A High-Throughput and Low-Complexity H.264/AVC Intra 16×16 Prediction Architecture for HD Video Sequences

Milica Orlandić, *Student Member, IEEE*, and Kjetil Svarstad, *Member, IEEE*

***Abstract* — H.264/AVC compression standard provides tools and solutions for an efficient coding of video sequences of various resolutions. Spatial redundancy in a video frame is removed by use of intra prediction algorithm. There are three block-wise types of intra prediction: 4×4, 8×8 and 16×16. This paper proposes an efficient, low-complexity architecture for intra 16×16 prediction that provides real-time processing of HD video sequences. All four prediction (V, H, DC, Plane) modes are supported in the implementation. The high-complexity plane mode computes a number of intermediate parameters required for creating prediction pixels. The local memory buffers are used for storing intermediate reconstructed data used as reference pixels in intra prediction process. The high throughput is achieved by 16-pixel parallelism and the proposed prediction process takes 48 cycles for processing one macroblock. The proposed architecture is synthesized and implemented on Kintex 705 -XC7K325T board and requires 94 MHz to encode a video sequence of HD 4k×2k (3840×2160) resolution at 60 fps in real time. This represents a significant improvement compared to the state of the art.**

***Keywords* — H.264/AVC, encoder, intra prediction, FPGA, hardware implementation, plane mode.**

I. Introduction

An increasing number of portable devices such as mobile phones or tablets demand a network-friendly and an error-resilient video representation. Moreover, technological developments facilitate the development and design of complex video coding systems so the system complexity increases in order to provide enhancement in the compression performances. Compression algorithms, and in particular H.264/AVC, are characterized by high data dependencies and computational complexity. The H.264/AVC standard defines a block-based video codec as shown in Fig. 1. The basic functional elements in the encoder are prediction, transformation, quantization and entropy coding. In the decoding process, the decoded residual samples are entropy decoded, inversely quantized and transformed and then processed by prediction algorithms. The prediction process can be spatial (intra prediction) performed on macroblocks within a frame or temporal (inter prediction) used to explore similarities between a number of consecutive frames [1]. New tools introduced in H.264/AVC, such as intra prediction algorithm, provide a lower bitrate that ensures network transmission of various HD video sequences with respect to the constraints of the network. Furthermore, it provides a significant enhancement in compression performance in comparison to its predecessors. The enhancements are achieved by maintaining the quality while lowering the bitrate. The lower bitrate is implied by an accurate prediction because data transmitted over the network are quantized and transformed residual samples. The residual is defined as a difference between the original pixel and the prediction pixel at the same position within a macroblock. Moreover, the high throughput is considered as an important quality of the video processing system and it is dependent on the bitrate and level of parallelism achieved in the video algorithm implementation.

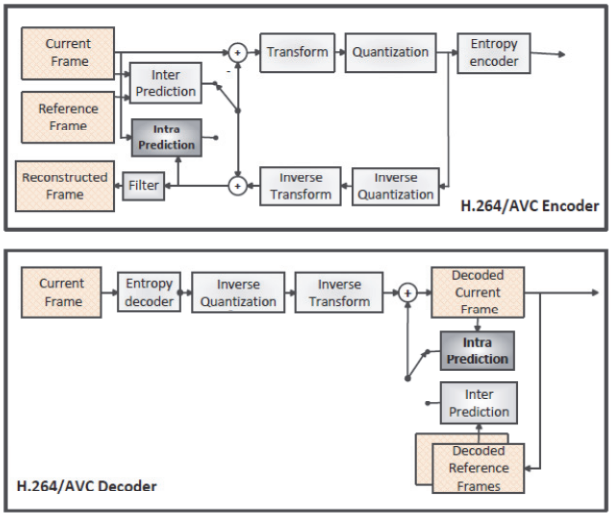

Fig. 1. Block diagram of H.264 Encoder and Decoder.

