IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 6, Ver. II (Nov. - Dec. 2016), PP 29-40
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Reliable Error free Coding for Resource Constraint Soc Design

Fairooz SK[1], Dr.Madhavi B.K[2]

[1] *Fairooz.SK, Research Scholar, JNTU, Hyderabad, India.*

[2] *Dr.Madhavi.B.K, Professor in ECE, SWEC, Hyderabad, India.*

Abstract: *With the extent of video coding applications in various domains, the need for faster and reliable modelling of video codec's is in raise. The conventional modelling of video codec's were developed for high resource compatibility. As the applications of video coding has moved to low resource devices, a need for the proposal of new coding, streaming and buffering logic in resource constraint environment is required. In this paper an approach for the modelling of video coding process using language of hardware description. The operational functionality of the proposed system is validated over Xilinx targeted FPGA device. A comparative analysis of the developed approach for 2 different search approaches for motion estimation is evaluated over the developed system. The comparative result observations were observed to be improved in estimation accuracy and minimization in processing time.*

Keywords: *Video codec; coding; steaming; buffering unit; codec modelling.*

I. Introduction

Video processors are developing for various needs of applications. Video has become an essential part for multimedia applications. There are, however, many contradicting limitations in video codec implementations. One challenge is the rapid evolution of compression standards with several different algorithms [1]. This requires programmability that is easy to achieve with processor-based platforms. However, achieving the best power, energy, and silicon area efficiency requires custom hardware implementations [14]. On the other hand, hardware (HW) design is more demanding than software (SW) development, and modifications are very expensive and time consuming [8]-[13]. For example, nonrecurring engineering costs, increase rapidly with each technology generation making frequent HW upgrades less favourable. Software only implementation solves the flexibility and upgradeability problem but is not an optimal solution from a performance versus real time need point of view [15],]16]. In video processing the modelling of SoC unit for multi rate service compatibility is still an open challenge. Various approaches were made in regard to obtain scalable coding approach in video coding [2]-[7]. However in such coding approach the processing error is not concentrated. To improvise the visual quality with demanded scalable rate retaining error limit in this paper a scalable error coding approach is proposed. A digital modelling for coding at different level of service is designed to provide maximum visualization quality to decoded video. To obtain the objective of reliable video coding with variable resource feasibility a new coding structure for video coding in SoC model is presented. To outline the suggested approach in this paper a video coding system designed for modelling of SoC is outlined in section 2. Section 3 outlines the proposed scalable coding approach for data streaming. For the presentation of image projection to higher grid interpolation a resolution projection approach is outlined in section 4.The experimental result for the developed system is outlined in section 5. Section 6 presents the conclusion made for the developed approach.

II. Video Coding: System Outline

For the designing modelling of proposed approach in a generalized system model is developed. The proposed design architecture for the suggested unit is as shown in below figure.

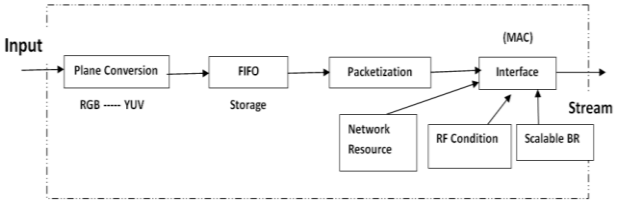

Figure 1: System Design for coding process in proposed architecture

The system reads the input data from the source file. The designed file consists of coefficients of video pixels in discrete coefficient. The obtained coefficients are transformed to corresponding gray intensity pixel level and coded to binarized coefficient in 16-bit precision. The coefficients are stored in a FIFO device of size 64x16 for burst processing. The FIFO data are packetized based on the coding standard and passed to interfacing unit for bit allocation based on the demanded resource and available resource unit. The incorporation of the packet scheduling interface unit helps in providing scalable coding for multi-bit rate service. The processing unit for the video processing is referred from the standard MPEG structure and a simplified model is realized as shown below,

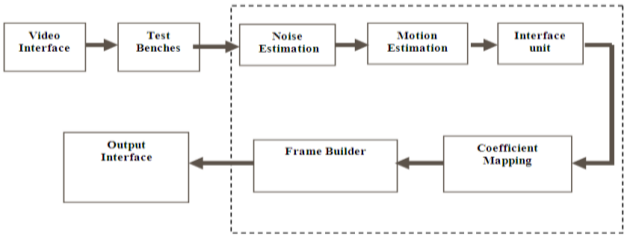

Figure 2: Digital modelling for coding structure of video processing

The designed system reads the video sequence and interface to the developed digital system via the test interface unit called test bench. The video sequences are read using the analog interface unit and is designed in Matlab tool. The developed function reads the passed video file and transform into successive frame sequence. These frame sequence are read and each coefficient of the read frame are transformed from its gray intensity to equivalent binary sequence. These binary sequences are passed as a input to the designed HDL unit via test bench declaration. The two-frame sequence are loaded as a test vector in this test bench and passed to the HDL for further computation. The first interfacing unit in the Designed Digital system is the noise estimation unit, which is used for the elimination of additional noise coefficients added due to transformation or transmission. The noise estimation unit computes the mean error difference between the two mapping blocks from frame 1 and frame 2. The two blocks are successively compared for distance between the two corresponding pixels and accumulated for the mean error calculation. The error values are calculated for all the blocks in a sequence and the least error value is computed for these obtained error values. This error value is taken as the threshold least mean error value for the noise elimination in motion estimation block. The Least mean error value for the estimation is given by,

