International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 1, No. 3, November 2012, pp. 95~102
ISSN: 2089-4864

VHDL Implementation of H.264 Video Coding Standard

Jignesh Patel*, Haresh Suthar, Jagrut Gadit*****
* Departement of Electronics & Commmunications Engineering, Parul institute of Engineering & Technology.
** Departement of Electronics & Commmunications Engineering, Parul institute Technology.
*** Departement of Electrical Engineering, M.S.University of Baroda.

Article Info

Article history:

Received Apr 17, 2012
Revised Aug 19, 2012
Accepted Sep 21, 2012

Keyword:

Intra prediction
Transform
Quantization
CAVLC

ABSTRACT

This Paper contains VHDL implementation of H.264 video coding standard, which is new video coding standard of the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group. The main goal of the H.264/AVC standardization effort is to enhance compression performance and provision of a “network-friendly” video representation addressing “conversational” (video telephony) and “no conversational” (storage, broadcast, or streaming) applications.H.264 video coder standard is having fundamental blocks like transform and quantization, Intra prediction, Inter prediction and Context Adaptive Variable Length Coding (CAVLC). Each block is designed and integrated to one top module in VHDL.

Corresponding Author:

Haresh Suthar,
Departement of Electronics & Commmunications Engineering,
Parul institute Technology,
PO - Lomda, Ta-Waghodia, Vadodara, India.
Email: hareshsuthar@rediffmail.com

1. INTRODUCTION

H.264 video coding standard has the same basic functional elements as previous standards (MPEG-1, MPEG-2, MPEG-4 part 2, H.261, and H.263), like, (1) transform for reduction of spatial, (2) quantization for bit rate control, (3) motion compensated prediction for reduction of temporal correlation and (4) entropy encoding for reduction of statistical correlation. Inordered to improve coding performance, the following important changes in H.264 standard occur by (1) including intra-picture prediction, (2) a new 4x4 integer transform, (3) multiple reference pictures, (4) variable block sizes, (5) a de-blocking filter, and (6) improved entropy coding [2].

Improved coding efficiency comes at the expense of added complexity to the coder/decoder. H.264 utilizes different methods to reduce the implementation complexity like, Multiplier-free integer transform is introduced. Multiplication operation for the exact transform is combined with the multiplication of quantization to reduce the quantization step size to improve PSNR to levels that can be considered visually lossless.

The noisy channel conditions like the wireless networks obstruct the perfect reception of coded video bit stream in the decoder. Incorrect decoding by the lost data degrades the subjective picture quality and propagates to the subsequent blocks or pictures. H.264 utilizes different methods to exploit error resilience to network noise. The parameter setting, flexible macroblock ordering, switched slice, redundant slice methods are added to the data partitioning, used in previous standards. Depending upon applications, H.264 defines the Profiles and Levels specifying restrictions on bit streams like some of the previous video standards. Seven Profiles are defined to cover the various applications from the wireless networks to digital cinema.In this paper, presented the VHDL design for the H.264 Standards. Algorithm is designed in Xilinx for H.264 and tested using TEXTIO Package of FPGA.

The paper is organized as follows. Section 2 provides overview of H.264 standards. Section 3 describes algorithm and proposed design for VHDL implementation. Section 4 is Results and Section 5 is Conclusion.

