An Efficient Hardware Design for Intra-prediction in H.264/AVC Decoder

Muhammad Nadeem, Stephan Wong, and Georgi Kuzmanov
Computer Engineering Laboratory, Delft University of Technology
Mekelweg 4, 2628CD Delft, The Netherlands
{M.Nadeem, J.S.S.M.Wong, G.K.Kuzmanov}@tudelft.nl

***Abstract*—The H.264/AVC intra-frame codec is widely used to compress image/video data for applications like *Digital Still Camera* (DSC), *Digital Video Camera* (DVC), *Television Studio Broadcast*, and *Surveillance video*. Intra-prediction is one of the top 3 compute-intensive processing functions in the H.264/AVC baseline decoder [6] and, therefore, consumes significant number of compute cycles a processor. In this paper, we propose a configurable, high-throughput, and area-efficient hardware design for the intra-prediction unit. The intra-prediction algorithm is optimized to significantly reduce the redundancy in addition operations (e.g., 27% reduction when compared with state-of-the-art in literature [12]). The area requirement for our hardware implementation of the optimized intra-prediction algorithm is further reduced by employing a configurable design to reuse data paths for mutually exclusive processing scenarios. The proposed design is described in VHDL and synthesized under $0.18\mu m$ CMOS standard cell technology. While working at a clock frequency of 150 MHz, it can easily meet the throughput requirement of HDTV resolutions and consumes only 21K gates.**

***Keywords*-intra-prediction; H.264/AVC decoder; image and video compression; inverse integer transform.**

I. INTRODUCTION

The Advanced Video Coding standard H.264/AVC, also known as MPEG-4 part 10, is jointly developed by ITU-T VCEG and ISO/IEC MPEG [1]. It is able to achieve significantly higher compression efficiency over the previous video coding standards, like H.263 and MPEG-2/4. The H.264/AVC provides similar subjective video quality as that of MPEG-2 with at least 50% more reduction in bit-rate [2], [3]. It provides up to 30% better compression when compared with H.263+ and MPEG-4 Advanced Simple Profile (ASP) [4]. This significantly higher compression efficiency is achieved at the cost of additional computational complexity of the video coding algorithms in H.264/AVC, approximately 10 times higher than the MPEG-4 simple profile [4].

The H.264/AVC supports multiple directional *intra prediction modes* (4 modes for the luma 16×16 / chroma 8×8 block types and 9 modes for the 4×4 block type) to reduce the spatial redundancy in the video signal. These multiple intra prediction modes help to significantly improve the encoding performance of an H.264 intra-frame encoder. Studies [6] shown that H.264/AVC outperforms JPEG2000, a state-of-the-art still-image coding standard, in terms of subjective as well as objective image quality. This makes the H.264/AVC intra-frame codec, an attractive choice for an image compression engine. Applications like *Digital Still Camera (DSC)* employ intra-frame encoding technique to compress high-resolution images.

In a video frame encoded in intra mode, the current macro-block (i.e., 16×16 pixels block) is predicted from the previously encoded neighboring macro-blocks (MB) from the same video frame. Therefore, an *intra-frame* with all *intra-MB* does not depend on any other video frame(s) and can be decoded independently. A video encoded with intra-frames only is easier to edit than video with inter-frames (frames predicted from past or future video frames). Similarly in many surveillance systems, the video is compressed using intra frames encoding mode due to legal reasons. Courts in many countries do not accept the predicted image frames as legal evidence [5]. As a result, a typical video surveillance system compresses video using intra encoded frames only. Consequently, intra-only video coding is widely used coding technique in *Television Studio Broadcast, Digital Cinema* and *Surveillance video* applications.

The H.264/AVC baseline decoder is approximately 2.5 times more time complex than the H.263 baseline decoder [6]. According to the analysis of run-time profiling data of H.264/AVC baseline decoder sub-functions, intra-prediction is one of the top 3 compute-intensive functions [6]. A high-throughput intra-frame processing chain is, therefore, an important requirement for H.264/AVC *intra-frame decoder* for real-time video processing applications. The demanding characteristics of the intra-prediction algorithm suggest a hardware implementation for such function for high-definition video applications, where even larger frame sizes at higher frame rates are to be processed in real time.

