The Multistandard Full Hd Video-Codec Engine On Low Power Devices

B.Susma
(M. Tech). Embedded Systems.
Aurora's Technological & Research Institute. Hyderabad.

B.Srinivas
Asst. professor. ECE,
Aurora's Technological & Research Institute. Hyderabad.

ABSTRACT

Full HD video coding has become an essential requirement. The functionality ranges from single channel encoding (camcorder) and decoding (video playback), to more complex use cases such as video conferencing (encode + decode) and transcoding. There is increasing demand for higher visual quality, and the wide spectrum of video content requires support for multiple standards. Low power and area efficiency are also equally critical requirements for these applications. Low power video codecs employ hardwired implementations targeted to a specific standard and address either encode or decode. Low power is typically achieved by using single codec optimized circuitry and massive parallelism so that the design can run at lower voltage and frequency. Multicore programmable processors can support multiple standards, but are not scalable to meet full HD performance of complex standards like H.264 High Profile (HP) in a low-power CMOS process, and are inefficient in terms of area and power. The approach of decoupling stream processing and pixel processing, and using a 2 macroblock pipeline helps meet the performance, but introduces a frame delay resulting in higher latency. Video conferencing not only requires low latency but also requires encoding with a fixed limit on the number of bits per slice. The 2 macro block pipeline cannot support this requirement without impacting the performance or the bit rate. The TI DM3730 application processor addresses these challenges with a dedicated video coding engine built with algorithm-specific hardware accelerators.

INTRODUCTION

In this paper, we present IVA-HD, a true multistandard, programmable, full HD video coding engine which adopts optimal hardware-

software partitioning to achieve the low-power and area requirements of the OMAP 4 processor. Unlike the approach of using separate IPs for encoder and decoder, IVA-HD uses an integrated codec engine which is area efficient, as most of the decoder logic is reused for the encoder. IVA-HD is architected to perform stream-rate and pixel rate Processing in a single pipeline (that processes one 16×16 macroblock at a time), so as to support the latency requirements of video conferencing. Some of the key features of IVA-HD include: (i) FIFOs between accelerators to allow asynchronous pipelining, resulting in up to 20% fewer cycles vs. a synchronous pipeline. (ii) Distributed control allows an accelerator to gate its clocks as soon as it completes macroblock processing. (iii) At the end of each frame,

IVA-HD goes into ultra-low power mode consuming zero dynamic power between frames. (iv) A run-time configurable pipeline enables switching off the deblocking filter in the case of non-referenced B frames in H.264 HP encoding.

This saves 10% of IVA-HD power and 15% of DDR bandwidth for B frames. (v)Local playback of recorded video uses closed-loop decode instead of a universal decoder, thus reducing DDR bandwidth by 30-40%. (vi) Simultaneous (vii) ME supports an arbitrary search pattern, variable search range, multiple reference frames, weighted prediction, multiple partitions and different frame types (P, B, Hierarchical-P/B). This enables high quality and allows trading off quality for power/DDR bandwidth. Beyond the video engine, the application processor design also directly impacts video coding performance and power dissipation.

Power management techniques, including Dynamic Voltage and Frequency Scaling (DVFS), Adaptive Voltage Scaling (AVS) and Adaptive Body Bias (ABB). For a given temperature and supply voltage, the logic delay at the fast process corner can be 10-15% lower than the delay at the slow process corner. AVS exploits this by lowering supply voltage by 13-17% for silicon at fast process corner, while meeting the overall performance. IVA-HD has a dedicated voltage domain, which enables setting its frequency and voltage independent of the rest of the system. For example, when decoding content of simpler standards, such as MPEG2, IVA-HD can operate at half the frequency (133MHz) and lower voltage.

DDR bandwidth is essential to lower system power. Some key power optimization techniques at the OMAP 4 processor level include: (i) Dedicated on chip buffer enables fetching a reference frame pixel only once per frame, resulting in 40-50% reduction in DDR bandwidth and tolerance for variable latency when accessing data from the shared SDRAMs. (ii) OCP 2.2-based chip interconnect with side-band signals to maximize DDR and on-chip interconnect efficiency. (iii) Low-power secondary CPU which manages the application layer functionality of video use cases, enabling the main CPU to operate at lower frequency, saving power. IVA-HD power dissipation varies depending on the characteristics of the video data and supply voltage scaling using AVS. OMAP 4 silicon power measurements with various HD video streams, including complex H.264 HP L4.0 streams show the decode power to range

from 65-to-95mW. For H.264 HP encode, the power dissipation ranges from 100-to-145mW.