2. OVERVIEW OF H.264 STANDARD

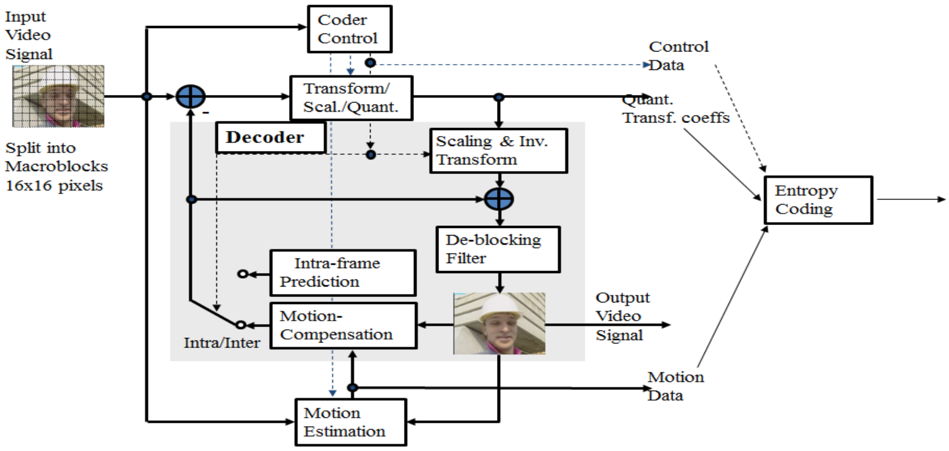

Figure 1. H.264 Block Diagram

Figure 1 shows the basic block diagram of H.264, which contains blocks of transform, quantization, Intra prediction, Inter prediction, and CAVLC. Source pictures and prediction residuals both have high spatial redundancies. H.264 Standard is based on the use of a block-based transform for spatial redundancy removal. H.264 uses an adaptive transform block size, 4x4 and 8x8 (High Profiles only), whereas previous video coding standards were using the 8x8 Discrete Cosine Transform (DCT). The smaller block size leads to a significant reduction in ringing artifacts. The 4x4 transform has the additional benefit of removing the need for multiplications. Transform convert spatial domain to frequency domain [4].

The H.264 extends the quantization step sizes QP by two additional octaves, by redefining the values (which contain value of M_f and V_i) and allowing QP to vary from 0 to 51. In general, transform and quantization require several multiplications resulting in high complexity for implementation. So, for simple implementation, the exact transform process is modified to avoid the multiplications. Then the transform and quantization are combined by the modified integer forward transform, quantization and scaling [4].

The complete forward transform, scaling and quantization process becomes:

$$Y = round\left(\left([C_f].[X].[C_f^T] \cdot M_f / 2^{floor(QP/6)}\right) \cdot {}^{1}/{}_{2^{15}}\right) \quad (1)$$

Where: $$C_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

The complete inverse transforms and scaling process becomes:

$$Z = round\left([C_i^T] \cdot \left[Y \cdot V_i \cdot 2^{floor\left({}^{QP}/{}_{6}\right)}\right] \cdot [C_i] \cdot \frac{1}{2^6}\right) \quad (2)$$

Where: $$C_i = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}$$

Intra prediction predicts blocks form leftmost and top blocks of same frame. If a block or macro block is encoded in intra mode, a prediction block is formed based on previously encoded and reconstructed blocks. There are total of 9 optional prediction modes for each 4x4 luma blocks shown in Figure 2. Same as 4 optional modes for a 16x16 luma block and 8x8 chromablocks [5].

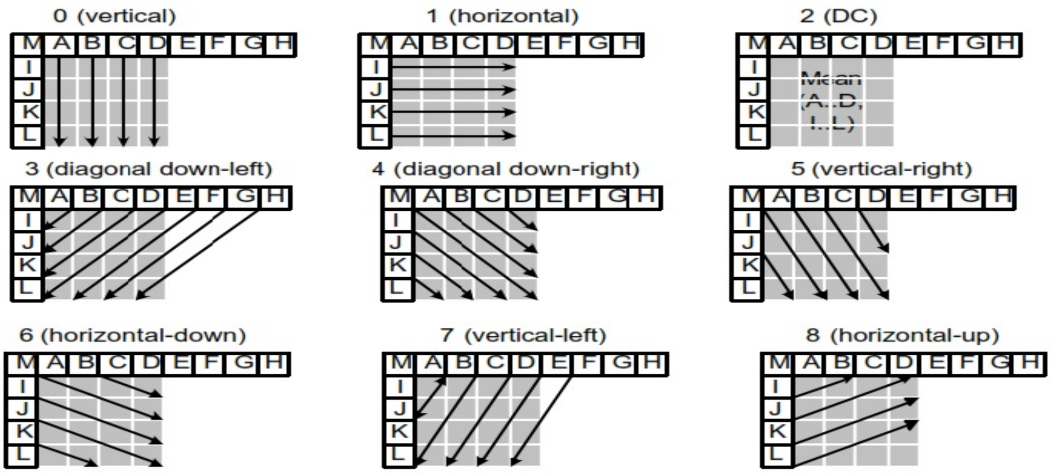

Figure 1. 4 × 4 Intra Prediction Modes

In successive frames of video, most of the information is same. So, Intra prediction predicts the current frame from the previous frame and finds motion vectors. Intra prediction uses motion estimation and motion compensation [6].