$LMSE_{er} = [mean(least\ error)^2 + 3 * \sigma(least\ error)]$ (1)

For the estimation of motion vector BMA algorithm is realized in full search mode. The operation flow chart for this algorithm is as outlined below,

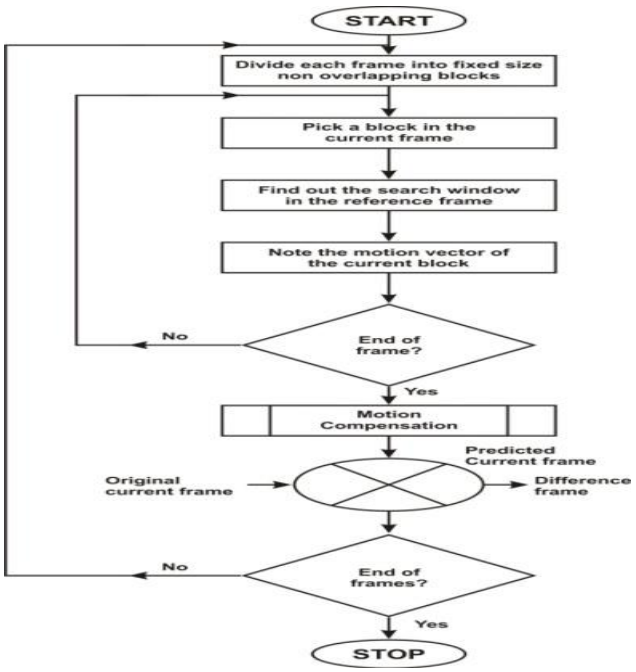

Figure 3(a): Flowchart for transmitter block

The following flowchart shows the order of activities carried out at the receiver block.

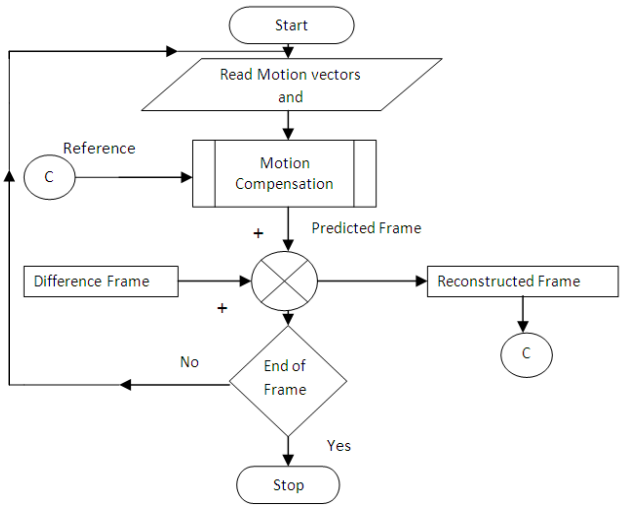

Figure 3 (b): Flowchart for Receiver block

For the realization of the suggested system the above stated algorithm is designed in digital operational units. The design consist of a comparator unit which generates the absolute difference of the two pixels and compares with the previous error value for the selectivity and accumulate the error value to the current error value in case of minimum of error evaluation. Figure below shows the digital unit realized for the evaluation of this operation.

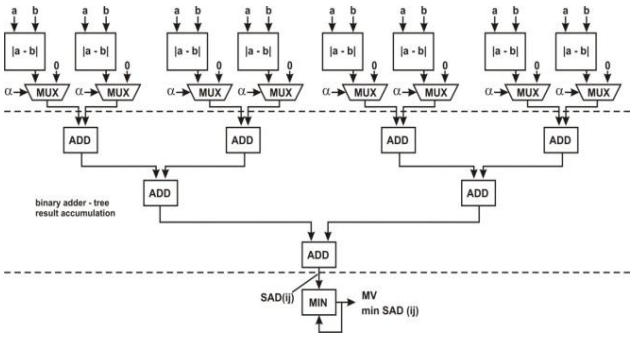

Figure 4: Digital model of matching algorithm for SAD calculation

The comparison operation is realized by the subtraction of the two input coefficients. The obtained difference value is added and this process is repeated for all the pixel values. On every SAD calculation the previous SAD is compared with the current SAD value for its comparison. From the obtained SAD value the motion vectors are predicted. The SAD is defined by,

$$SAD(d_1,d_2) = \sum_{(n_1,n_2)\in B} |S_1(n_1,n_2,k) - S_2(n_1+d_1,n_2+d_2),k+1| \quad (2)$$

Where s_1 (n_1, n_2, k) is the pixel value at (n_1, n_2) in frame k and s_2 (n_1+d_1, n_2+d_2, k+1) is the pixel value at (n_1+d_1, n_2+d_2) in frame k+1, d_1 and d_2 are the horizontal and vertical motion vectors respectively. Once the SAD value for a particular frame is calculated the motion vector (d_1, d_2) is calculated given by,

$$[d_1 d_2]^T = arg\ \min_{d_1 d_2} SAD(d_1,d_2) \quad (3)$$

The developed BMA architecture is interfaced with a general video interface unit providing error correction coding and entropy encoding for its evaluation. The obtained motion vector is passed to entropy encoder and the compressed data is processed with a CRC unit for error coding in the transmitting bit.