No	Codec	Applications	Details	IVA-HD Clocking
1	H.264 Encoder	Most popular latest generation technology deployed across broadcast, surveillance, conferencing, cell phone, infrastructure, camera, set-top box, Blu-ray™ HD-DVD etc	BP/MP/HP@L4.0, 1080p/i @ 30 fps, 20 Mbps, Standard quality	233 MHz
2	H.264 Encoder		BP/MP/HP@L4.0, 1080p/i @ 30 fps, 20 Mbps, High quality	266 MHz
3	H.263 & MPEG4 Encoder	Cell phones, MMS, Camera, Video telephony & conferencing	SP@L6, 1080p/i @ 30 fps, 20 Mbps, Standard quality	233 MHz
4	H.263 & MPEG4 Decoder		SP@L6, 1080p 30fps, 20 Mbps	133 MHz
5	MPEG4 Decoder	XVID & DivX Movies	SP/ASP@L6, 1080p/i, 30fps, 20 Mbps	196 MHz
6	MPEG1/2 Decoder	VCD, DVD, Blu-ray™ HD-DVD, Set-top box etc	SP/MP@HL, 1080p/i 30fps, 20 Mbps	133 MHz
7	WMV9 & VC1 Encoder & Decoder	Blu-ray™ HD-DVD, Microsoft products	SP/MP/AP@L3, 1080p/i @ 30 fps, 20 Mbps, Typical worst case	266 MHz
8	MJPEG Encoder & Decoder	Camera content, Automotive & Vision	Baseline, 1080p 30fps, 20 Mbps, YUV 4:2:0	133 MHz
9	RV™ 8/9/10 Decoder	Streaming, China market	1080p 30fps, 20 Mbps	266 MHz
10	ON2™ VP6/7 Decoder	Flash video, Skype video conferencing	1080p 30fps, 20 Mbps	233 MHz
11	MVC Encoder & Decoder	3D Stereoscopic record & playback	720p, 30fps, 20 Mbps	233 MHz
12	SVC Encoder & Decoder*	Video conferencing, Surveillance & broadcast	SVC-Temporal & SVC-Simulcast 480p+ 720p or 1080p15+ 1080p30	266 MHz
13	AVS1.0 Encoder & Decoder	Chinese national standard	1080p/i 30fps, 20 Mbps	266 MHz

OVERVIEW OF H.264

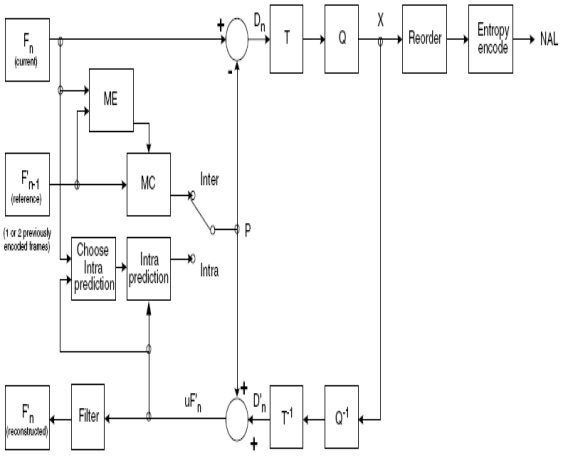
H.264 has a narrower scope than MPEG-4 Visual and is designed primarily to support efficient and robust coding and transport of rectangular video frames. Its original aim was to provide similar functionality to earlier standards such as H.263+ and MPEG-4 Visual (Simple Profile), but with significantly better compression performance and improved support for reliable transmission. Target applications include two-way video communication (videoconferencing or Video telephony), coding for broadcast and high quality video and video streaming over packet Networks. Support for robust transmission over networks is built in and the standard is designed to facilitate implementation on as wide a range of processor platforms as possible.

The standard is specified in more detail than many earlier standards (including MPEG-4 Visual) in an attempt to minimize the possibility for misinterpretation by developers. The detailed level of specification means the standard document is relatively long and makes some Sections of the standard difficult to follow.