Intra prediction produces an estimate value of pixels in a macroblock by extrapolation of the neighboring macroblock pixels. The pixels used for prediction are 16 reconstructed pixels of the last column from the left neighbor and 16 reconstructed pixels of the last row from the upper neighbor macroblock. The estimation is based on the assumption of similarity between two macroblocks along the local edges. Prediction pixels are computed for a

Paper received March 11, 2014; accepted October 6, 2014. Date of publication November 15, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Miodrag Popović.

This paper is a revised and expanded version of the paper presented at the 21th Telecommunications Forum TELFOR 2013.

M. Orlandić and K. Svarstad are with the Department of Electronics and Telecommunication, Norwegian University of Science and Technology-NTNU, Trondheim, Norway, e-mail: milica.orlandic@iet.ntnu.no; kjetil.svarstad@iet.ntnu.no.

number of modes. The data to be transferred are a result of the mode selection process in a number of block-wise intra prediction algorithms. The H.264 standard specifies three block-wise intra coding types for luminance component: 4×4, 8×8 and 16×16. Smaller prediction blocks tend to give a more accurate prediction. There are defined nine directional modes for intra 4×4 and 8×8 predictions. Intra 16×16 prediction is used for smooth luminance blocks and it consists of four prediction modes: vertical, horizontal, DC and plane mode.

The computational complexity of video encoders increases due to the requirement for higher video resolutions. High definition (HD) video sequence transmission requires a high throughput in order to meet real-time processing requirements and a low bitrate due to the network limitations. In that sense, a number of optimized prediction algorithms and high-parallelism hardware implementations have been recently proposed. Related ASIC designs for H.264/AVC intra frame prediction have been proposed in [2]–[6]. A VLSI architecture [4] computes all four modes and employs a central-based computation for plane mode to remove multiplication operations. However, the increase of control and storage resources for reorganization of the incoming data needs to be accounted. An efficient architecture [5] focuses on low power consumption achieved by the reduction of decoding operations, elimination of redundant operations, data reuse and high pipeline employment. A partial datapath overlap of intra 4×4 and 16×16 predictions is achieved in [6] by the modification of plane mode equations. An increasing number of FPGA solutions have been proposed recently. The FPGA architecture [7] contains intra 16×16 prediction with the limitation that only three modes (V, H, DC) are performed in parallel whereas the complex plane mode is excluded. The supported resolutions are 720p (1280×720) and 1080i (1920×1080). An FPGA solution [8] achieves 4-sample parallelism. The critical path of the design is improved by use of DSP blocks and the architecture can process HD1080p resolution at 60 frames per second.

In this paper, an efficient hardware architecture of intra 16×16 prediction algorithm is proposed. The requirement is real-time encoding of HD video sequences such as 1080p and 4k×2k (3840×2160) at various frame rates (30fps, 60 fps) whereas the architecture is implemented on FPGA. The architecture supports 16-pixel parallelism in order to meet requirements for real-time processing of HD sequences. The architecture also contains an organized memory system for the intra-predicted data samples used in the prediction process. The paper represents an extended version of the work published in [9].

The paper is organized as follows. An introduction to intra prediction process, in particular to prediction modes in the intra 16×16 prediction, is presented in Section II. The proposed hardware architecture of the prediction module with memory support is presented in Section III. The results and discussion are presented in Section IV. Finally, conclusions are presented in Section V.

II. Background

The H.264 standard defines three intra coding types for luminance component: 4×4, 8×8 and 16×16. Both Intra 4×4 and Intra 16×16 are present in all profiles defined in the standard (main, baseline and high profiles), whereas Intra 8×8 is optional in the high profiles. Intra 16×16 prediction is used for smooth luminance blocks and it specifies four prediction modes: vertical, horizontal, DC and plane mode. The modes are depicted in Fig. 2. The plane mode takes significantly more computing cycles compared to the other prediction modes due to a number of computation stages required for final prediction.