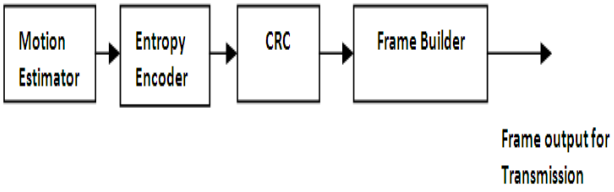

Figure 5: Video codec interface with motion estimator unit

Once the compressed bit sequence is obtained and the FCS bits are generated the data is aligned in a frame format before transmission. A general format of the data transmission is developed for the transmission of this data.

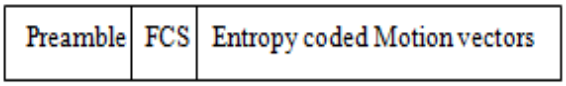

Figure 6: Frame format for developed codec

The frame is developed with the following fields of entry, for the generated frame sequence resource allocation is performed. The designing of allocation logic is developed based scheduler logic under multiple level service compatibility. The approach of scalability approach for developed system is outlined in following section.

III. Scalable Coding

To provide scalability under different demanded data rate a scheduler scheme is proposed. For the processing of data into end to end streaming a cross layer design is proposed. This system repetitively records the average bandwidth to the streaming server according to the residual BS capacity and RF condition on the period of Report Period. In each of the Report Period, the unit requests a target bit-rate from the coding unit. The data at lower spatial-SNR resolution are given more importance and in each spatial-SNR layer, the lower temporal layer is more important. Therefore, according to the requested bit-rate from the coding unit, the lower spatial-SNR layers are firstly extracted.

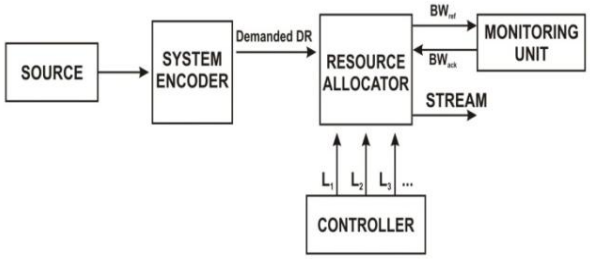

Figure 7: video streaming encoder on level selection

The coding structure for such a unit is as shown below to support the data rate at different demanded service rate the coding unit keep a track of the available bandwidth and allocates the resource based on request priority. This results in the coding of smoother data streaming under multiple data rate demand. To achieve the transmission rate faster and retraining the visual quality a projection modelling using linear interpolation is proposed.

IV. Visual Projection Coding

The projection approach developed codes the resolution under horizontal and vertical grid projection to project the sample the developed projection algorithm is as outlined below. The Pv and PH values defined in equation below.

$P_V = [Q_V(X_V/L_V - Z_V) + \epsilon]$ (4)
$P_H = [Q_H(X_H/L_H - Z_H) + \epsilon]$ (5)
Where,
P_V =Vertical phase.
P_H = Horizontal phase.
X_V = Vertical coordinates of high resolution pixel.
X_H = Horizontal coordinates of high resolution pixel.
Є = Very small number like 10-6.
Z_V = Vertical coordinates of low resolution pixel.
Z_H = Horizontal coordinates of low resolution pixel.
For scaling ratios less than two, selects the pixels to be omitted using equation shown below.
$Z_V = [X_V/L_V]$ $Z_H = [X_H/L_H]$ (6)
The processing data are passed via a input interface unit and the Input memory (IM) unit operates at input pixel clock frequency Fclk_in. The block consists of four cache buffers, to provide a 5 x 1 pixel column to the CU.

The length of the line buffers is equal to the input video's horizontal size. Shown below is the block diagram of the Input Unit.

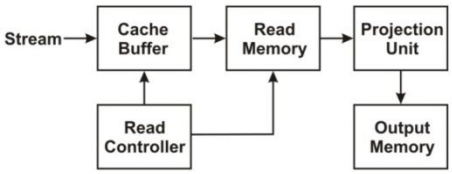

Figure 8: Input Unit

(a)Operation of Segregator

Segregator unit operates at core clock frequency CU, and generates four high resolution pixels using the selected Fclk_core. Figure below shows the architecture of the feature extraction and context classification units. Parallel implementation where Dr = 1, will use 15 adders, 32 multipliers, and a serial-parallel implementation, where Dr= 4 reduces the number of adders/subtractors to four and the multipliers to eight, with a negligible increase in the number of pipeline registers.