Several different hardware architectures have been proposed in the literature in last few years and most of them are developed from the encoder's requirements point of view. In a video decoder, while decoding the compressed input video bit-stream, the block type and intra-prediction mode for the current macro-block are

978-1-4577-0069-9/11/$26.00 ©2011 IEEE

already known. Therefore, this information can be used to reduce on-chip area cost by designing overlapped data paths for mutually exclusive processing scenarios. In this paper, we focus on the realization of a high-throughput and area-efficient design of intra-prediction unit in H.264/AVC video decoder and provide the following specific contributions:

- *A proposal to optimize the intra-prediction algorithm for* 4×4 *luma blocks by decomposing the filter kernels. The proposed decomposition significantly reduces the additions operations for its hardware implementation (*27% - 60% *reduction).*
- *A configurable hardware design of intra-prediction module to reduce on-chip area by using hardware sharing approach for mutually exclusive processing scenarios (approx.* 21*K gates).*

Furthermore, we also optimize the common equations in the intra-prediction algorithm, to compute the prediction samples for 16×16 luminance and 8×8 chrominance blocks.

The remainder of this paper is organized as follows. The related work is presented in Section II. Section III provides an overview of the intra-prediction algorithm and also describes the proposed optimizations in the algorithm. The proposed configurable hardware design is presented in Section IV, whereas design evaluation is provided in Section V. Finally, Section VI concludes this paper.

II. RELATED WORK

In the last few years, many researchers proposed a number of optimized algorithms and efficient hardware implementations for intra-prediction unit in H.264/AVC. For instance, a hardware implementation with five-stage registers and three configurable data paths for intra-prediction is proposed in [8]. The proposed solution can process approximately 100 VGA frames in real time.

In [9], an efficient intra-frame codec is proposed. The proposed solution can process 720p @ 30 fps in real time and can be used for both the encoder and decoder implementations. This solution, however, excludes the plane mode for 16×16 luma and 8×8 chroma blocks. The proposed solution, therefore, can be used in a matched encoder-decoder scenarios only where it is guaranteed that plane mode is not used for intra-prediction.

High-throughput hardware designs for a H.264/AVC decoder are proposed in [10] and [11]. The proposed designs can process high-definition (HD) video in real time. Since no attempts were made to optimize the intra-prediction algorithm to reduce the arithmetic operations, therefore, the hardware implementation of these designs

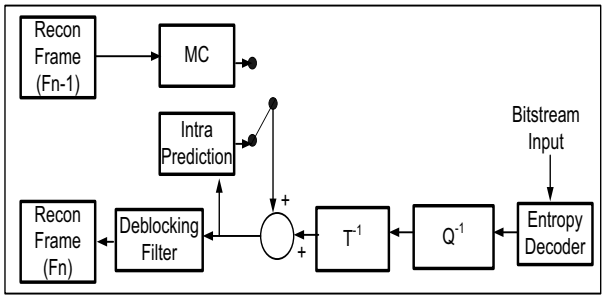

Figure 1. Functional block diagram of H.264/AVC decoder.

cost significant amount of hardware resources (approximately 29K gates).

Similarly, in [12], an efficient hardware implementation for intra-prediction unit is proposed. The design targets the H.264/AVC encoder and uses a so-called combined module approach to generate a subset of intra-prediction modes in parallel. Furthermore, the intra-prediction algorithm is optimized to significantly reduce the number of arithmetic operation for the computation of prediction samples. The proposed solution, however, does not fully eliminate the redundancy in the algorithm and also primarily targets the H.264/AVC encoder.