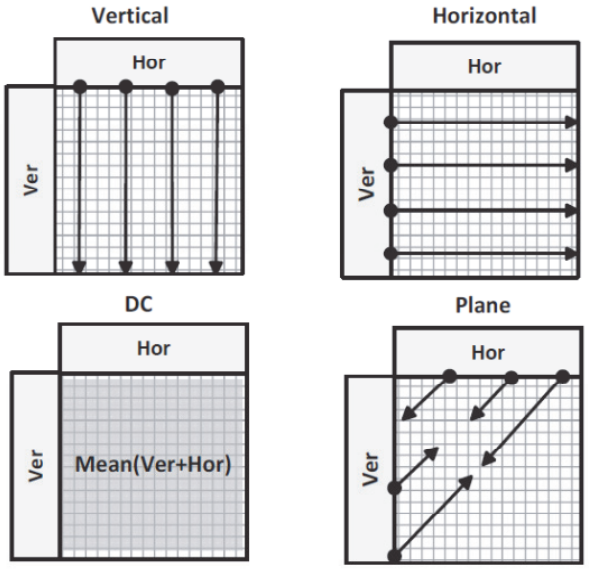

Fig. 2. Directional modes for intra 16×16 prediction.

Vertical mode extrapolates reconstructed intra-predicted pixel samples $P(x,-1)$ with x=0..15. Horizontal prediction mode extrapolates samples $P(-1,y)$ with y=0..15. DC mode computes the mean of the upper and/or left neighbor. In the case of the left edge of the frame, left macroblock neighbor does not exist and only pixels from the upper macroblock are used for computing prediction pixels in DC mode. Smoothly varying luminance regions are predicted by plane mode. The plane mode performs linear spatial interpolation of the neighbor samples. The mode is highly computational and requires the computation of 5 intermediate parameters. According to the H.264/AVC specification, the intermediate parameters H, V, a, b and c are computed from the reconstructed intra-predicted samples:

$$H = \sum_{x=0}^{7} (x+1) \times \left(P(8+x,-1) - P(6-x,-1)\right), \quad (1)$$

$$V = \sum_{y=0}^{7} (y+1) \times \left(P(-1,8+y) - P(-1,6-y)\right), \quad (2)$$

$$a = 16 \times \left(P(-1,15) + P(15,-1)\right), \quad (3)$$

$$b = \left(5 \times H\right) >> 6, \quad (4)$$

$$b = \left(5 \times H\right) >> 6, \quad (5)$$

where (x, y) represent position of the pixel in the macroblock.

Finally, the round-off function limits the prediction pixel values $a + b(x - 7) + c(y - 7) >> 5$ to the range [0,255]:

$$Plane(x,y) = R_{off}(a + b(x-7) + c(y-7) >> 5). \quad (6)$$

III. Hardware Architecture

Prediction algorithms in H.264/AVC standard are characterized by high and complex data dependency. This can provoke a long idle time (bubbles) in the hardware while waiting for data. On the other side, an important requirement for the implementation is real-time encoding of HD video resolution sequences (720p, 1080p, 4k×2k). In this paper an efficient and fast intra 16×16 prediction architecture that can process HD video sequences is presented. The complete intra prediction engine in the encoder contains two parts: a prediction unit and a reconstruction loop unit. The block diagram of the entire system is drawn in Fig 3. The prediction unit computes the prediction modes and finds the best prediction mode by a minimal cost function. A number of cost functions can be used for this purpose such as SSD, SATD or SAD. The Sum of Absolute Differences (SAD) is used in the proposed design and is formulated by:

$$SAD_{\text{mode}}(O, P_{\text{mode}}) = \sum_{y=1}^{16}\sum_{x=1}^{16} |O(x,y) - P_{\text{mode}}(x,y)|, \quad (7)$$

where O and P_{mode} are original macroblock and prediction samples macroblock for each supported mode, respectively.

