Jae-Beom Lee
Hari Kalva

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

Springer

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

MULTIMEDIA SYSTEMS AND APPLICATIONS SERIES

Consulting Editor

Borko Furht
Florida Atlantic University

Recently Published Titles:

SIGNAL PROCESSING FOR IMAGE ENHANCEMENT AND MULTIMEDIA PROCESSING edited by E. Damiani, A. Dipanda, K. Yetongnon, L. Legrand, P. Schelkens, and R. Chbeir; ISBN: 978-0-387-72499-7

MACHINE LEARNING FOR MULTIMEDIA CONTENT ANALYSIS by Yihong Gong and Wei Xu**;** ISBN: 978-0-387-69938-7

DISTRIBUTED MULTIMEDIA RETRIEVAL STRATEGIES FOR LARGE SCALE NETWORKED SYSTEMS by Bharadwaj Veeravalli and Gerassimos Barlas; ISBN: 978-0-387-28873-4

MULTIMEDIA ENCRYPTION AND WATERMARKING by Borko Furht, Edin Muharemagic, Daniel Socek: ISBN: 0-387-24425-5

SIGNAL PROCESSING FOR TELECOMMUNICATIONS AND MULTIMEDIA edited by T.A Wysocki,. B. Honary, B.J. Wysocki; ISBN 0-387-22847-0

ADVANCED WIRED AND WIRELESS NETWORKS by T.A.Wysocki,, A. Dadej, B.J. Wysocki; ISBN 0-387-22781-4

CONTENT-BASED VIDEO RETRIEVAL: A Database Perspective by Milan Petkovic and Willem Jonker; ISBN: 1-4020-7617-7

MASTERING E-BUSINESS INFRASTRUCTURE edited by Veljko Milutinović, Frédéric Patricelli; ISBN: 1-4020-7413-1

SHAPE ANALYSIS AND RETRIEVAL OF MULTIMEDIA OBJECTS by Maytham H. Safar and Cyrus Shahabi; ISBN: 1-4020-7252-X

MULTIMEDIA MINING: A Highway to Intelligent Multimedia Documents edited by Chabane Djeraba; ISBN: 1-4020-7247-3

CONTENT-BASED IMAGE AND VIDEO RETRIEVAL by Oge Marques and Borko Furht; ISBN: 1-4020-7004-7

ELECTRONIC BUSINESS AND EDUCATION: Recent Advances in Internet Infrastructures edited by Wendy Chin, Frédéric Patricelli, Veljko Milutinović; ISBN: 0-7923-7508-4

INFRASTRUCTURE FOR ELECTRONIC BUSINESS ON THE INTERNET by Veljko Milutinović; ISBN: 0-7923-7384-7

DELIVERING MPEG-4 BASED AUDIO-VISUAL SERVICES by Hari Kalva; ISBN: 0-7923-7255-7

Visit the series on our website: www.springer.com

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

by

Jae-Beom Lee
Sarnoff Corporation
USA

Hari Kalva
Florida Atlantic University
USA

Springer

Authors:
Jae-Beom Lee
Sarnoff Corp.
Video, Communications and
Networking Systems Division
201 Washington Road
Princeton, NJ 08540
jlee@sarnoff.com

Hari Kalva
Florida Atlantic University
Dept. Computer Science & Engineering
777 Glades Road, PO Box 3091
Boca Raton, FL 33431
hari@cse.fau.edu

Series Editor:
Borko Furht
Florida Atlantic University
Department of Computer Science & Engineering
777 Glades Road, PO Box 3091
Boca Raton, FL 33431
borko@cse.fau.edu

ISBN-13: 978-0-387-71042-6
e-ISBN-13: 978-0-387-71043-3

Library of Congress Control Number: 2008927600

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Contents

Preface

Probably the most interesting and influential class to the authors about video compression was EE E6830 (Digital Image Processing and Understanding) at Columbia University in 1995, offered by adjunct Professors Dr. Netravali, Dr. Haskell and Dr. Puri at AT&T. In the class, they impressed the authors with how such difficult and mysterious statements in video standards could be interpreted/ understood in plain human languages. Since then, the authors had had a dream that similar services could also be provided to interpret difficult video subjects into reasonable level of explanations in the future.