In short, most of the hardware implementations presented in the literature do not specifically target the H.264/AVC decoder implementation and, *therefore, fails to fully exploit the prior intra-prediction mode information to reduce area cost.* Some of them do not support the computation of all of the intra-prediction modes and *therefore, cannot be used in a general H.264/AVC video decoder where any of the intra-prediction modes can appear in the encoded bit-stream.* Similarly, some of them have made attempts to reduce the redundancy in the intra-prediction algorithm. *However, with the proposed decomposition in this paper, further arithmetic operations reduction is possible.*

III. INTRA-PREDICTION ALGORITHM

In this section, we briefly introduce the intra-frame processing chain in an H.264/AVC decoder. Subsequently, an overview of the intra-prediction algorithm for 4×4 luminance blocks, 16×16 luminance blocks, and 8×8 chrominance blocks is provided in separate subsections. The optimizations to reduce the number of operations in the intra-prediction algorithm are also explained along with algorithm overview.

The functional block diagram for an H.264/AVC intra-frame decoder is depicted in Fig. 1. The entropy decoder unit parses the input bit-stream and decodes the intra-prediction mode used for the current MB. This intra-prediction mode information is passed on to the intra-prediction unit to generate the prediction block.

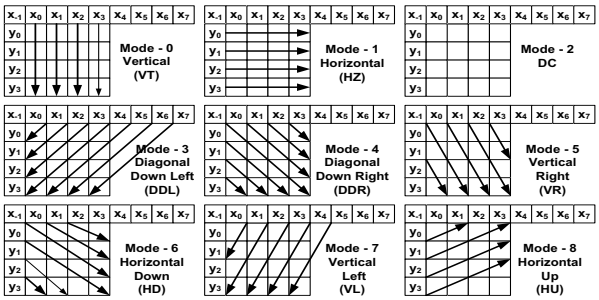

Figure 2. Illustration of nine 4x4 luminance prediction modes.

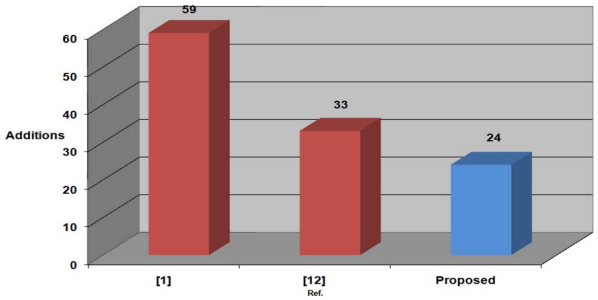

Figure 3. Comparison: Additions for 4x4 luminance intra-prediction modes.

The residual block can be computed in a parallel data-path using the *inverse quantization* (Q^{-1}) and the *inverse integer transform* (T^{-1}) processing units as depicted in Fig. 1. Once the intra-prediction block is made available by intra-prediction processing unit, the current pixel block is reconstructed by adding the residual block to the predicted block. The unfiltered reconstructed pixels of the current block are used as neighbor or boundary pixels for the generation of the prediction samples for the next block in the video frame.

The H.264/AVC video coding standard supports multiple intra-prediction modes for 4×4 and 16×16 luminance pixels block types, and the 8×8 chrominance pixels block type as explained in the following subsections.

A. *Intra-prediction* 4×4 *Luminance Block*

There are 9 intra-prediction modes for a 4×4 luminance pixels blocks in the H.264/AVC. Each of these prediction modes generate 4×4 predicted pixel block using some or all of the neighboring pixels as depicted in Fig. 2. Since the prediction samples for Vertical (VT) and Horizontal (HZ) prediction modes are same as the boundary pixels, therefore, these prediction modes do not require any processing and are easy to implement. Similarly, the DC prediction mode compute the average value using valid boundary pixels for all the prediction samples. The remaining 6 modes (modes 3 to 8), however, compute the prediction samples using 3- or 2-taps filter kernels. The filter equations for these modes in the H.264/AVC video coding standard require 59 addition operations [1] for the computation of prediction samples. Efforts have been made to reduce the addition operations by efficient reuse of intermediate results. The optimized data path proposed in [12] requires 33 addition operations and is the lowest number reported in literature.