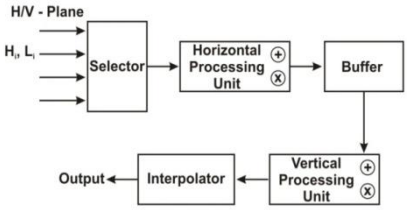

Figure 9: Context Classification and Feature Extraction.

(b) Operation of Interpolation

Interpolation unit (IN) unit operates at core clock frequency. 100 multiplications, and 96 additions required to perform 5 x 5 convolution for four HD pixels is resource shared with Dr= 4 to reduce the number of multipliers to 25, adders to 24. Therefore at each Fclk_core clock cycle, one convolution operation is performed, generating four HD pixels at every Fclk-in, clock cycle.

(c) Output Unit

Output memory (OM) unit also operates at core clock frequency. It basically chooses the appropriate pixels between the high resolutions pixels generated by IN unit, and arranges them in raster scan order according to the scaling ratio.

V. Experimental Results

For the functional evaluation of the designed system the developed system is designed using VHDL definition and simulated on the Active-HDL simulator for its operational verification. For the testing of the designed system a test bench file is generated where two frame data is passed. This frame data is read by the video_codec file and divide into 8 X 8 Block. The block data is processed for noise estimation, motion estimation, and compression. Under recovery the data is decoded back for its regeneration. The obtained simulation results are as illustrated below. For the real time Realization and resource utilization the developed design is synthesized using Xilinx synthesizer. The obtained synthesis results and their logical, mechanical, and timing reports are also presented in the following section.

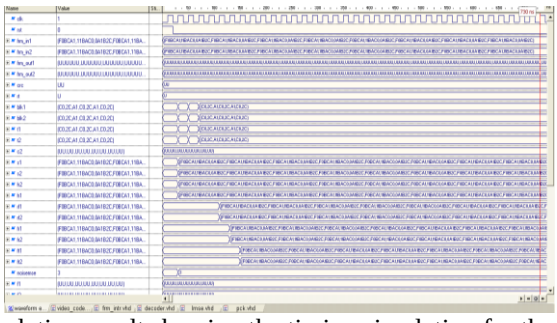

Figure 10: simulation result showing the timing simulation for the block division

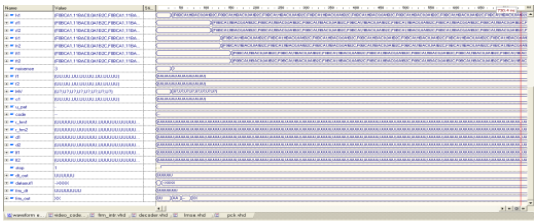

Figure 11: simulation result illustrating the moving vectors and noise estimation

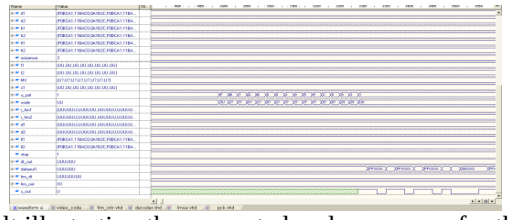

Figure 12: simulation result illustrating the generated code sequence for the coefficient compression

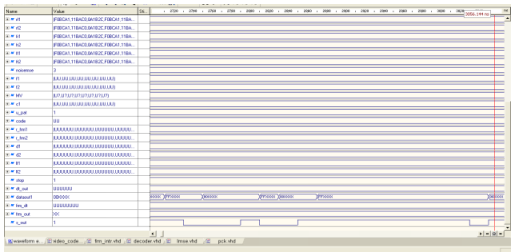

Figure 13: simulation result illustrating the generation of serial output information

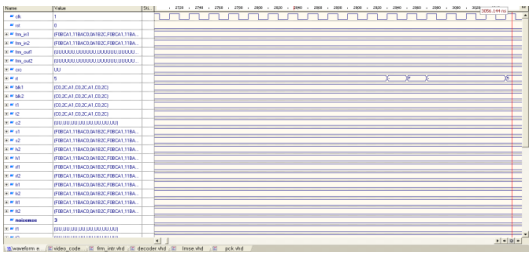

Figure 14: simulation resulting illustrating the generation of decoded information

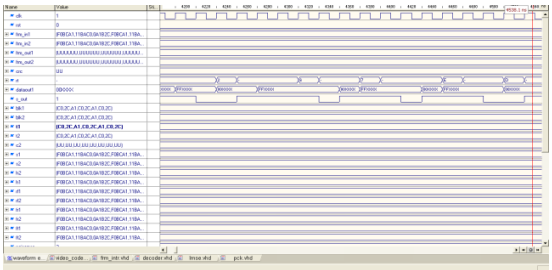

Figure 15: simulation results illustrating buffered code words received and their corresponding decoded word bits.