A filter can be applied to every decoded macro block in order 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 has two benefits: (1) block edges are smoothed, improving the appearance of decoded images (particularly at higher compression ratios) and (2) the filtered macro block is used for motion-compensated prediction of further frames in the encoder, resulting in a smaller residual after prediction [7].Filtering is applied to vertical or horizontal edges of 4x4 blocks in a macroblock, in the following order as shown in Figure 3:

1. Four filters for vertical boundaries of the luma component.
2. Four filters for horizontal boundaries of the luma component.
3. Two filters for vertical boundaries of each chroma component.
4. Two filters horizontal boundaries of each chroma component.

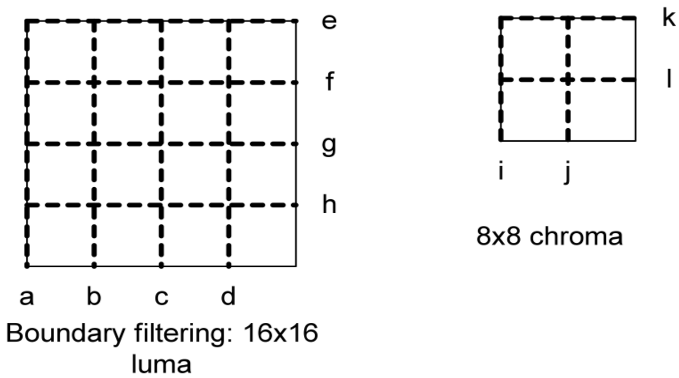

Figure 2. Edge Filtering Order in a Macroblock

H.264 uses entropy coding for matching a symbol to a code, based on the context characteristics. All syntax elements, except for the residual data are encoded by the Exp-Golomb codes. To read the residual data (quantized transform coefficients), zigzag 1scan (interlaced) or alternate scan (no interlaced or field) is used. For coding the residual data, a more sophisticated method called Context-Based Adaptive Variable Length Coding (CAVLC) is employed. Context-based Adaptive Binary Arithmetic Coding (CABAC) is also employed in Main and High Profiles, CABAC has more coding efficiency but higher complexity compared to Context-Based Adaptive Variable Length Coding (CAVLC). A coded H.264 stream or an H.264 file consists of a series of coded symbols. These symbols make up the syntax and include parameters, identifiers, delimiting codes, prediction types, differentially coded motion vectors and transform coefficients [8] [9].

3. DESIGN AND DEVELOPMENT OF H.264 STANDARD

The basic block diagram of Figure 4 consist integration of H.264. First of all, YCbCr image of pixel value need to give the RAM. RAM stores the data and transmits as compatible to the top module.Prediction stage fetches the data from the RAM in the form of pixel value. Two types of prediction are done in that stage I & P prediction. I slice (Intra-coded slice) is the coded slice by using prediction only from decoded samples within the same slice. P slice (Predictive-coded slice) is the coded slice by using inter-prediction from previously decoded reference pictures, using at most one motion vector and reference index to predict the sample values of each block. The prediction requires previous data which is taken from the reconstruction block. By using the previous data of prediction stage frame is predicted. Prediction stage transmits DATA to transform and quantization stage.

Transform and quantization Stage convert spatial domain to frequency domain using 2d DCT transform. After that data is transmited to the Context-Based Adaptive Variable Length Coding (CAVLC) and then Inverse transform and De-quantization operation is performed. Inverse transform and De-Quantization block converts data again to spatial domain for the reconstruction block.