In this paper, we propose to decompose the filter kernels in unique intra prediction equations for luminance intra 4×4 modes 3 to 8. The final 24 filter equations after decomposition are depicted in Fig. 4. With the proposed

$$
\begin{aligned}
&1.\ \ s_1 = x_{-1} + x_0 + 1\\
&2.\ \ s_2 = x_0 + x_1 + 1\\
&3.\ \ s_3 = x_1 + x_2 + 1\\
&4.\ \ s_4 = x_2 + x_3 + 1\\
&5.\ \ s_5 = x_3 + x_4 + 1\\
&6.\ \ s_6 = x_4 + x_5 + 1\\
&7.\ \ s_7 = x_5 + x_6 + 1\\
&8.\ \ s_8 = x_6 + x_7 + 1\\
&9.\ \ s_9 = x_{-1} + y_0 + 1\\
&10.\ s_{10} = y_0 + y_1 + 1\\
&11.\ s_{11} = y_1 + y_2 + 1\\
&12.\ s_{12} = y_2 + y_3 + 1\\
&\\
&13.\ t_1 = s_1 + s_2 + 1 = (x_{-1} + 2x_0 + x_1 + 2)\\
&14.\ t_2 = s_2 + s_3 + 1 = (x_0 + 2x_1 + x_2 + 2)\\
&15.\ t_3 = s_3 + s_4 + 1 = (x_1 + 2x_2 + x_3 + 2)\\
&16.\ t_4 = s_4 + s_5 + 1 = (x_2 + 2x_3 + x_4 + 2)\\
&17.\ t_5 = s_5 + s_6 + 1 = (x_3 + 2x_4 + x_5 + 2)\\
&18.\ t_6 = s_6 + s_7 + 1 = (x_4 + 2x_5 + x_6 + 2)\\
&19.\ t_7 = s_7 + s_8 + 1 = (x_5 + 2x_6 + x_7 + 2)\\
&20.\ t_8 = s_8 + 2x_7 + 1 = (x_6 + 3x_7 + 2)\\
&\\
&21.\ t_9 = s_9 + s_{10} + 1 = (x_{-1} + 2y_0 + y_1 + 2)\\
&22.\ t_{10} = s_{10} + s_{11} + 1 = (y_0 + 2y_1 + y_2 + 2)\\
&23.\ t_{11} = s_{11} + s_{12} + 1 = (y_1 + 2y_2 + y_3 + 2)\\
&24.\ t_{12} = s_{12} + 2y_3 + 1 = (y_2 + 3y_3 + 2)
\end{aligned}
$$

Figure 4. Proposed decomposed filter kernels for 4x4 intra prediction modes.

decomposition, the number of additions to compute the predicted samples for these modes is reduced to 24. Our approach provides 60% reduction in addition operations when compared with standard unique intra-prediction equations and 27% reduction when compared with the state-of-the-art [12]. A comparison with respect to number of addition operations required for the 4x4 intra-prediction modes is provided in Fig. 3. In Section IV, we show that the proposed decomposition helps to design the common data-paths for the intra-prediction unit which further reduces the on-chip area cost for its hardware implementation.

```
1.  Intra Chroma Plane Prediction Mode-3
2.  If (p[x, -1] and p[-1, y] with x = 0 .. 7 and, y = -1 ..7 are available) Then
3.
4.      The values of the prediction samples pred_c[x,y] are derived as follows:
5.      pred_c[x,y] = Clip (a + b * (x - 3) + c * (y - 3) + 16 >> 5)
6.
7.      Where
8.      a = 16 * (p[-1, 7] + p[7,-1])
9.      b = (5 * H + 32) >> 6
10.     c = (5 * V + 32) >> 6
11.
12.     and H and V are specified as
13.     H = Σ (x' +1) * (p[4+x', -1] - p[2-x', -1], where x' = 0 .. 3
14.     V = Σ (y' +1) * (p[-1, 4+y'] - p[-1, 2-y'], where y' = 0 .. 3
15.
16 . End If
```