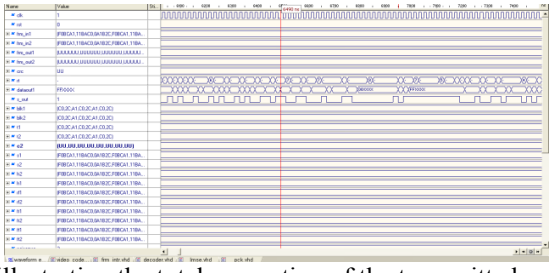

Figure 16: simulation result illustrating the total generation of the transmitted code sequence and it's decoded bit stream.

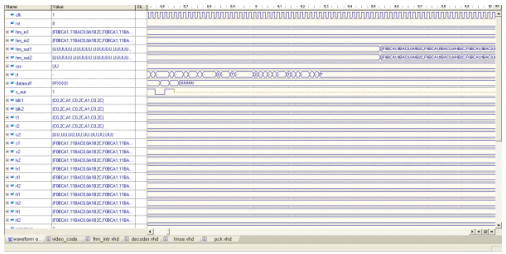

Figure 17: simulation result illustrating the generated frame value after decoding operation

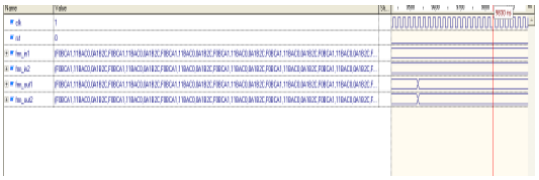

Figure 18: Illustrating the comparison of the transmitted and recovered frame sequence

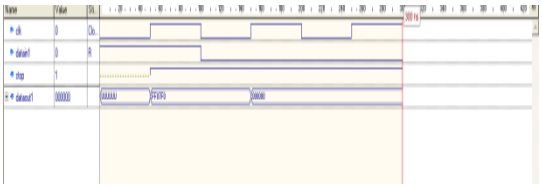

Figure 19: simulation result illustrating the CRC operation for the developed system

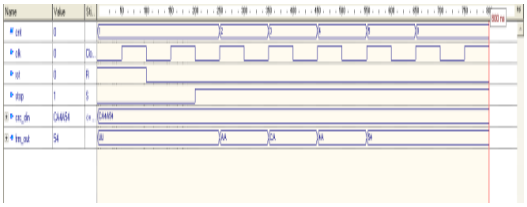

Figure 20: simulation of frame generation unit

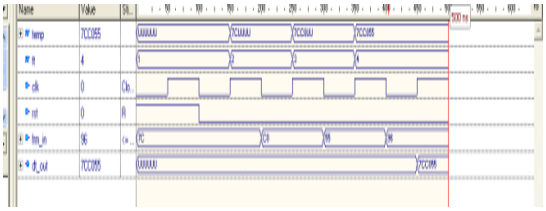

Figure 21: simulation result for frame reader unit.

Minimum period : 7.693ns
(Maximum Frequency: 129.984MHz)
Design Statistics
IOs : 34
Cell Usage:
BELS : 542

TRB Partition Summary

No partition information was found.

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	113	88,192	1%	
Number of 4 input LUTs	123	88,192	1%	
Logic Distribution				
Number of occupied Slices	134	44,096	1%	
Number of Slices containing only related logic	134	134	100%	
Number of Slices containing unrelated logic	0	134	0%	
Total Number of 4 input LUTs	261	88,192	1%	
Number used as logic	123			
Number used as a route-thru	138			
Number of bonded IOBs	34	1,164	2%	
IOB Flip Flops	14			
Number of PPC405s	0	2	0%	
Number of GCLKs	1	16	6%	
Number of GTs	0	20	0%	
Number of GT10s	0	0	0%	
Total equivalent gate count for design	2,633			
Additional JTAG gate count for IOBs	1,632			

Synthesis Report

Figure 22: summarized synthesis report for the developed desig

Release 9.1i - XPower SoftwareVersion:J.30
Copyright (c) 1995-2007 Xilinx, Inc. All rights reserved.
Design: C:\Xilinx91i\BMA\video_codec.ncd
Preferences: video_codec.pcf
Part: 2vpx70ff1704-6
Data version: PREVIEW,v1.0,05-28-03

Power summary:		I(mA)	P(mW)
Total estimated power consumption:			181
Vccint 1.50V	:	85	128
Vccaux 2.50V	:	20	50
Vcco25 2.50V	:	2	4
Clocks	:	0	0
Inputs	:	0	0
Logic	:	0	0
Outputs	:		
Vcc	:	25	0
Signals	:	0	0
Quiescent Vccint 1.50V	:	85	128
Quiescent Vccaux 2.50V	:	20	50
Quiescent Vcco25 2.50V	:	2	4

Thermal summary:

Estimated junction temperature	:	25C
Ambient temp	:	25C
Case temp	:	25C
Theta J-A	:	0C/W

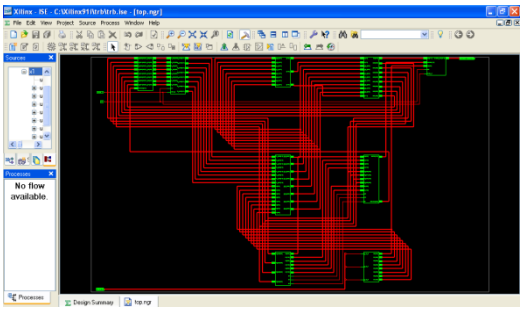

Figure 23: RTL view of the implemented system using Xilinx synthesizer.