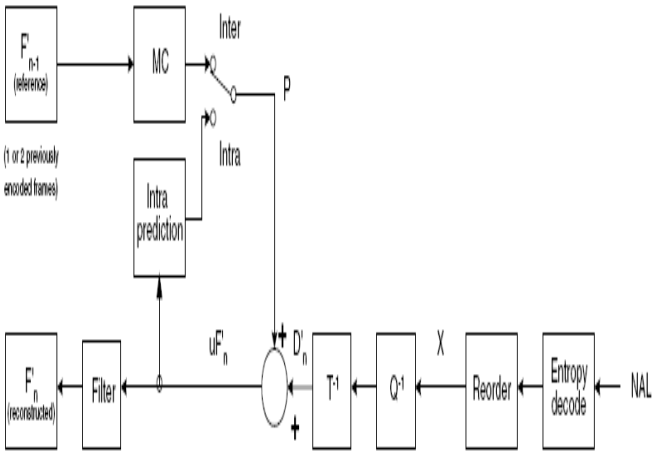
H.264 CODEC

In common with earlier standards (such as MPEG1, MPEG2 and MPEG4), the H.264 standard does not explicitly define a CODEC (encoder / Decoder pair). Rather, the standard defines the syntax of an encoded video bit-stream together with the method of decoding this bit stream. In practice, however, a compliant encoder and decoder are likely to include the functional elements shown in Figures. Whilst the functions shown in these Figures are likely to be necessary for compliance, there is scope for considerable variation in the structure of the CODEC. The basic functional elements (prediction, transform, quantization, entropy encoding) are little different from previous standards (MPEG1, MPEG2,MPEG4, H.261, H.263); the important changes in H.264 occur in the details of each functional element.

The Encoder includes two dataflow paths, a "forward" path (left to right, shown in blue) and a "Reconstruction" path (right to left, shown in magenta). The dataflow path in the Decoder is shown from right to left to illustrate the similarities between Encoder and Decoder.



H.264 Encoder



H.264 Decoder

Encoder (forward path)

An input frame Fn is presented for encoding. The frame is processed in units of a macroblock (Corresponding to 16x16 pixels in the original image). Each macro block is encoded in **intra** or **inter** mode.

In either case, a prediction macro block P is formed based on a reconstructed frame. In Intra mode, P is formed from samples in the current frame n that have previously encoded, decoded and reconstructed to produce a residual or difference macro block Dn. This is transformed (using a block transform) and quantized to give X, a set of quantized transform coefficients. These coefficients are re-ordered and entropy encoded. The entropy-encoded coefficients, together with side information required to decode the macroblock (such as the macroblock prediction mode, quantizer step size, motion vector information describing how the macroblock was motion-compensated, etc) form the compressed bit stream. This is passed to a Network Abstraction Layer (NAL) for transmission or storage. (UF'n in the Figures; note that the **unaltered** samples are used to form P). In Inter mode, P is formed by motion-compensated prediction from one or more reference frame(s). In the Figures, the reference frame is shown as the previous encoded frame F'n-1; however, the prediction for each macroblock may be formed from one or two past or future frames (in time order) that have already been encoded and reconstructed.

The prediction P is subtracted from the current macroblock

Encoder (reconstruction path)

The quantized macroblock coefficients X are decoded in order to reconstruct a frame for encoding of further macro blocks. The coefficients X are re-scaled (Q-1) and inverse transformed (T-1) to produce a difference macroblock Dn'. This is not identical to the original difference macroblock Dn; the quantization process introduces losses and so Dn' is a distorted version of Dn.

The prediction macroblock P is added to Dn' to create a reconstructed macroblock uF'n (a distorted version of the original macroblock). Alter is applied to reduce the effects of blocking distortion and reconstructed reference frame is created from a series of macroblocks F'n

Decoder

The decoder receives a compressed bit stream from the NAL. The data elements are entropy decoded and reordered to produce a set of quantized coefficients X. These are rescaled and inverse transformed to give Dn' (this identical to the Dn' shown in the Encoder). Using the header information decoded from the bitstream, the decoder creates a prediction macroblock P, identical to the original prediction P formed in the encoder. P is added to Dn' to produce uF'n which this is altered to create the decoded macroblock F'n. It should be clear from the Figures and from the discussion above that the purpose of the reconstruction path in the encoder is to ensure that both encoder and decoder use identical reference frames to create the prediction P. If this is not the case, then the predictions P in encoder and decoder will not be identical, leading to an increasing error or "drift" between the encoder and decoder.