Reconstruction Block provides the Feedback for that it requires storing data in RAM which is used in future. This data is used in prediction stages. Headers store all header data and combine headers to byte stream for Decoding side.

Context-Based Adaptive Variable Length Coding (CAVLC) is doing entropy coding for transmit byte stream for the data transfer.

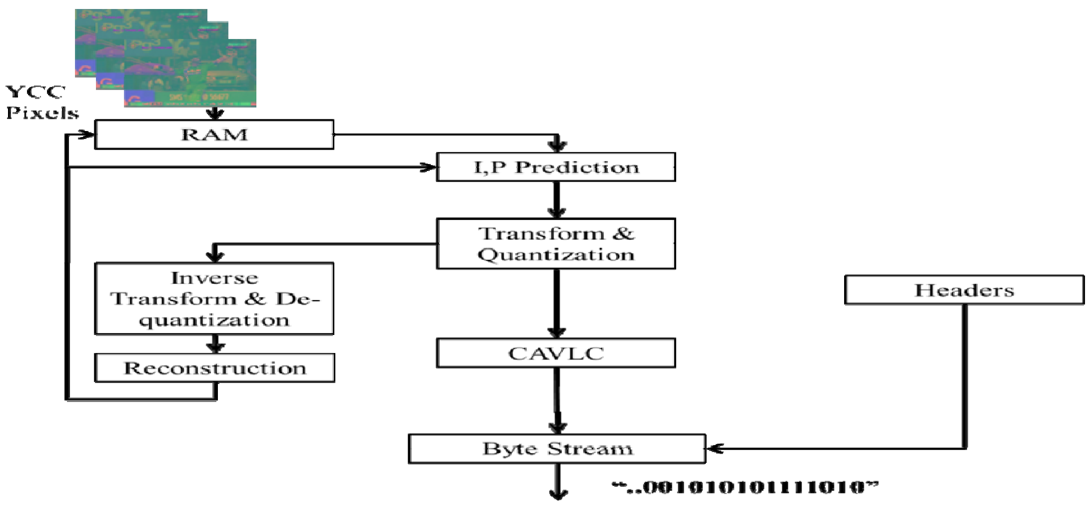

Figure 3. Basic Block Diagram For System Integration For H.264

4. SIMULATION RESULTS

The simulation result of Intra Prediction module in software tool ISIM 13.4 is shown in Figure 5. Before the first macroblock in a line NEWLINE is set this initializes all pointers and resets the prediction mode to no adjacent blocks. NEWLINE should be set for at least 1 CLK before STROBEI goes high. If this is the first in a slice, NEWSLICE should be set at the same time as NEWLINE. Data is clocked in by STROBEI and appropriate mode is selected and generated output DATAO.