The system is evaluated for (1) Peak Signal-to-Noise Ratio (PSNR) calculated between the predicted motion-compensated frame and the original frame (2) Mean Square Error (MSE) between the predicted frame and the original frame (3) Number of absolute operations per frame for calculating the matching error (4) Execution time according to our implementation and hardware environment.
The obtained observation for certain cases are illustrated below,

Case 1:
Name of the sample: Suzie.qcif
Number of frames : 150

Figure 24(a) A frame from Suzie.qcif

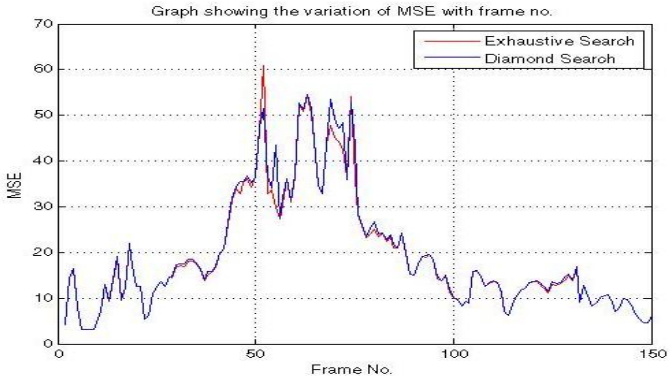

Figure 24(b) Frame No. Vs MSE

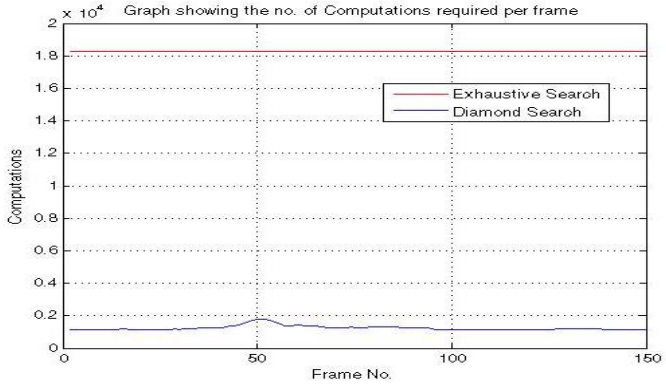

Figure 24(c) Frame No. Vs Computations

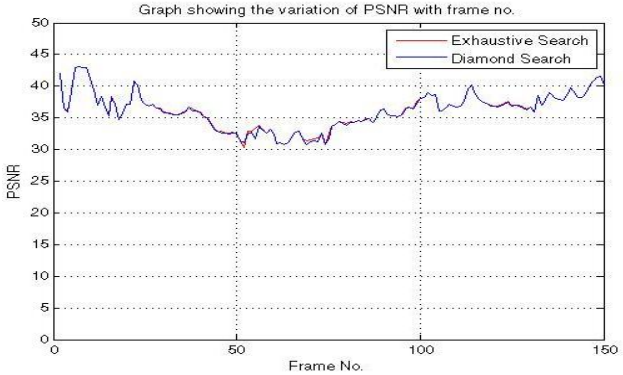

Figure 24(d) Frame No. Vs PSNR

Table 1: Performance comparison of FS and DS for suzie image sequence

Algorithm	MSE	PSNR (dB)	Computations		Execution time(sec)		
			number	%	T_1	T_2	$T=T_1+T_2$
FS	15.65	36.18	18271	100	162.13	0.42	162.55
DS	15.91	36.19	1229	6.72	15.11	0.45	15.56