The VC-1 standard is fundamentally the same as WMV-9. WMV-x video compression technologies of Microsoft have long been the most popular over the Internet due to popularity of Microsoft Operating Systems. The technologies were published in August 2005 for the first time in a formal SMPTE document in the name of VC-1, and the official standard then was finalized in April 2006. In contrast, the MPEG committee recently standardized the MPEG AVC (H.264) video coding standard, whose first version was officially published in May 2003, and several subsequent amendments and corrigenda then followed until recently. These two are highly efficient compression standards that can make high-quality video services possible for Digital Storage Media (e.g., Blu-ray DVD or HD DVD) and/or broadband networks applications (e.g., IPTV).

In the industry, on the other hand, video compression text/ reference books have become less useful due to the advance of bitstream analyzer tools such as Interra or Vprove. The tools cross-link statements in the standards in the middle of bitstream verification. In other words, documents explaining in low level are not useful very much any more. Therefore, the focus on the text/

reference books might need to shift from definitions of bits and pieces to ideas/ philosophies about technologies/ tools. This book is designed to present the readers with reasonable understanding and reasoning about why such tools are devised in such ways – as was once done by Dr. Netravali, Dr. Haskell and Dr. Puri. Only the domain is shifted in this book from MPEG-2 to VC-1/ H.264.

The authors are grateful to Professors Anastassiou, Chang and Eleftheriadis (now with the University of Athens, Greece) in the department of Electrical Engineering at Columbia University who helped to shape our understanding about video compression more than a decade ago with the ADVENT project at Center for Telecommunications Research.

A companion website for this book is available at: www.cse.fau.edu/~hari/vc1-h264. The web site will be updatated with useful resources, software tools, and errata. The authors hope that the readers find this book enjoyable and useful.

Dr. Jae-Beom Lee
Princeton, NJ

Dr. Hari Kalva
Boca Raton, FL

April, 2008

Acknowledgements

The authors would like to thank the Series Editor Dr. Borko Furht and Publishing Editor Susan Lagerstrom-Fife for their encouragement and support. The authors would also like to thank Dr. Bill Lin at Sarnoff Corporation and Dr. Gary Sullivan at Microsoft Corporation for their technical and general advice/ comments to improve the quality of this book.

Dr. Jae-Beom Lee also expresses gratitude to colleagues at Sarnoff Corporation for their kind interactions and discussions in deepening video compression knowledge: Arkady Kopansky, Yanjun Xu, Ric Conover, Dennis McClary, Norm Hurst, Mike Isnardi, Hans Baumgartner, Iris Caesar, Jun Hu, Azfar Inayatullah, Joe Frank, Indu Kandaswami, Sandip Parikh, Lin Her, Yumin Zhang, Mike Patti, Khaleel Udyawar, Saurabh Shandilya, Prashant Laddha, Bedarakota Madhu Sudhan, Anup Mankar, Vishvanath Deshpande, Ramanan Narayanan, Mattamari Seshagiri Srividya, Veena Parashuram, Penmestsa Raju and Iyengar Sridhar.

To Bong-Gum Lee, Yun-Hee Jang and Da-En Lee – the three most important women in my life

Jae-Beom Lee

To my parents

Hari Kalva

1. Multimedia Systems

1.1 Overview of MPEG-2 Systems

Systems and Synchronization

The video compression system is a part of a multimedia system that provides a means of multiplexing several types of multimedia information into one single stream [haskell:MPEG2, yu:MPEG2systems, ITU:MPEG2systems]. Summarizing a global picture of such multimedia systems helps further understanding VC-1 and H.264/ AVC video compression topics. This section describes issues regarding general aspects of system timing and media synchronization. It then discusses solutions and their practical implications including sender/ receiver system timing synchronization (i.e., Transport synchronization), video/ audio synchronization (i.e., Inter-media synchronization) and encoder/ decoder resource synchronization (i.e., Resource synchronization). It further details practical examples to implement such synchronization mechanisms on MPEG Systems.