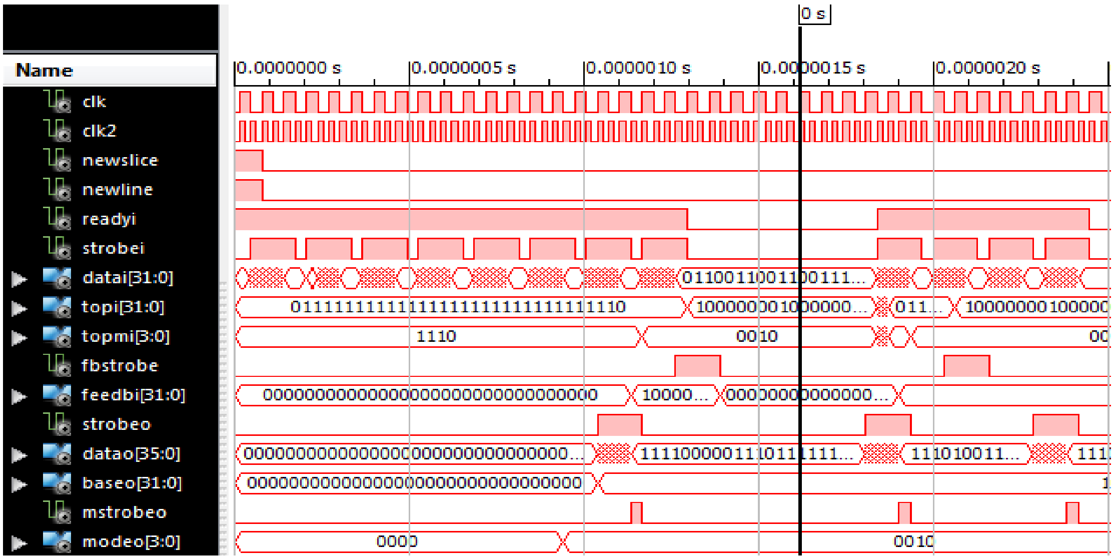

Figure 4. Simulation Result of Intra Prediction in ISIM 13.4

Figure 6 shows the simulation result for the Transform, Quantization, Dequantization and Inv-Transform. Input pixel value is represented in 8 bit and reconstructed pixel value represented in 10 bit because of extra carry bits.

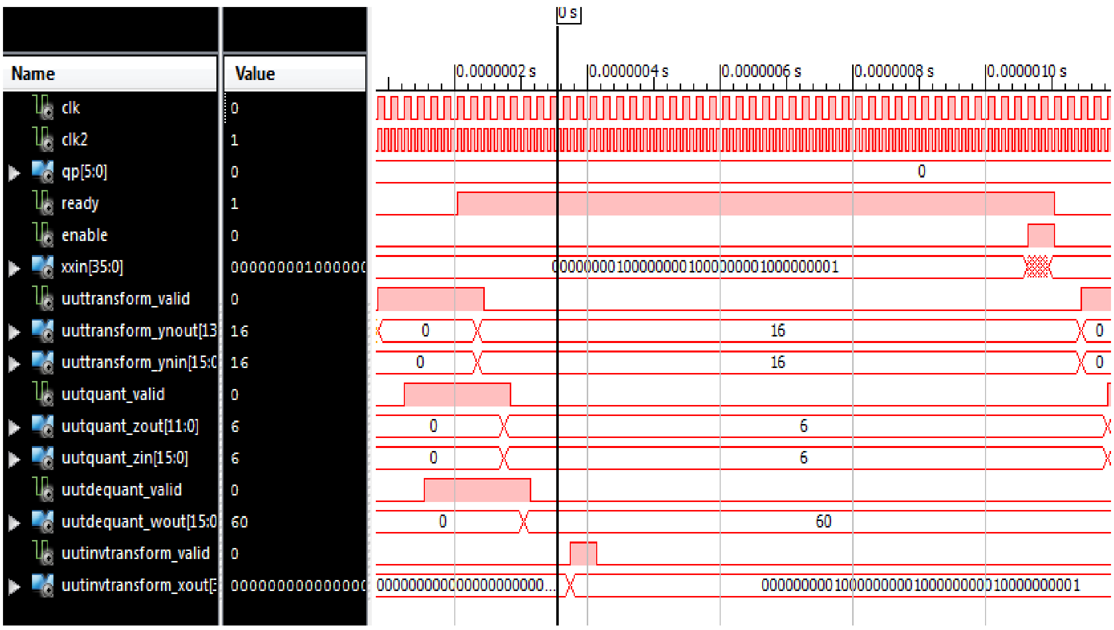

Figure 5. Simulation Result of Transform, Quantization, Dequantization and Inverse Transforming ISIM 13.4

Simulation result for Context-Based Adaptive Variable Length Coding (CAVLC) shown in Figure 7 All modules are integrated in TOP module and simulation result of TOP module is shown in Figure 8.