Case 2:
Name of the sample : Table tennis.qcif
Number of frames : 150

Figure 25(a): A frame from tabletennis.qcif

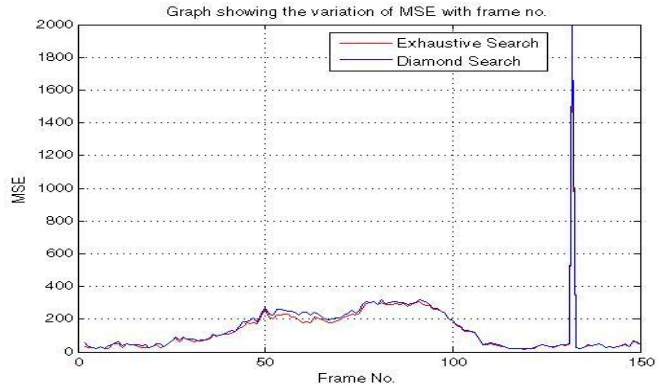

Figure 25(b): Frame No. Vs MSE

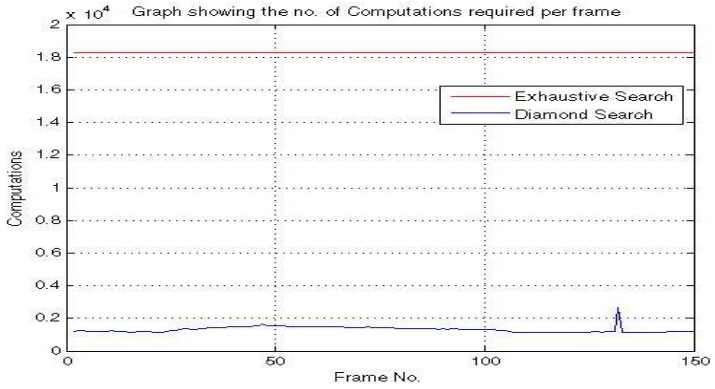

Figure 25(c): Frame No. Vs Computations

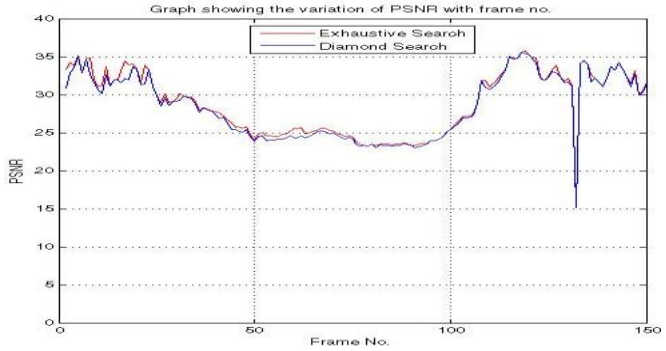

Figure 25(d): Frame No. Vs PSNR

Table 2: Performance comparison of FS and DS for table tennis image sequence

Algorithm	MSE	PSNR (dB)	Computations		Execution time(sec)		
			Number	%	T_1	T_2	$T=T_1+T_2$
FS	85	28.8	1827 1	100	160.6	0.32	160.92
DS	92	28.4	1303	7.13	16.8	0.32	17.12

Case 3:
Name of the Sample: Nba.cif
Number of frames: 150

Figure 26(a) Nba

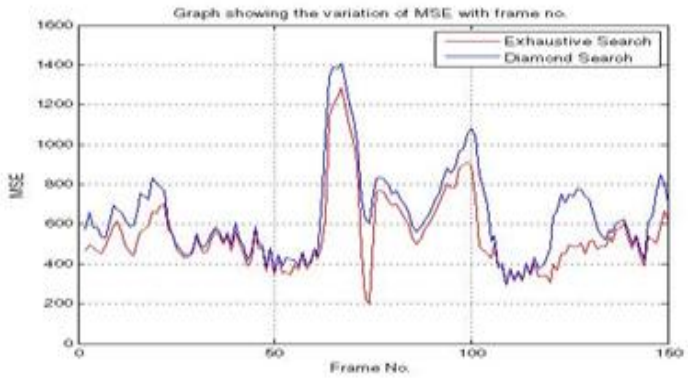

Figure 26(b): Frame No. Vs MSE

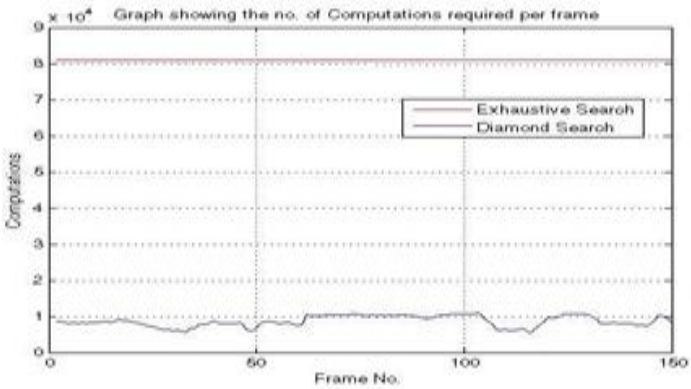

Figure 26(c): Frame No. Vs Computations

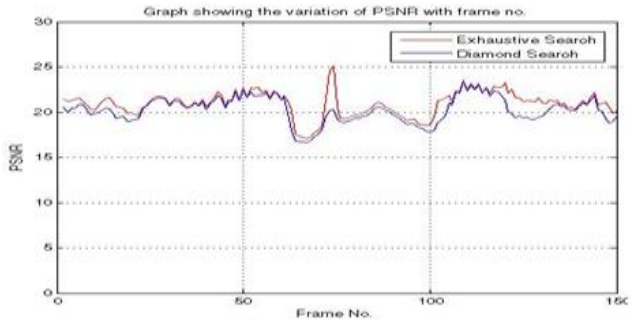

Figure 26(d): Frame No. Vs PSNR

Table 3: Performance Comparison of FS and DS with Nba image sequence

Algorithm	MSE	PSNR (dB)	Computations		Execution time(sec)		
			Number	%	T_1	T_2	$T=T_1+T_2$
FS	522	20.9	80896	100	696.4	11.1	707.5
DS	612	20.25	8635	10.6	109.4	13.4	122.8

