, the first field is predicted from the previous decoded fields while the second field is predicted from the first field and the previous decoded fields. These two fields share a picture header but should belong to different slices.

In the AVS video standard, a Jizhun profile has been defined to target SD/HD broadcast and storage. The Jizhun profile contains all techniques defined in the AVS video standard except for the advanced prediction mode. There are four levels defined in AVS. The maximum picture size ranges from 720 x 576 to 1,920 x 1,080 while d the maximum bit rate ranges from 10Mbits/s to 30Mbits/s.

AVS provides the same compression performance as H.264 and as much as two times that of MPEG-2. It has lower computation and memory complexity, and lower required bandwidth for data storage and transmission.

With its acquisition of Protocom Corp., SigmaTel now owns the industry's only full-duplex, hard-wired MPEG-4 ASP SoC solution that promises high-quality video for portable devices.

The GoldenREEL™ family of silicon codecs (PR818S and PR828S) optimizes full-range Quality Motion Video, while overcoming the low bit rates and power constraints of ultra-small form factor products such as digital camcorders, IP cameras, handheld personal video recorders and personal media players. The product also has applications in the consumer electronics and the industrial security/surveillance markets. SigmaTel is currently developing hard-wired products that include H.264/MPEG-4 AVC and SMPTE-VC1 (WMV9) compression technologies.

With its ARM9 core, the GoldenREEL is the centerpiece of Emerge, the Universal Evaluation Development Platform (U-EDP), which delivers a complete solution designed to help OEMs reduce time-to-market.