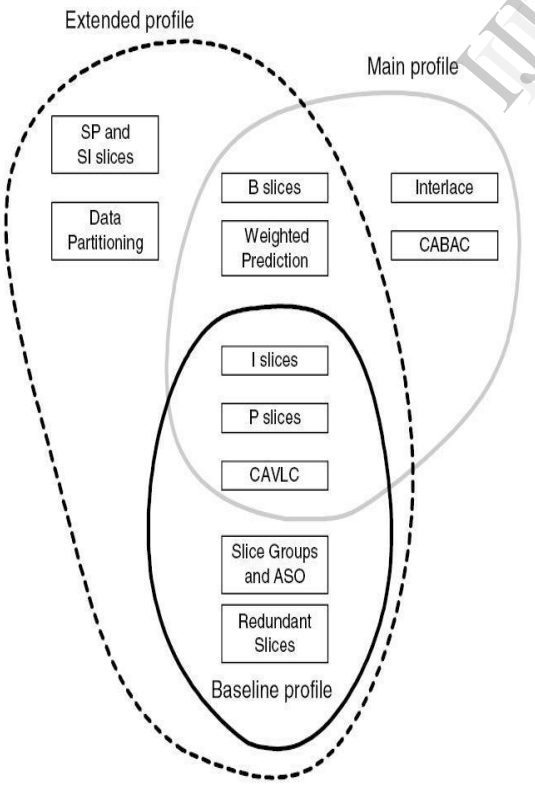
H.264 STRUCTURE

Profiles and Levels

H.264 defines a set of three *Profiles*, each supporting a particular set of coding functions and each specifying what is required of an encoder or decoder that complies with the Profile.

The *Baseline Profile* supports intra and inter-coding (using I-slices and P-slices) and entropy coding with context-adaptive variable-length codes (CAVLC). The *Main Profile* includes support for interlaced video, inter-coding using B-slices, inter coding using weighted prediction and entropy coding using context-based arithmetic coding (CABAC). The *Extended Profile* does not support interlaced video or CABAC but adds modes to enable efficient switching between coded bit streams (SP- and SI-slices) and improved error resilience (Data Partitioning).

Figure shows the relationship between the three Profiles and the coding tools supported by the standard.



Slices

A video picture is coded as one or more slices, each containing an integral number of macro blocks from 1 (1 MB per slice) to the total number of macroblocks in a picture (1 slice per picture) The number of macroblocks per slice need not be constant within a picture. There is minimal inter-dependency between coded slices which can help to limit the propagation of errors. There are five types of coded slice and a coded picture may be composed of different types of slices. For example, a Baseline Profile coded picture may contain a mixture of I and P slices and a Main or Extended Profile picture may contain a mixture of I, P and B slices.

Macroblocks

A macroblock contains coded data corresponding to a 16 × 16 sample region of the video frame (16 × 16 luma samples, 8 × 8 Cb and 8 × 8 Cr samples) and contains the syntax elements. Macroblocks are numbered (addressed) in raster scan order within a frame.

Inter Prediction

Inter prediction creates a prediction model from one or more previously encoded video frames Or fields using block-based motion compensation. Important differences from earlier standards include the support for a range of block sizes (from 16 × 16 down to 4×4) and fine subsample motion vectors (quarter-sample resolution in the luma component).

Tree structured motion compensation

The luminance component of each macroblock (16×16 samples) may be split up in four ways and motion compensated either as one 16×16 *macroblock partition*, two 16×8 partitions, two 8×16 partitions or four 8×8 partitions. If the 8×8 mode is chosen, each of the four 8×8 sub-macroblocks within the macroblock may be split in a further 4 ways either as one 8 × 8 sub-macroblock partition, two 8 × 4 sub-macroblock

partitions, two 4×8 sub-macroblock partitions or four 4×4 sub-macroblock partitions. These partitions and sub-macroblock give rise to a large number of possible combinations within each macroblock.

This method of partitioning macroblocks into motion compensated sub-blocks of varying size is known as *tree structured motion compensation.*

A separate motion vector is required for each partition or sub-macroblock. Each motion vector must be coded and transmitted and the choice of partition(s) must be encoded in the compressed bitstream. Choosing a large partition size (16×16, 16×8, 8×16) means that a small number of bits are required to signal the choice of motion vector(s) and the type of partition but the motion compensated residual may contain a significant amount of energy in frame areas with high detail. Choosing a small partition size (8×4, 4×4, etc.) may give a lower-energy residual after motion compensation but requires a larger number of bits to signal the motion vectors and choice of partition(s). The choice of partition size therefore has a significant impact on compression performance. In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for detailed areas.