SMPTE RP227 recommends VC-1 bitstream encoding provisions that define a minimum set of rules for the carriage of a VC-1 elementary stream in an MPEG-2 Transport Stream with additional intention to provide a generic means of carrying a VC-1 video elementary stream in an MPEG-2 Program Stream as used by the DVD Forum. In addition, Amendment 3 of ITU-T Recommendation H.222.0 recommends H.264/ AVC bitstream encoding provisions that define a minimum set of rules for the carriage of a H.264/ AVC elementary stream in an MPEG-2 Transport Stream, with additional intention to provide a generic means of carrying a H.264/ AVC video elementary stream in an MPEG-2 Program Stream as used by the DVD Forum. In other words, both VC-1 and H.264/ AVC standards adopt MPEG-2 Systems as a major system encapsulation/ transport mechanism due to its popularity in the real world [SMPTE:VC1systems, ISO:MPEG2systems.amd].

MPEG-2 Systems have been used for an extraordinary number of applications that require solid transport delivery or local playback mechanisms, where strict transport / inter-media/ resource

synchronizations are recommended. However, mobile applications like cellular video streaming do not require strict transport timing synchronization. Streaming video applications using the TCP-IP protocol, where no Quality of Service (QoS) is provided, do not have a fixed delay (or a constant incoming bitrate). In such a case, transport synchronization could be ignored. Local timer-based inter media synchronization might still need to be performed though.

In general, for any video compression/ transmission standard, the inter media synchronization is performed using a Presentation Time Stamp (PTS) that dictates when a particular media unit (for example, video picture or audio frame) should be played back. The nature in which PTS is decided for audio and video units varies from standard to standard. The PTS may be described at the media level (for example, within the video bitstream in H.264/ AVC), Packetized Elementary Stream (PES) level (for example, within PES Header in MPEG-2 or VC-1) or transport level (for example, MPEG-2 video on RTP packets).

The following sections describe in detail mechanisms for transport, media and resource synchronizations in any multimedia systems.

Figure 1-1 Sender Receiver Synchronization in Communication Systems

Transport Synchronization

Compressed multimedia generally has of two forms in applications – networked playback or local playback. Local playback is much more relaxed in terms of timing since all data is already available in the player. On the other hand, networked playback requires more strict system timing as any loose synchronization of sender/ receiver ends up with playback

jitter. Therefore, the discussion in this section mainly covers the networked scenario. Figure 1-1 depicts a general situation of a sender and a receiver, where typical multimedia communication systems assume delivery mechanisms on local clocks.

In standards like MPEG-2, the operating clock is defined as 27MHz. However, there are no perfectly matched two local clocks – one in the sender, the other in the receiver – in terms of speed in real life. Since the data transmission rate in modern communication systems is extremely high, constantly faster or slower (even with the slightest imaginable mismatch) clock at receiver side makes the receiver buffer underflow or overflow. For example, 20Mbps bandwidth communication accumulates 20 million bits in the receiver side buffer when two clocks are out of synchronization only for 1 sec.

To manage receiver side buffer for a long time (for example, two to three hours movie time), the time-average speed of two local clocks should be exactly the same. Note that slight jittering between two clocks is not a problem as long as the time-averages of two local clocks are exactly matched and a reasonable size of receiver side buffer is allowed.

To implement the same speed of two local clocks in terms of time-average, PLL and Voltage Controlled Oscillator (VCO) are used in MPEG Systems with a special type of Time Stamp called Program Clock Reference (PCR), where Time Stamp is defined to be nothing but a sampling value of a counter that runs on a local clock as shown in Figure 1-4. The MPEG-2 standard specifies PCR as being driven by a 27MHz clock.

In the delivery of MPEG multimedia information, sender and receiver synchronization is achieved with PCR in Transport Stream (TS) packets. In the TS packet header, PCR Time Stamp is written based on the local clock of the sender (i.e., system encoder).