Figure 5. Intra-prediction algorithm for 8x8 luminance block.

```
1.  Optimized Intra Chroma Plane Prediction Mode-3
2.
3.      H = (x4 – x2) + 2 * (x5 – x1) + 3 * (x6 – x0) + 4 * (x7 – x-1)
4.  => H = (x4 + 2 * x5 + 3 * x6 + 4 * x7 + 4) - (x2 + 2 * x1 + 3 * x0 + 4 * x-1 + 4)
5.  => H = ((x4 + 2 * x5 + x6 + 2) + 2 * (x6 + x7 + 1) + 2 * x7) ) -
6.                  ((x2 + 2 * x1 + x0 + 2) + 2 * (x0 + x-1 + 1) + 2 * x-1) )
7.  => H = (t6 + 2 * s8 + 2 * x7) - (t2 + 2 * s1 + 2 * x-1)
8.  => H = (t6 - t2) + 2 * ((s8 - s1) + (x7 - x-1))
9.
10.     Similarly
11.     V = (t6 - t2) + 2 * ((s8 - s1) + (x7 - x-1))
12.
13.     pred_c[x,y] = Clip ((k +  b*x + c*y ) >> 5), From line 5
14.     Where    k = a – 3*(b + c) + 16
15. => k = 16 * (x7 + y7) - 3*(b + c) + 16,  From line 8
16. => k = ((x7 + y7 + 1) << 4) -( (b + c) + ((b + c) << 1))
```

Figure 6. Optimized intra-prediction algorithm for 8x8 luminance block.

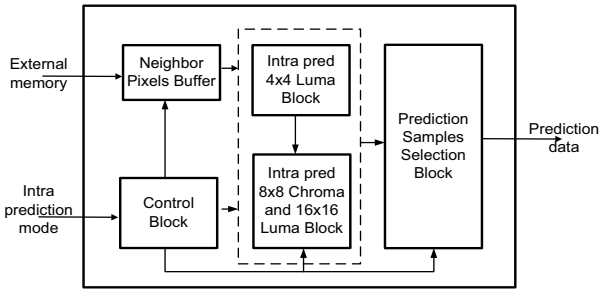

Figure 7. Top level organization of intra-prediction unit in H.264/AVC decoder.

$$pred_c[x,y] = clip\left(\left(\begin{bmatrix} k & k & k & k & k & k & k & k \\ k & k & k & k & k & k & k & k \\ k & k & k & k & k & k & k & k \\ k & k & k & k & k & k & k & k \\ k & k & k & k & k & k & k & k \\ k & k & k & k & k & k & k & k \\ k & k & k & k & k & k & k & k \\ k & k & k & k & k & k & k & k \end{bmatrix} + \begin{bmatrix} 0 & b & 2b & 3b & 4b & 5b & 6b & 7b \\ 0 & b & 2b & 3b & 4b & 5b & 6b & 7b \\ 0 & b & 2b & 3b & 4b & 5b & 6b & 7b \\ 0 & b & 2b & 3b & 4b & 5b & 6b & 7b \\ 0 & b & 2b & 3b & 4b & 5b & 6b & 7b \\ 0 & b & 2b & 3b & 4b & 5b & 6b & 7b \\ 0 & b & 2b & 3b & 4b & 5b & 6b & 7b \\ 0 & b & 2b & 3b & 4b & 5b & 6b & 7b \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c & c & c & c & c & c & c & c \\ 2c & 2c & 2c & 2c & 2c & 2c & 2c & 2c \\ 3c & 3c & 3c & 3c & 3c & 3c & 3c & 3c \\ 4c & 4c & 4c & 4c & 4c & 4c & 4c & 4c \\ 5c & 5c & 5c & 5c & 5c & 5c & 5c & 5c \\ 6c & 6c & 6c & 6c & 6c & 6c & 6c & 6c \\ 7c & 7c & 7c & 7c & 7c & 7c & 7c & 7c \end{bmatrix}\right) >> 5\right) \quad (1)$$

IV. THE HARDWARE ACCELERATOR ORGANIZATION

In this section, different building blocks in the top level hardware design are first introduced. Subsequently, the data flow for the different intra mode generation in the core processing block is explained. The top level hardware organization of the intra-prediction unit is depicted in Fig. 7. It consists of Neighbor Pixel Buffer block, Prediction Samples Selection block, Control block, and Intra-prediction Core Processing block.