Motion Vectors

Each partition or sub-macroblock partition in an inter-coded macroblock is predicted from an area of the same size in a reference picture. The offset between the two areas (the motion vector) has quarter-sample resolution for the luma component and one-eighth-sample resolution for the chroma components. The luma and chroma samples at sub-sample positions do not exist in the reference picture and so it is necessary to create them using interpolation from nearby Coded samples.

Intra Prediction

In intra mode a prediction block P is formed based on previously encoded and reconstructed blocks and is subtracted from the current block prior to encoding. For the luma samples, P is formed for each 4 × 4 block or for a 16 × 16 macroblock. There are a total of nine optional prediction modes for each 4 × 4 luma block, four modes for a 16 × 16 luma block and four modes for the chroma components. The encoder typically selects the prediction mode for each block that minimizes the difference between P and the block to be encoded.

1 4 × 4 Luma Prediction Modes

Figure 6.24 shows a 4 × 4 luma block that is required to be predicted. The samples above and to the left have previously been encoded and reconstructed and are therefore available in the encoder and decoder to form a prediction reference. The samples a, b, c. . . p of the prediction block P are calculated based on the samples A–M as follows. Mode 2 (DC prediction) is modified depending on which samples A–M have previously been coded; each of the other modes may only be used if all of the required prediction samples are available.

Predicted luma frame formed using H.264 intra prediction

Luma 4x4 Intra Prediction Mode:

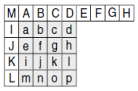


Figure 6.23 Labelling of prediction samples (4 × 4)

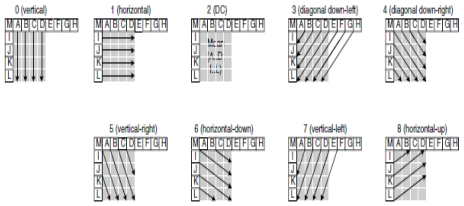


Figure 6.24 4 × 4 luma prediction modes

De-blocking Filter

A filter is applied to each decoded macro block to reduce blocking distortion. The de-blocking filter is applied after the inverse transform in the encoder (before reconstructing and storing the macro block for future predictions) and in the decoder (before reconstructing and displaying the macro block). The filter smoothes block edges, improving the appearance of decoded frames. The filtered image is used for motion-compensated prediction of future frames and this can improve compression performance because the filtered image is often a more faithful reproduction of the original frame than a blocky, unfiltered image3. The default operation of the filter is as follows; it is possible for the encoder to alter the filter strength or to disable the filter.

Filtering is applied to vertical or horizontal edges of 4×4 blocks in a macroblock (except for edges on slice boundaries), in the following order.

1. Filter 4 vertical boundaries of the luma component (in order a, b, c, d).
2. Filter 4 horizontal boundaries of the luma component (in order e, f, g, h).
3. Filter 2 vertical boundaries of each chroma component (i, j).
4. Filter 2 horizontal boundaries of each chroma component (k, l).

Transform and Quantization

H.264 uses three transforms depending on the type of residual data that is to be coded: a Hadamard transform for the 4×4 array of coefficients in intra macroblocks predicted in 16×16 mode, a Hadamard transform for the 2 × 2 array of chroma DC coefficients (in any macro block) and a DCT-based transform for all other 4 × 4 blocks in the residual data.

Data within a macro block are transmitted in the order if the macro block is coded in 16×16 Intra mode, then the block labeled '−1', containing the transformed DC coefficient of each 4 × 4 luma block, is transmitted first. Next, the luma residual blocks 0–15 are transmitted in the order shown (the DC coefficients in a macro block Coded in 16×16 Intra modes are not sent). Blocks 16 and 17 are sent, containing a 2×2 array of DC coefficients from the Cb and Cr chroma components respectively and finally, chroma residual blocks 18–25 (without DC coefficients) are sent.

4 × 4 Residual Transform and Quantization (blocks 0–15, 18–25)

This transform operates on 4 × 4 blocks of residual data after motion-compensated prediction or Intra prediction. The H.264 transform is based on the DCT but with some fundamental differences:

1. It is an integer transform (all operations can be carried out using integer arithmetic, without loss of decoding accuracy).
2. It is possible to ensure zero mismatches between encoder and decoder inverse transforms (Using integer arithmetic).
3. The core part of the transform can be implemented using only additions and shifts.
4. A scaling multiplication (part of the transform) is integrated into the quantize, reducing the total number of multiplications.