When a receiver receives the first PCR in a TS packet, it copies PCR as an initial value into a counter increased by the local clock of the receiver. With any PCR received later, the receiver compares it with its local counter to determine whether the receiver local clock is faster or slower than that of the sender. The differential value of the received PCR and the local counter is fed back to adjust the speed of the local clock of the receiver side with VCO. For example, let's say that a received PCR is 2000, while local counter at the moment is 2001. This implies that the

operating speed of the local clock at the receiver is faster than that of the sender. Such difference can be adjusted with VCO at the receiver. Once transport synchronization is achieved with Time Stamps in the transport layer, inter-media synchronization is next to be considered with Time Stamps in the media layer.

This synchronization mechanism between two close-speed running local clocks is a fundamental means in Asynchronous Transfer Mode (ATM) communications. ATM was supposed to work best for point-to-point communications and MPEG-2 TS packet was designed to be accommodated to four ATM packets with the fixed length of 188 bytes while at MPEG-2 System standardization effort. In contrast, Synchronous Transfer Mode (STM) requires nation-wide distribution of a single master clock to all nodes, thus making the time-average of the speed of the clocks the same with slight jittering. This jittering is not a big problem as aforementioned as long as buffer size is enough. Therefore, STM does not require PLL-VCO. SONET from AT&T is a famous STM backbone network that allows direct add/ drop multiplexing without intermediate demux necessity down to any layer.

MPEG-2 Systems define a provision about deviation of system clock frequencies among individual implementations as follows:

$$27000000 - 810 \le system\_clock\_frequency \le 27000000 + 810$$

and rate of change of *system_clock_frequency* with time $\le 75 \times 10^{-3} Hz / s$. (1-1)

Therefore, the speeds of two local clocks should be reasonably close at the beginning.

The *system clock frequency* 27MHz is chosen for historical reasons. A common frequency from which NTSC (525/60) and PAL (625/50) line and field rates can be derived is 4.5MHz as shown in 4.5MHz/143 (a.k.a., 2fH for NTSC) and 4.5MHz/144 (a.k.a., 2fH PAL), respectively. The luma sampling rate was chosen as 13.5MHz (a.k.a., $4.5 \times 3MHz$), three times of the common frequency. To apply luma sampling rate to YUV4:2:2 type format, the sampling rate is chosen as 27MHz (a.k.a., 13.5MHz + 6.75MHz + 6.75MHz).

Inter-Media Synchronization with PTS

In general applications, more than two medias are captured/ compressed in the encoder (sender) side. Even when only two medias (video and audio) are captured simultaneously, the absolute time order cannot be maintained due to time division packet multiplexing in the delivery mechanism.

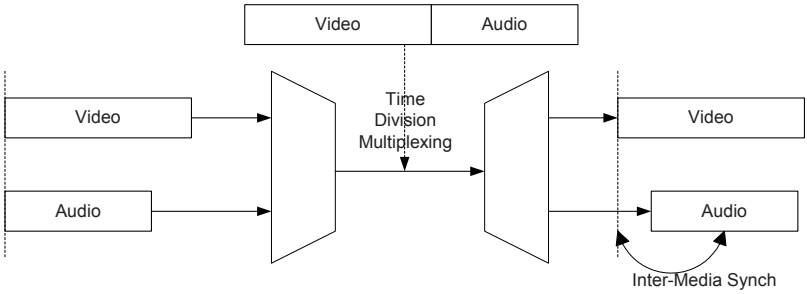

Figure 1-2 Time Division Multiplexing

Packet multiplexing generally causes different delay between media due to serialization as shown in Figure 1-2. On top of the aforementioned "packet multiplexer jitter," "network delivery jitter" is also added. Since concurrent media should play back simultaneously, an inter-media synchronization mechanism is to be devised.

To this end, Access Units (AU) for different media and Presentation Time Stamp (PTS) are defined. Two framed-media captured at the same time share the same PTS. And, PTS is ideally attached to each AU. If a PTS is not found in a certain AU, the value should be inferred based on the values recently received at the receiver. The encoder commands every decoder when a specific AU should be displayed with a specific PTS command. At receiver side, inter-media synchronization can be achieved with two media being buffered and played back based on PTS. Note that network delivery jitter and packet multiplexer jitter can be eliminated through buffering at the receiver side. PTS is a 90KHz-based Time Stamp locked to the 27MHz-based PCR in MPEG-2.