The Neighbor Pixel Buffer, as the name suggests, stores the boundary pixels for the current block. For a 4×4 block, depending upon the decoding order, block position, and encoding conditions, up to 13 boundary pixels (8-bit each) are stored in the buffer, whereas for the 16×16 block, it provides storage registers to holds 33 boundary pixels at most. The valid boundary pixels are provided to the core intra-prediction processing block for computation of prediction samples. The same buffer is also used to store the partial results during the processing for 16×16/8×8, luminance/chrominance DC and plane-modes.

All of the processing to compute any given intra-prediction mode takes place in the intra-prediction core

B. Intra-prediction 16×16 *Luminance, and* 8×8 *Chrominance Blocks*

The H.264/AVC video coding standard supports 4 intra-prediction modes for the 16×16 luminance and 8×8 chrominance block types. The Horizontal, Vertical, and DC modes are similar to those for 4×4 luminance block type and are easy to implement. The fourth intra-prediction mode, also known as the plane-mode, utilizes bilinear function to derive the prediction samples. Since the 8×8 chrominance and 16×16 luminance plane-modes are similar, therefore, we shall only discuss the 8×8 chrominance plane-mode in this paper. The 8×8 chrominance plane-mode algorithm is depicted in Fig.5.

The algorithm computes three variables a, b, and c (Fig. 5: lines 8, 9, and 10) using intermediate results H and V (Fig. 5: lines 13, and 14). We extract the equations for the computation of the H and V variables from the algorithm and propose another set of equations for the H and V (Fig. 6: lines 8, and 9). The proposed modified equation for H can be partially overlapped with data-path for the 4×4 intra-prediction block type. The intermediate results [variables t2, t6, s1, and s8] are computed using common overlapped data-paths. Consequently, the data-path for the computation of H (and also for V) can be shared between the intra 16×16 luminance, 8×8 chrominance plane-mode and the intra 4×4 luminance prediction modes. The original equation for computation of H and V requires 16 arithmetic operations (8 additions + 8 subtractions). [Note: we replace multiplications by shifts and additions.] The proposed equations (Fig. 6: line 8, 11), because of shared data path, requires only 9 operations (vs. 16) to be implemented in hardware for computation of H and V. Similarly, pred[x,y] (Fig. 5: line 5) is modified to extract the constants involved, and are computed once (Fig. 6: variable k in line 16). Finally, these constants (k, b, and c) are used to derive the prediction samples of plane mode using (1).

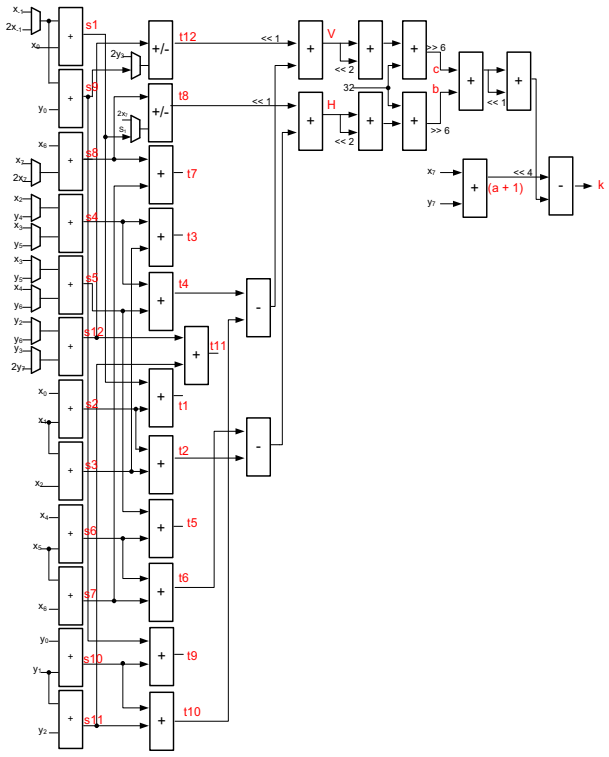

Figure 8. Intra-prediction core processing block.