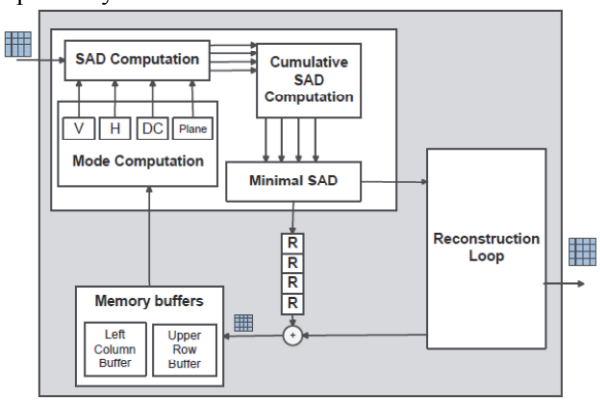

Fig. 3. Block diagram of intra 16×16 prediction algorithm.

In this paper the implementation of the intra prediction unit and its memory support is presented. The reconstruction loop unit is implemented in [10]. The loop provides data needed for the prediction process of the incoming macroblocks. It consists of four modules: forward integer DCT, quantization, inverse quantization and inverse integer DCT transform. The input of the reconstruction loop consists of residual samples of a 4×4 block, whereas the output of interest consists of reconstructed residual samples of the same block. In order to lower the processing time for one macroblock prediction, the reconstruction loop implementation is characterized by a high throughput achieved by processing a 4×4 block in parallel. Residual samples within a 4×4 block are processed by the reconstructed loop in 4 clock cycles. This time lap required for reconstruction of the residual data has been taken into account for the purpose of data synchronization during the implementation process of intra 16×16 prediction unit. The output of the reconstruction loop is the input data required for the computation of reference samples for the prediction. Furthermore, the reconstruction loop time has been included in the total time required for intra 16×16 prediction of one macroblock.

The intra prediction processes are highly data-dependent, and data prediction of an active macroblock is computed based on the reference samples that are along the left and upper edge. The complete dataflow of the intra 16×16 prediction process is presented in Fig. 4. The processing cycle for one macroblock takes 48 clock cycles and contains four phases: pre-processing phase, input phase, reconstruction phase and memory-communication phase. The processing cycle starts by fetching the reconstructed data samples from the local memory buffers. The fetched data are used for computing the prediction samples for each mode. This phase is called pre-processing phase and it lasts 8 clock cycles. The values of prediction samples assigned by Vertical and Horizontal modes do not require additional computation, but position tracking of an active 4×4 block is required. DC mode performs summation over the entire set of reference samples. The summation operation is followed by shifting. When the macroblock has both left and upper neighbor macroblocks, i.e it is not on the left or upper edge of the frame, the summation of 32 samples followed by shifting by 5 is performed. In the case when the macroblock has only left or upper neighbor, 16 reference samples that correspond to left/upper edge are summed up and shifted by 4. Prediction pixels through the complete macroblock have the same DC value.

The plane mode computes five intermediate parameters H, V, a, b and c. The positions of the neighbor reference samples are used for the computation of parameters H and V. In this phase only prediction samples for the first 4×4 block are computed. The prediction samples within a macroblock are represented as $P(x,y)$ where x, y=0..15. The computations of all intermediate parameters based on Eq. 1 - Eq. 5 are only performed for a seed prediction pixel $P(0, 0)$. The round-off operation for a seed sample includes the coordinates of the prediction sample and its form is presented by:

$$P(0,0) = a - 7b - 7c. \quad (8)$$

The equations for each prediction sample in a macroblock are different due to the dependency on the position of the prediction samples. However, there is the equations inter-dependency pattern that provides the elimination of redundant computations. Depending on the position within the 4×4 block, the prediction samples for plane mode are formed by adding parameters b or c to the prediction value of the neighboring pixels. The relation of the prediction samples over a4×4 block is presented Fig. 5. For example, the seed value $P(0,0)$ of a 4×4 block (labeled 1) is computed. The prediction value of its lower neighbor $P(1, 0)$ is defined as:

$$P(1,0) = P(0,0) + b, \quad (9)$$

whereas the prediction value of its neighbor sample $P(0, 1)$ on the right is computed as:

$$P(0,1) = P(0,0) + c. \quad (10)$$

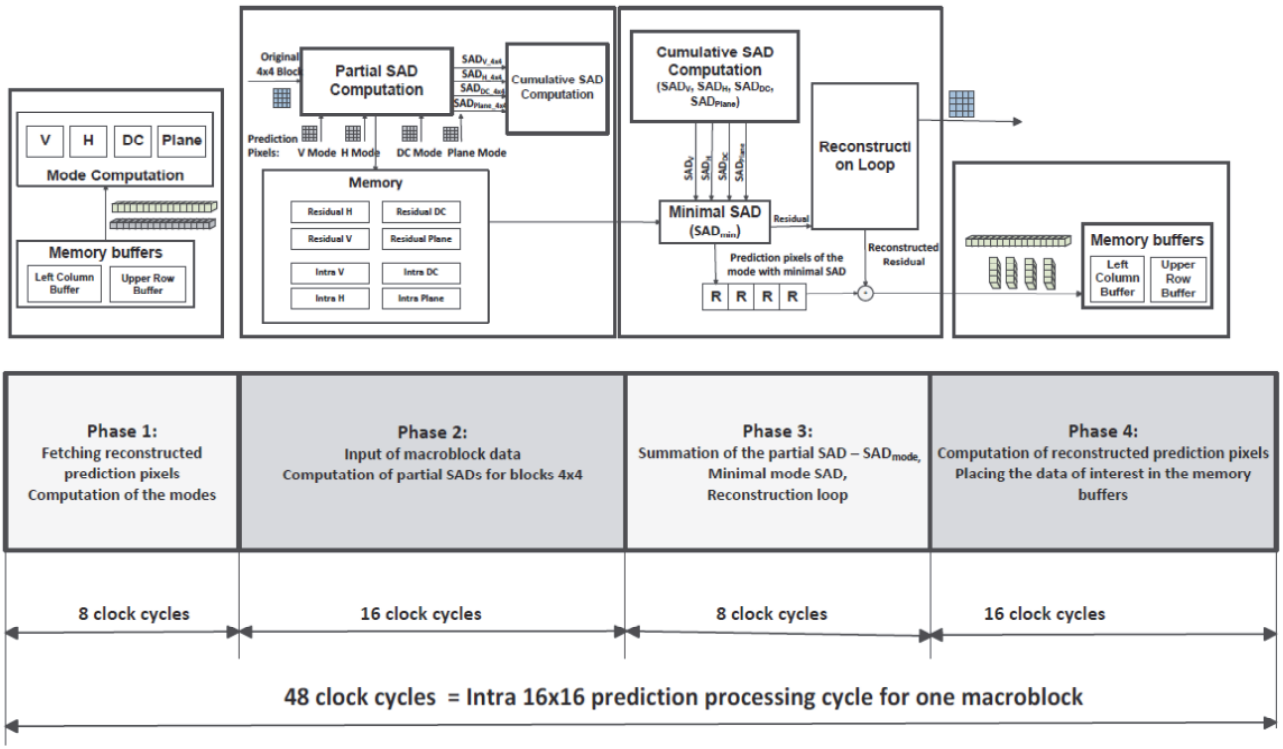

Fig. 4. Intra 16×16 prediction dataflow.

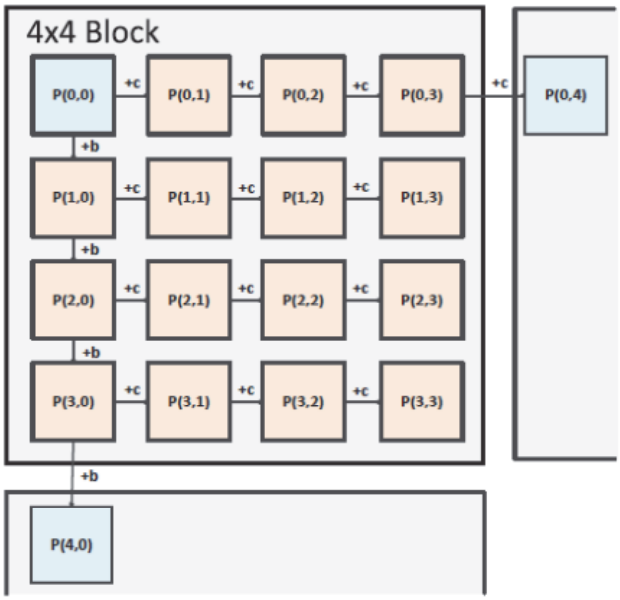

Fig. 5. Plane mode calculation for one 4×4 block.