Resource Synchronization with DTS

The Video Buffer Verifier (VBV), which is the Hypothetical Reference Decoder (HRD) in VC-1 or H.264, is the MPEG-2 hypothetical buffer model for a video decoder. VBV is meant to connect to the output of an encoder virtually while encoding. As bitstreams are created, the VBV fullness must be checked to ensure that it does not overflow or underflow. A dummy decoder is assumed with certain predefined behavior such as infinite processing speed at decoding. The encoder sends commands about buffer size policy to every decoder with decoder input buffer size parameters (such as VBV_buffer_size in MPEG-2, HRD_BUFFER in VC-1 and Cpb_size_scale and Cpb_size_value_minus1 in H.264), and may also send commands to every decoder when the decoding fetch-out has to happen with Decoding Time Stamp (DTS).

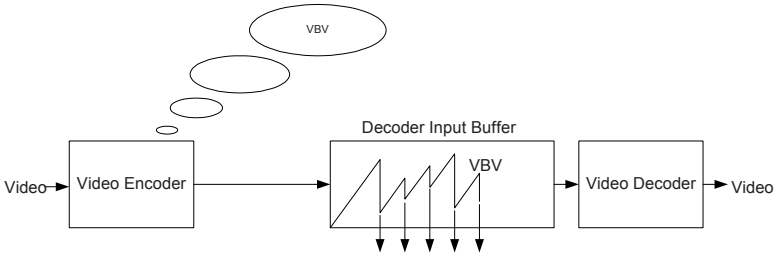

Figure 1-3 VBV Model and Resource Synchronization

The encoder takes full advantage of the remaining buffer resource in the VBV for bit allocation policy at GOP-level, Frame-level and MB-level.

If a decoder doesn't follow the encoder's commands regarding DTS, the remaining buffer resource assumed by the encoder can be different from that in actual decoder input buffer – this could cause an unexpected behavior resulting in overflow or underflow at the decoder input buffer. Therefore, a decoder has to do its best to conform to DTS commands of the encoder. DTS is ideally attached to each AU, as is PTS. If a DTS is not found in a certain AU, the value should be inferred based on the values recently received at the receiver. DTS is a 90KHz-based Time Stamp locked to the 27MHz-based PCR in MPEG-2.

DTS/ PTS Locking Mechanism to PCR

A PCR is a sampling value of a counter that runs on a 27MHz-driven local clock. As such, a DTS and a PTS are sampling values of a counter that runs on a 90KHz-driven local clock. PCR is present in a 42-bit Time Stamp, while DTS/ PTS is present in a 33-bit Time Stamp in MPEG-2 as shown in Figure 1-4. For example, 0 and 2100 as shown in Figure 1-4 are two sampling values of the counter that runs at 27MHz-driven local clock – PCR Time Stamps. Also, 0 and 7 are two sampling values of the counter that runs at 90KHz-driven local clock – DTS/ PTS Time Stamps. Since a Program is defined to share a common time-base in MPEG Systems, all media streams (videos and audios) in a single program are captured and time-tagged based on a single common clock. Note that 27MHz is divisible by 90KHz with a factor of 300. In other words, a 90KHz clock can be generated from a single source of 27MHz clock and a DTS/ PTS can be directly compared/ locked to PCR by multiplying it by 300 as shown in Figure 1-4.