processing block. The detailed internal design of the intra-prediction core processing block is depicted in Fig. 8. The Vertical and Horizontal intra-prediction modes for all luma / chroma block type (4×4, 8×8 and 16×16) do not require any computation. The boundary pixels for these modes bypass the Intra-prediction core block (not shown in Fig. 8). In case of 4×4 luma intra-prediction mode 3 to 8, all the corresponding unique prediction samples are generated in parallel in 1 clock cycle. Similarly, in case of 4×4 luma mode-2 (DC mode), the prediction sample (DC value)is computed in 2 clock cycles (worst case scenario: For luma 4×4 prediction modes). The core processing block computes the unique sample values only. The distribution of these prediction samples to form a 4×4 prediction block is performed in the Prediction Samples Selection block. In case of the plane-mode, the core processing block computes the constants (b, c, and k) and the corresponding offset values for b and c (nb, nc) in 6 clock cycles. Using these constants, while computing 16 prediction samples in each cycle, all 64 prediction samples for the plane-mode, intra 8×8 chroma predicted block are generated in subsequent 4 clock cycles. Therefore, it takes just 10 clock cycles in total to deliver the plane-mode, intra 8×8 chroma predicted block. Similarly 22 clock cycles are required to compute the plane-mode, intra 16×16 luma predicted MB.

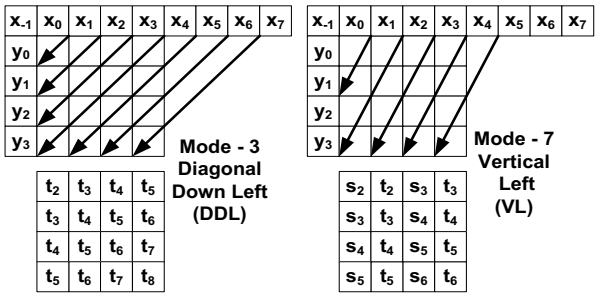

Figure 9. Prediction samples distribution in 4x4 DDL and VL modes.

In most of the 4×4 intra-prediction modes, some of the prediction samples are identical in the final intra-predicted block. Fig. 9 illustrates this fact for DDL and VL intra prediction modes where some of the prediction samples (t_x and s_x in Fig. 9; defined in Fig. 4) are redistributed in the predicted block. Therefore, the Prediction Sample Selection block, for such cases, is responsible for the distribution of the unique prediction samples in the final 4×4 predicted block accordingly and pass it on to the next processing unit in the intra-processing chain of the H.264/AVC decoder.

V. DESIGN EVALUATION

The design proposed in this paper for intra-prediction unit was described in VHDL and synthesized by Synopsys Design Compiler (v2002, rev. 05) for a maximum operating frequency of 150 MHz with 0.18μm CMOS standard cell technology (v1.5). The implementation was verified by simulation results generated using ModelSim 6.5 and comparing with that of the JM13.2 reference software [7].

The number of cycles required to compute any given mode are provided in Table I. The performance comparison with other state-of-the-art intra-prediction hardware designs for H.264/AVC video decoder is also provided in the same table. The comparison suggests that our design provides 50%-75% performance improvement for the Luma 4×4 modes, when compared with [11]. For the Luma 16×16 case, our design provides similar performance when compared with [13].

The proposed hardware design consumes 21K gates. The area cost of our design can not be directly compared with [8] and [13] as these designs are implemented on FPGA. The hardware design in [11] consumes 28.7K gates. Since the proposed design consumes only 21K gates, therefore, it provides 27% area saving when compared with [11].