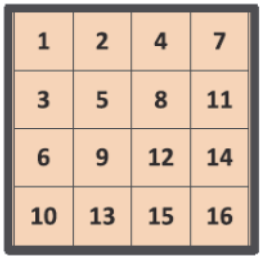

Fig. 6. Block scanning order within a macroblock.

The samples $P(0, 4)=P(0, 3) + c$ and $P(4, 0)=P(3, 0) + b$ are saved in registers to be used as seed values for the 4×4 blocks that are labeled 2 and 3 in Fig. 6.

The second phase is called input phase. After the samples for all four prediction modes are computed, intra 16×16 prediction unit receives sixteen original uncompressed pixels in parallel during the period of 16 clock cycles. The proposed scanning 4×4 block order is presented in Fig. 6. The SAD values on the 4×4 block level are calculated for the each mode. The comparison of SAD function values should be performed on the macroblock level, therefore it is required to save and sum up partial SAD values computed for each 4×4 block. The macroblock cost function SADmode for each prediction mode (SAD_V, SAD_H, SAD_{DC}, SAD_{Plane}) is a sum of partial SAD values over sixteen 4×4 blocks. The computation of the SAD_{mode} is performed during the input phase by adding partial SAD for each block. The value of SAD_{mode} for the complete macroblock is computed in the last clock cycle of the input phase. In the first stage of SAD computation, the residual samples that are a difference between original pixels and prediction samples are computed. These data samples are stored in the memory.

The third phase, reconstruction phase, starts by a comparison of the SAD_V, SAD_H, SAD_{DC}, SAD_{Plane} values where the mode with the minimal SAD_{mode} is chosen for further processing. The residual samples that correspond to the mode with minimal SAD_{mode} are further transferred to the reconstruction loop. The reconstruction loop process takes 4 clock cycles per 4×4 block. Firstly, the residual is transformed by an integer DCT transform and quantized. The quantized residual is forwarded to the entropy encoder to be sent out to the network. On the other hand, the quantized residual is reconstructed by inverse quantization and inverse transform.

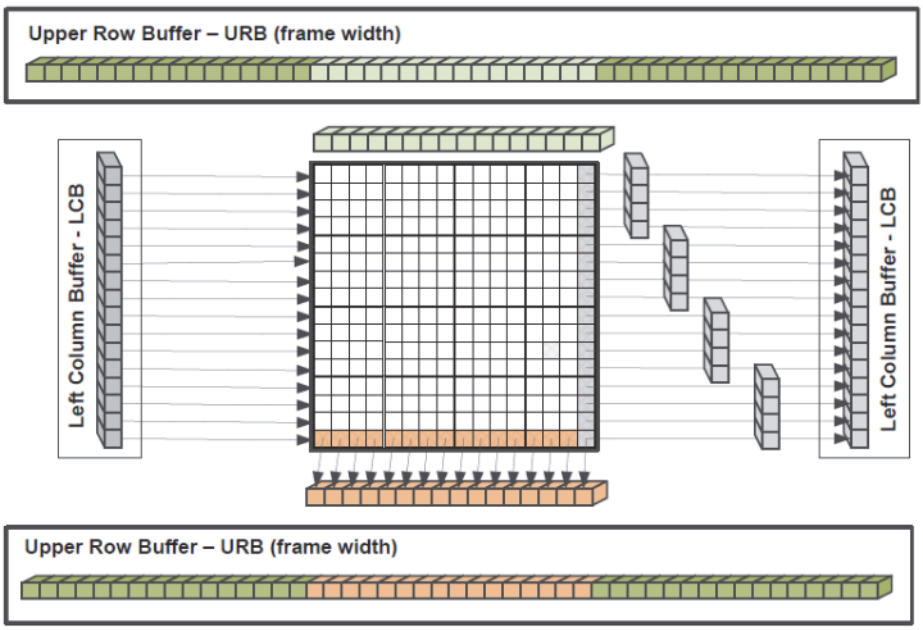

Fig. 7. Proposed memory organization.