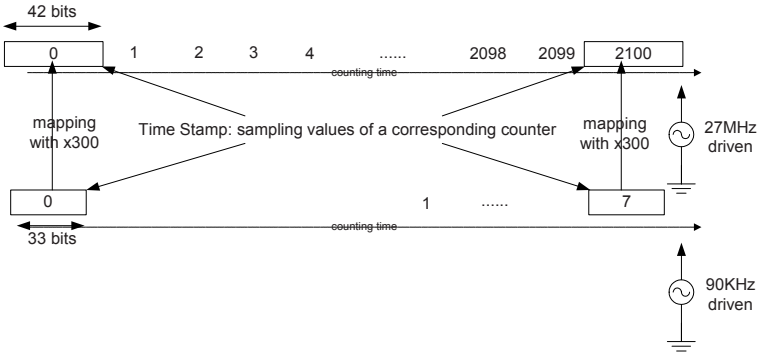

Figure 1-4 DTS/ PTS Locking to PCR

In summary, MPEG-2 Systems synchronization is performed in two levels – Transport level and Media level. First, Transport synchronization is carried out with a PCR Time Stamp to prevent overflow or underflow in communication buffers. Second, Inter-media synchronization and Resource synchronization are performed at Media level with PTS/ DTS locked on Transport clock already synchronized between the sender and the receiver (or potentially multiple receivers).

General MPEG System Architecture

A Program can contain multiple media streams of video and audio. To handle multiple streams in a synchronized manner, the cores consist of three blocks – system demux, video decoders and audio decoders as shown in Figure 1-5. The system demux de-packetizes and routes video, audio and systems data to the appropriate buffers. PCR data is extracted from system data for Transport synchronization. The first PCR data is copied to a local counter of the receiver as an initial Time Stamp and the counter is increased by a PLL-VCO rectified 27MHz local clock.

The AUs of video and audio are all called "frames." Video decoders are controlled by DTS and PTS together since sometimes display order and decoding order are different. However, audio decoders are controlled only by PTS since playback order and decoding order are the same. In the video controller, DTS values are multiplied by 300 to compare with PCR value at the local counter. When the two Time Stamps are the same, fetch-out of video bitstream and immediate decoding is initiated by the video decoding controller. When the PCR hits the time of $PTS \times 300$, the corresponding AU is played in the video display. In the audio controller, the PTS values are multiplied by 300 to compare with PCR at the local counter. When the two Time Stamps are the same, fetch-out of audio bitstream and immediate decoding are initiated by the audio decoding controller.

Unlike video decoding, the audio is supposed to play back immediately after decoding in theory. However, actual implementation is slightly different in real life. The theoretical assumption that dummy decoder has infinite processing speed is not correct in practice. Audio and video decoders take some time to decode one frame-worth of data. If frame-based pipelining is used for decoder implementation, even more than one frame time delay is to be expected. In such a case, the delay is mainly dependent on how many frame-based pipelining stages are used internally in the design. Therefore, PTS values are modified to accommodate such delay. Since the video decoder requires more computation, PTS values of audio are incremented to accommodate video processing delay. In other words, audio play-out buffers are needed to hold up data until the concurrent video data is ready for inter-media synchronization.

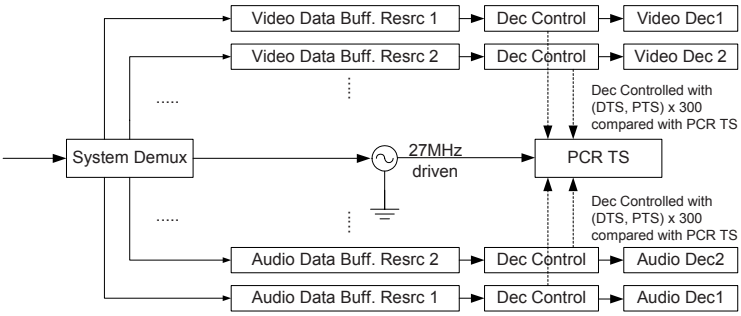

Figure 1-5 MPEG System Architecture

Processor Mapping of MPEG System

Two important functional blocks of MPEG Systems are, first, System Demux and Clock Recovery and, second, Presentation Scheduler as shown in Figure 1-6.

System Demux and Clock Recovery function handles depacketization and extraction of PCR information with rectification. The Presentation Scheduler handles Inter-media synchronization and Resource synchronization based on DTS/ PTS.

One implementation of System functions can be to put the two functions on the audio processor. Audio processing is less computationally intensive, compared with video processing. This is mainly because the amount of 2-D video data handled is much more than that of 1-D audio data. Therefore, a certain amount of computational room can be available to System implementation.