The inverse quantization (scaling) and inverse transform operations can be carried out using 16-bit integer arithmetic (footnote: except in the case of certain anomalous residual data patterns) with only a single multiply per coefficient, without any loss of accuracy.

Reordering

In the encoder, each 4×4 block of quantized transform coefficients is mapped to a 16-element array in a zig-zag order. In a macro block encoded in 16 × 16 Intra mode, the DC coefficients (top-left) of each 4 × 4 luminance block are scanned first and these DC coefficients form a 4 × 4 array that is scanned in the order. This leaves 15 AC coefficients in each luma block that are scanned starting from the 2nd position. Similarly, the 2 × 2 DC coefficients of each chroma component are first scanned (in raster order) and then the 15 AC coefficients in each chroma 4 × 4 block are scanned starting from the 2nd position.

Entropy Coding

Above the slice layer, syntax elements are encoded as fixed- or variable-length binary codes.
At the slice layer and below, elements are coded using either variable-length codes (VLCs) or context-adaptive arithmetic coding (CABAC) depending on the entropy encoding mode.
When entropy coding mode is set to 0, residual block data is coded using a context-adaptive variable length coding (CAVLC) scheme and other variable-length coded units are coded using Exp-Golomb codes.

THE MAIN PROFILE

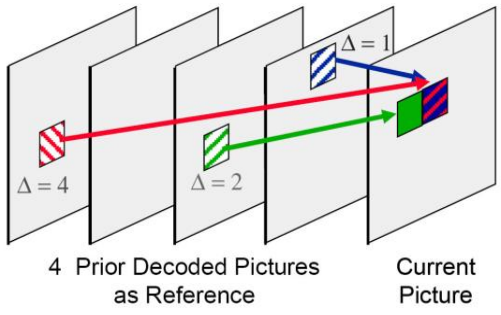
Suitable application for the Main Profile include (but are not limited to) broadcast media applications such as digital television and stored digital video. The Main Profile is almost a superset of the Baseline Profile, except that multiple slice groups, ASO and redundant slices (all included in the Baseline Profile) are not supported. The additional tools provided by Main
Profile are B slices (bi-predicted slices for greater coding efficiency), weighted prediction (Providing increased flexibility in creating a motion-compensated prediction block), support for interlaced video (coding of fields as well as frames) and CABAC (an alternative entropy Coding method based on Arithmetic Coding).

B slices

Each macro block partition in an inter coded macro block in a B slice may be predicted from one or two reference pictures, before or after the current picture in temporal order. Depending on the reference pictures stored in the encoder and decoder gives many options for choosing the prediction references for macro block partitions in a B macro block type.

Reference pictures

B slices use two lists of previously-coded reference pictures, list 0 and list 1, containing short term and long term pictures. These two lists can each contain past and/or future coded pictures (pictures before or after the current picture in display order). The long term pictures in each list behaves in a similar way to the description. The short term pictures may be past and/or future coded pictures and the default index order of these pictures is as follows:
List 0: The closest past picture (based on picture order count) is assigned index 0, followed by any other past pictures (increasing in picture order count), followed by any future pictures (in increasing picture order count from the current picture).
List 1: The closest future picture is assigned index 0, followed by any other future picture (in Increasing picture order count), followed by any past picture (in increasing picture order count).



EXTENDED PROFILE

The Extended Profile (known as the X Profile in earlier versions of the draft H.264 standard) may be particularly useful for applications such as video streaming. It includes all of the features of the Baseline Profile (i.e. it is a superset of the Baseline Profile, unlike Main Profile), together with B-slices, Weighted Prediction and additional
features to support efficient streaming over networks such as the Internet. SP and SI slices facilitate switching between different coded streams and 'VCR-like' functionality and Data Partitioned slices can provide improved performance in error-prone transmission environments.