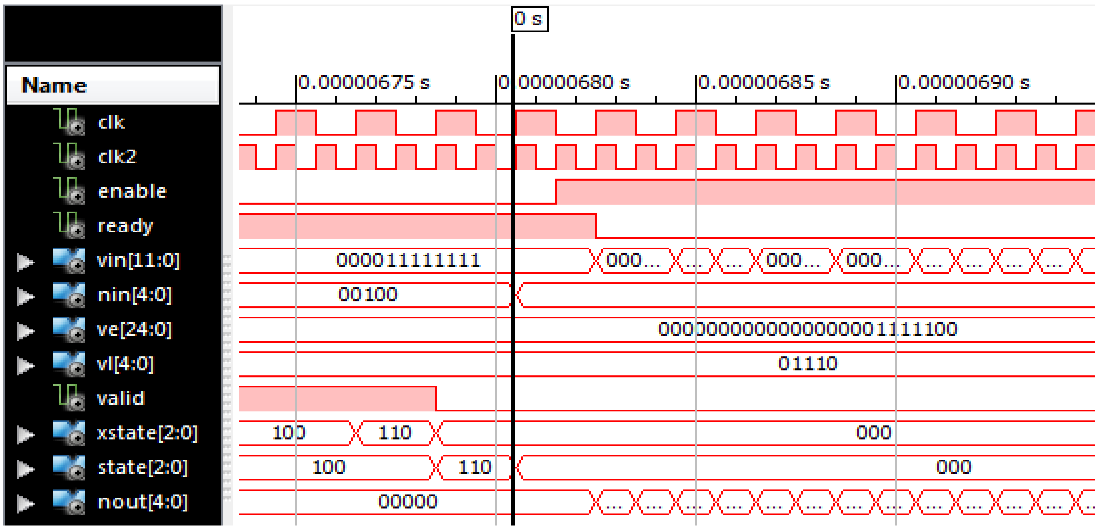

Figure 6. Simulation Result of CAVLC in ISIM 13.4

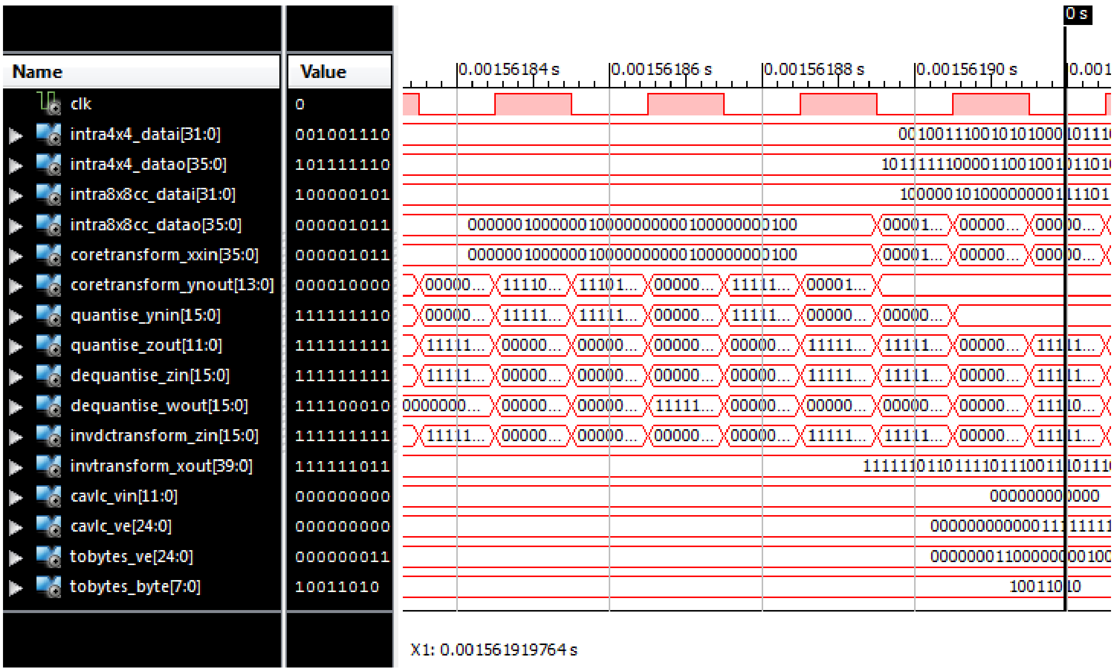

Figure 7. Simulation result of TOP module in ISIM 13.4

YUV 4:2:0 Video formats is used for the input of the whole design. TEXTIO package defines input byte file which can be read in the format of character. First of all Y frame is read for frame size and then U, V frame is read for half of frame size. Frame is divided in macroblocks and sub macroblocks. Design algorithm takes input from the input macroblock and applies to whole design of H.264 standard. Output byte stream is converted to character to extract for output file.