One popular practical implementation of the video Presentation Scheduler function is based on Interrupt Service Routine (ISR), which is based on an interrupt signal (Field and/or Frame time) generated by the display processing device as shown in Figure 1-7. At another interrupt signal point (Direct Memory Access(DMA)-done, synchronization for decoding start), System Demux and Clock Recovery function can be performed since typically the outside Host delivers bitstream based on Field or Frame time. The key idea behind this is to guarantee secure availability of resources at the decoder – one picture decoding is performed only when one decoded picture is displayed/ released from the display buffer at PTS time.

Decompressed audio frames reside at the audio data buffer resources. Each audio frame is tagged with PTS in the memory. When a video frame is displayed, audio frames with PTS within the next one Field/ Frame tick time are serialized/ played back together. The PTS of every audio frame practically need not be verified during such a tick time as long as audio PTS is locked to that of video at the start of each Field/ Frame tick time.

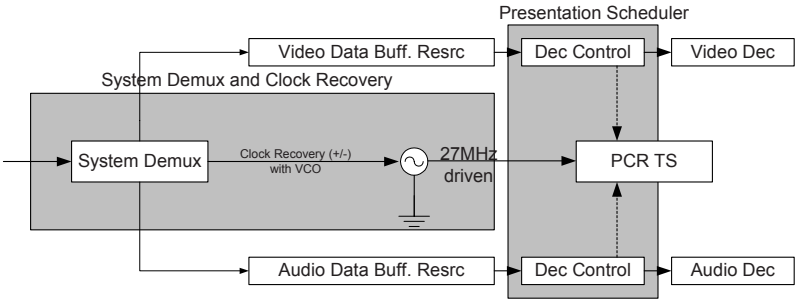

Figure 1-6 A Typical Processor Mapping of MPEG System

Mobile applications like cellular video streaming don't require strict timing. Still, a lot of streaming video applications depend on TCP-IP protocol where no Quality of Service is provided. So, the server and clients have a side channel to negotiate in a timely manner. In such a case, PCR Clock Recovery could be ignored and Timer-based interrupt signals (for DTS and PTS) are sufficient. The implementation mentioned above can be slightly modified to accommodate different application scenarios.

Display and Decoder Interlocking Mechanism

Display processors can be defined in various ways. A basic function of a display processor is to perform DMA from picture memories and to convert from YUV format to certain display signals physically pre-defined. It could generate outputs for analog TVs and/or for digital displays. It could also generate interrupt signals at Field and/or Frame time. The input interface has a couple of Luma/ Chroma data pointers and Go-bit for an immediate action.

Once a display processor starts working, it cannot typically be stalled or delayed by any means (except clock rate rectification based on PLLed local clock). It is continuously displaying empty or meaningful data at the very regular rate of 30 frames/second or any other pre-defined rate. Since the display unit or processor works almost independently, other parts of the SOC need to be synchronized with the display processor operation.

Interlocking of display and decoder is the major means to secure buffer resources at the decoder. Two interrupts generated by the display processor are generally required to inter-lock between display processor and the decoder – 1) Field and/or Frame time interrupt and 2) DMA-done interrupt. The first interrupt tells about the time (Field and/or Frame) point when the display scanning line reaches those points. The second interrupt indicates the time when DMA from external memory to the display processor is done. How to use these two interrupts for interlocking is an implementation issue. However, the decoder is generally allowed to go ahead for next unit (Field or Frame) decoding at the DMA-done interrupt. Note that the decoder stalls for a while after a data unit (Field or Frame) decoding is finished.

```
While (1) { //infinite loop++
          display and decoder synch point;
          main decoding part;
          buffer reordering for display;
} //infinite loop--
```

Figure 1-7 Display and Decoder Inter-locking

The "buffer reordering for display" is performed at the end of the decoding stage. Generally, display order comes after a couple of frame/field ticks. The only exceptions are I and P pictures. When the next reference picture is reached, the previous reference picture can be displayed. This is not true generally for advanced video codecs such as H.264, though.