In the last phase the communication with memory buffers is performed. The reconstructed residual and prediction samples for the chosen mode, fetched from the memory in the input phase, are summed and the resulting pixels of interest (last column and last row of reconstructed macroblock pixels) are stored in memory buffers. The data (i.e reconstructed intra-predicted samples) used as prediction reference samples for incoming blocks are stored in two local memory buffers. The communication dataflow between intra prediction module and memory buffers is depicted in Fig. 7. There are two buffers for storing intermediate data: Upper Row Buffer (URB) and Left Column Buffer (LCB). The horizontal and vertical buffers are organized with respect to read and write indexing in such a way that data dependency in each prediction computation should be immediately available thus minimizing latency.

The Upper Row Buffer (URB) stores data from the upper neighbor macroblocks. The buffer can store a number of samples that correspond to the frame width. The mechanism includes tracking the horizontal position of an active macroblock within the frame. The data from the URB buffer in the same horizontal position contains the last row data (16 samples) from the upper macroblock neighbor. These data are used in the prediction process for computing the samples of prediction modes. Furthermore, the tracking within a macroblock is performed due to the dependency of prediction modes and position of a 4×4 block. For example, in the case of the Vertical mode (V) the values of the first four samples fetched from the upper memory buffer are assigned to blocks that are labeled 1, 3, 6, 10 in Fig. 6. In the last phase of the processing cycle, the sum of intra predicted samples and reconstructed residual is stored at the same position data was previously gathered from.

Left Column Buffer (LCB) stores data that corresponds to a macroblock column. The data in the buffer are samples from the left neighbor macroblock. The buffer is partially updated in the memory communication phase when specific 4×4 blocks (labeled 7, 11, 14, 16 in Fig. 6) are processed. For example, in the pre-processing phase the complete content of LCB buffer is collected from the buffer and used for prediction modes (Horizontal, DC and Plane). In the memory-communication phase, the blocks from the last macroblock column contain data of interest for prediction process. The last column of the reconstructed 4×4 block with label 7 is stored in the first four positions of the LCB. The column from block 11 is stored in the positions 5-8 of LCB and so on. The process is depicted in Fig. 7.

IV. Results

The proposed architecture is designed in VHDL, synthesized with Xilinx Project Navigator 14.4 and placed and routed on Kintex 705 -XC7K325T development board. Matlab model has been developed for experimental verification in order to produce input vectors. The Matlab reference model results were compared with JM18.3 reference software results [11].

The functional verification is performed by Active HDL software using test vectors generated by Matlab reference model.

The VHDL description of the architecture is synthesized for Kintex 705 -XC7K325T. The results are summarized in Table I. The proposed design provides intra 16×16 prediction with four prediction modes. The processing cycle for one macroblock lasts 48 clock cycles. It contains four phases during which reference samples are fetched and stored from/in the local memory buffer, prediction computations are created and reconstruction process is performed. To avoid the use of external memories and minimize the late, two local memory buffers (URB and LCB) and a number of local registers are used. The performance characteristics are presented and compared with other FPGA implementation solutions in Table II. The throughput of 629 MB/s is achieved. The proposed

design performance takes fewer clock cycles when compared with the state of the art. Results show that the real-time processing of HD video sequences (4k×2k, 1080p and 720p) is achieved. The required system operating frequency is as low as 94 MHz for real time encoding of 4k×2k (3840x2160) at 60 fps.