The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE gives the lower the error. The higher the PSNR better the quality of the compression.

To compute the PSNR, of the block first calculate the mean-squared error using the following equation:

$$MSE = \frac{\sum_{M,N}[I_1(M,N) - I_2(M,N)]^2}{M \times N} \tag{3}$$

In the Equation 3, M and N are the number of rows and columns in the input images, respectively. Then the block computes the PSNR using the following equation:

$$PSNR = 10\log_{10}\left(\frac{R^2}{MSE}\right) \tag{4}$$

In the Equation 4, R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc. For the YUV video sequences City is taken as test input sequence and using TEXTIO pixel value is extracted from the frame of the sequence. One frame of sequence is used to measure PSNR as Equation 4. PSNR-Y is calculated by comparing with original image Y frame component to compressed video Y frame for each component by developed algorithm for calculation.In this PSNR-Y of output sequence shown in Figure 9.

Original Sequence

City sequence

Output Sequence

PSNR-Y = 34.7 dB

Figure 8. Result of Video Sequence

5. CONCLUSION

This paper describesVHDL implementation of H.264 video coding standard and for that each module of the H.264 standard is designed and tested individually. Ultimately all design is integrated in one design to achieve the standard.

Design is developed for Transform and Quantization to scaling the input video and converted for generating byte stream of input video. Prediction stage is designed for the same frame and consecutive frames. CAVLC variable coding is designed for entropy coding. As per H.264 Syntax byte stream is generated.

TEXTIO package of VHDL is used for testing design. YUV Video sequence is taken as input of Test Bench of design using TEXTIO, and H.264 codec output video is generated. Video quality is checked by measuring PSNR of o/p video with original which is 34.7dB.

REFERENCES

[1]. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC”, 2003.

[2]. Thomas Wiegand, Gary J. Sullivan, "Overview of the H.264/AVC Video Coding Standard", IEEE transactions on circuits and systems for video technology, vol. 13, no. 7, July 2003.
[3]. Soon-kak Kwon a, A. Tamhankar, K.R. Rao, "Overview of H.264/MPEG-4 part 10", J. Vis. Commun. Image R. 17, 2006, 186–216.
[4]. Iain Richardson "White Paper: 4x4 Transform and Quantization in H.264/AVC", Vcodex © 2002-2011.
[5]. Iain Richardson "White Paper: H.264 / AVC Intra Prediction" ,Vcodex © 2002-2011.
[6]. Iain Richardson "White Paper: H.264 / AVC Inter Prediction" Vcodex © 2002-2011
[7]. Iain Richardson "White Paper: H.264 / AVC loop filter" Vcodex © 2002-2011
[8]. Iain Richardson "White Paper: H.264 / AVC CAVLC" Vcodex © 2002-2011
[9]. Iain Richardson "White Paper: H.264 / AVC CABAC" Vcodex © 2002-2011
[10]. R. Schafer, T. Wiegand, and H. Schwarz, "The Emerging H.264 AVC Standard", EBU Technical Review, January 2003.

BIOGRAPHIES OF AUTHORS

Jignesh R. Patel
He is purshing his Master in Electronics & Communication Enggineering at Parul Institute of Engineering & Technolohy,Vadodara.

Haresh A.Suthar
He is Reader and Head of the E&C Engineering Department at Parul Institute of Technolgy,Vadodara.He got his BE-Electronics and Master in Autmatic Control & Robotics from M.S.U of Baroda. His area of interest is embedded, control systems and Artificial Intelligence.

Dr. Jagrut J. Gadit
He is Reader in Electrical Engineering Department at Faculty of Technology & Engineering, M.S.U of Baroda.He received his BE-Electrical and Master in Automatic Control and Robotics.He is Phd in the area of control system.