VI. Conclusion

A digital modelling for the processing of video coder under variable resource allocation and demanded data rate is proposed. The developed design provides a simplified approach to the modelling of video processor for high rate processing and higher visual quality observation. The objective of developing a simplified approach of video coder is achieved. The obtained results illustrate the significance of utilizing the proposed system for quality of service in video codec. A relative accuracy in the obtained estimation for video sample at full search is observed. The PSNR value is obtained in improvement to the existing system. The computation time for such coding is observed to be minimized and hence the operational speed is improved in hardware system.

References

[1] Kristof Denolf, Adrian Chirila-Rus, Paul Schumacher, Robert Turney, Kees Vissers, Diederik Verkest, and Henk Corporaal, "A Systematic Approach to Design Low-Power Video Codec Cores", EURASIP Journal on Embedded Systems Volume 2007.
[2] Paul Schumacher, Wilson Chung, "FPGA-Based MPEG-4 Codec", DSP magazine, October 2005.
[3] Y. Zhang, W. Gao, Y. Lu, Q. Huang, and D. Zhao, "Joint source channel rate-distortion optimization for H.264 video coding over error prone networks," IEEE Trans. Multimedia, vol. 9, no. 3, pp. 445–454, Apr. 2007.
[4] Z. Chen, M. Li and Y.-P. Tan, "Perception-aware multiple scalable video streaming over WLANs," IEEE Signal Processing Letters, vol. 17, no. 7, pp. 675-678, July 2010.
[5] A. Argyriou, "Cross-layer adaptive ARQ for uplink video streaming in tandem wireless/wireline networks," in Proc. IEEE Multimedia Signal Processing Workshop (MMSP), Oct. 2007.
[6] S. Choudhury and J. D. Gibson, "Payload length and rate adaptation for multimedia communications in wireless LANs," IEEE J. Select. Areas Commun., vol. 25, no. 4, pp. 796–807, May 2007.
[7] Olivia nemethova , ameen al-moghrabi, "An adaptive Error concealment mechanism for H.264/AVC encoded low-resolution video streaming", Eurasip – 2006.
[8] Jordi Ortiz , Eduardo Martínez Graciá and Antonio F Skarmeta,: SCTP as scalable video coding transport," *EURASIP Journal on Advances in Signal Processing* 2013.
[9] Daniel Denning1, Neil Harold2, Malachy Devlin2, James Irvine3," Using System Generator To Design A Reconfigurable Video Encryption System," Institute of System Level Integration, Alba Centre, Alba Campus, Livingston, EH54 7EG, UK
[10] Shahrukh Agha and Vincent M Dwyer, "Algorithms and VLSI Architectures for MPEG-4 Motion Estimation", Electronic Systems and Control Division Research, 2003.
[11] Gahlot, S. Arya, D. Ghosh, "Object-Based Affine Motion Estimation, IEEE, 2003.
[12] Jie-Bin Xu, Lai-Man Po, and Chok-Kwan Cheung, "Adaptive Motion Tracking Block Matching Algorithms for Video Coding" , IEEE Transactions On Circuits And Systems For Video Technology, Vol. 9, No. 7, October 1999.
[13] Hwang-Seok Oh1 and Heung-Kyu Lee2, "Block-matching algorithm based on an adaptive reduction of the search area for motion estimation", Real-Time Imaging 6, 407-414, 2000.
[14] Yi-Hon Hsiao, Tien-Hsu Lee, and Pao-Chi Chang, "Short/Long-Term Motion Vector Prediction in Multi-Frame Video Coding System", IEEE, 2004.
[15] Y. Huang, S. Ma, C. Shen and L. Chen, "Predictive line search: An efficient motion estimation algorithm for MPEG-4 encoding systems on multimedia processors," IEEE Trans. on Circuits and Systems for Video Technology, vol. 13, pp. 111-117, Jan. 2003.
[16] J. Chen and K. Liu, "Efficient architecture and design of an embedded video coding engine," IEEE Trans. On Multimedia, vol. 3, pp. 285-297, Sep. 2001.
[17 C. Vleeschouwer, T. Nilsson, K. Denolf and J. Bormans, "Algorithmic and architectural co-design of a motion estimation engine for low-power video devices," IEEE Trans. on Circuits and Systems for Video Technology, vol. 12, pp. 1093-1105, Dec. 2002.Kavishseth, P.
[18] Rangarajan, S.Srinivasan, V.Kamakati, V.BalaKuteshwar, "A parallel Architectural Implementation of The New Three-Step Search Algorithm For Block Motion Estimation," Proceedings of the 17th International Conference on VLSI Design(VLSID' 04) 2004.