TABLE 1: SYNTHESIS RESULTS

Module	FPGA Slices	FF Utilization	LUTS	RAM (kbits)	Multipliers
Intra Prediction	152.5	2412	4465	18.7	-

TABLE 2: COMPARISON TO RECENT WORKS

Design	[8]	[7]	Proposed
Technology	FPGA Virtex-5 VLX110T	FPGA Stratix III EP3SL150	Kintex 705 XC7K325T
Maximum Throughput [MB/s]	20	177	629
Cycles/Macrobl.	300	188	48
Frequency [MHz]	168	130	118
Minimum frequency HD4k×2k/60fps	-	-	94
Minimum frequency HD4k×2k/30fps	-	-	47
Minimum frequency HD1080p/30fps	73.4	-	12
Minimum frequency HD720p/30fps	32.4	20	5

V. CONCLUSIONS

This paper presents an efficient and low-complexity architecture for intra 16×16 prediction algorithm in H.264/AVC standard. The architecture is characterized by a high degree of parallelism that provides a high throughput requested by increasing video resolutions. The intra prediction algorithm contains a high level of data dependency between macroblocks. The high computational burden of the plane prediction mode is decreased by the removal of computational redundancy in the prediction sample equations. To solve the conflict between increased coding complexity and increasing resolution requirements, we propose a fast hardware implementation of intra 16×16 prediction algorithm that is characterized by 16-pixel parallelism. The prediction process consists of four phases. The parameters and reference samples for each prediction mode are calculated in the first phase. The original macroblock data are fetched and the cost functions are computed in the second phase. The last two phases that are fully pipelined and provide reference samples for the incoming macroblocks. The design can process SD and HD video sequences up to HD 4k×2k (3840x2160). The proposed design takes fewer clock cycles for real time processing of HD 1080p video sequences compared to the state of the art. In addition, the prediction module supports real time encoding of HD 4k×2k (3840x2160) at 30 fps and 60 fps.

REFERENCES

[1] I. Richardson, *The H.264 Advanced Video Compression Standard*. Wiley, 2010. [Online]. Available: http://books.google.no/books?id=LJoDiPnBzQ8C

[2] Y. W. Huang, B. Y. Hsieh, T. C. Chen, and L. G. Chen, "Analysis, fast algorithm, and vlsi architecture design for h. 264/avc intra frame coder," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 3, pp. 378–401, 2005.

[3] H. Ren, Y. Fan, X. Chen, and X. Zeng, "A 16-pixel parallel architecture with block-level/mode-level co-reordering approach for intra prediction in 4k× 2k h. 264/avc video encoder," in *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, IEEE, 2012, pp. 801–806.

[4] S. Hsia, W. Hsu, and Y. Chou, "Fast low-complexity computation and real-time architecture for h.264/avc intraprediction," *Real time Image Processing*, 2012.

[5] K. Xu and C.-S. Choy, "A power-efficient and self-adaptive prediction engine for h. 264/avc decoding," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 16, no. 3, pp. 302–313, 2008.

[6] M. Nadeem, S. Wong, and G. Kuzmanov, "An efficient hardware design for intra-prediction in h. 264/avc decoder," in *Electronics, Communications and Photonics Conference (SIECPC)*, 2011 Saudi International. IEEE, 2011, pp. 1–6.

[7] A. B. Atitallah, H. Loukil, and N. Masmoudi, "Fpga design for h. 264/avc encoder," *International Journal of Computer Science, Engineer-ing and Applications*, vol. 1, no. 5, 2011.

[8] C. M. Diniz, A. A. Susin, and S. Bampi, "Fpga design of h. 264/avc intra-frame prediction architecture for high resolution video encoding," in *Programmable Logic (SPL), 2012 VIII Southern Conference on*, IEEE, 2012, pp. 1–6.

[9] M. Orlandic and K. Svarstad, "A high-throughput and low-complexity h.264/avc intra 16x16 prediction architecture for hd video sequences," in *Telecommunications Forum (TELFOR), 2013 21st*, Nov 2013, pp. 529–532.

[10] M. Orlandic and K. Svarstad, "An area efficient hardware architecture design for h. 264/avc intra prediction reconstruction path based on partial reconfiguration," in *2013 IEEE 16th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, IEEE, 2013, pp. 86–91.

[11] (2012) Reference software for h.264/avc codec jm18. [Online]. Available: http://www.iphome.hhi.de/suehring/uml/index.htm