SP and SI slices

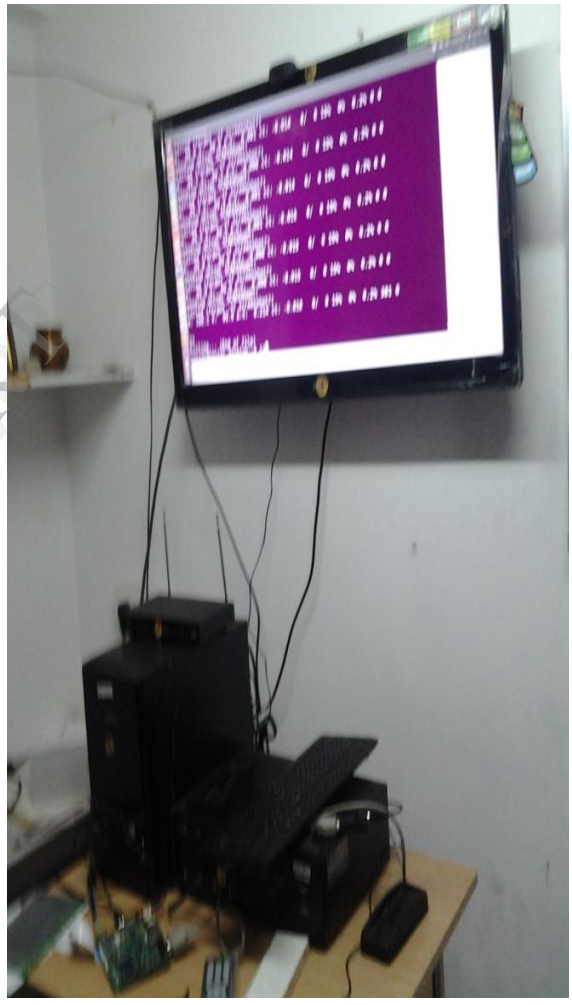
SP and SI slices are specially-coded slices that enable (among other things) efficient switching between video streams and efficient random access for video decoders .A common
requirement in a streaming application is for a video decoder to switch between one of several encoded streams. For example, the same video material is coded at multiple bitrates for
transmission across the Internet and a decoder attempts to decode the highest-bit rate stream
it can receive but may require switching automatically to a lower-bit rate stream if the data
throughput drops.

CONCLUSION

H.264 provides mechanisms for coding video that are optimized for compression efficiency
and aim to meet the needs of practical multimedia communication applications. The range of available coding tools is more restricted than MPEG-4 Visual (due to the narrower focus of H.264) but there are still many possible choices of coding parameters and strategies. The success of a practical implementation of H.264 (or MPEG-4 Visual) depends on careful design of the CODEC and effective choices of coding parameters. The next chapter examines design issues for each of the main functional blocks of a videoCODECand compares the performance of MPEG-4 Visual and H.264.

Thus to implement a H264 codec in C and design a player such that it should play a HD video either on TV or monitor using low cost and portable device has been achieved.

OUTPUT



REFERENCES

[1] D. Finchelstein, et al., "A Low-Power 0.7-V H.264 720p Video Decoder",
Asian Solid-State Circuits Conference, pp. 173-176, April, 2008.
[2] Y. Lin, et al., "A 242mW 10mm2 1080P H.264/AVC High-Profile Encoder
Chip", ISSCC Dig. Tech. Papers, pp. 314-315, Feb. 2008.
[3] Y. Kikuchi, "A 222mW H.264 Full-HD Decoding Application Processor with
x512b Stacked DRAM in 40nm", ISSCC Dig. Tech Papers, pp. 326-327, Feb.
2010.
[4] K. Iwata, et al., "A 342mW Mobile Application Processor with Full-HD Multi-
Standard Video Codec", ISSCC Dig. Tech. Papers, pp. 158-159, Feb. 2009.
[5] S. Nomura, et al., "A 9.7mW AAC-Decoding, 620mW H.264 720p 60fps
Decoding, 8-Core Media Processor with Embedded Forward-Body-Biasing and
Power-Gating Circuit in 65nm CMOS Technology", ISSCC Dig. Tech Papers, pp.
262-263, Feb. 2008.
[6] G. Gammie, et al., "A 45-nm 3.5G Baseband and Multimedia Application
Processor using Adaptive Body-Bias and Ultra-Low Power Techniques", ISSCC
Dig. Tech. Papers, pp. 258-259, Feb. 2008.

B.SUSMA received my B-tech degree in Electronics& Communication Engineering from JNTU. Hyderabad in 2011. Currently an M-tech student in the Department of Embedded System at JNTU. Hyderabad.

B.SRINIVAS Received his Btech and M-tech degree in Electronic & and Communication Engineering from JNTU. Hyderabad. Currently working as a Asst. Professor
in Aurora"s Technological & Research Institute under JNTU. Hyderabad.