Table I. Performance Comparison Table

	Mode	Performance [cycles]			
		[8]	[11]	[13]	Proposed
Luma 4x4	VT	17	4	17	1
	HZ	17	4	17	1
	DC	21	4	20	2
	DDL	24	4	18	1
	DDR	24	4	18	1
	VR	23	4	19	1
	HD	23	4	19	1
	VL	22	4	19	1
	HU	20	4	18	1
Luma 16x16	VT	-	66	17	16
	HZ	-	66	17	16
	DC	-	66	20	22
	Plane	340	66	20	22

The area cost is reduced because of implementation of optimized intra-prediction algorithm with significantly reduced number of additions operations (27-60% reduction). Moreover, the configurable design with common data paths for processing of 4×4, 8×8, and 16×16 block types also help to reduce the area cost.

VI. CONCLUSION

A high-throughput and area-efficient hardware design for the intra-prediction unit in the H.264/AVC decoder is proposed in this paper. The optimization of algorithm for 4×4 intra prediction modes enabled us to reduce the number of addition by 60% when compared with that of standard prediction equations. Similarly, overlapped data-paths for the 4×4 , 8×8, and 16×16 block types also helped to greatly reduce the area cost for hardware implementation of intra prediction algorithm. With an operating frequency of 150MHz, the proposed intra prediction unit can easily meet the real time processing requirement of HDTV video frame formats.

REFERENCES

[1] ITU-T Rec. H.264 and ISO/IEC 14496-10:2005 (E) (MPEG-4 AVC). "Advanced Video Coding for Generic Audiovisual Services", 2005.

[2] ISO/IEC JTC1/SC29/WG11. "Report of the Formal Verification Tests on H.264/AVC". In *doc. N6231, Waikoloa, USA*, 2003.

[3] G. Sullivan, P. Topiwala, and A. Luthra. "The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions". In *SPIE Conference On Applications of Digital Image Processing*, vol. 558 pp. 454–474. 2004.

[4] J. Ostermann, J. Bormans, P. List, D. Maroe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi. "Video Coding with H.264/AVC: Tools, Performance and Complexity". In *IEEE Circuit and Systems Magazine*, vol. 4 pp. 7–28. 2004.

[5] G.L. Foresti, P. Mahonen, and C.S. Regazzoni. "Multimedia Video-Based Surveillance Systems: Requirements, Issues, and Solutions". In *Kluwer Academic Publishers*, ISBN 0-7923-7927-6, 2000.

[6] 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". In *IEEE Transactions on Circuit and Systems for Video Technology (CSVT)*, vol. 15, no. 3, pp. 378–401, 2005.

[7] Reference Software for H.264 codec JM 13.2. http://iphome.hhi.de/suehring/tml/index.htm.

[8] E. Sahin, and I. Hamzaoglu. "An Efficient Intra Prediction Hardware Architecture for H.264 Video Decoding". In *Euromicro Conference on Digital System Design Architectures, Methods and Tool*, pp. 448–454, 2007.

[9] C. W. Ku, C. C. Cheng, G. S. Yu, M. C. Tsai, and T. S. Chang. "A High-Definition H.264/AVC Intra-Fame Codec IP for Digital Video and Still Camera Applications". In *IEEE Transactions on Circuit and Systems for Video Technology (CSVT)*, vol. 16, no. 8, pp. 917–928, 2006.

[10] W. T. Staehler. E. A. Berriel, A. A. Susin, and S. Bampi. "Architecture of an HDTV Intra-frame Predictor for H.264 Decoder". In *IFIP International Conference on Very Large Scale Integration*, pp. 228–232, 2006.

[11] X. He, D. Zhou, J. Zhou, and S. Goto. "A New Architecture for High Performance Intra Prediction in H.264 Decoder". In *International Symposium on Intelligent Processing and Communication Systems*, pp. 41–44, 2009.

[12] M. Shafique, L. Bauer, and J. Henkel. "A Parallel Approach for High Performance Hardware Design of Intra Prediction in H.264/AVC Video Codec". In *Design, Automation and Test in Europe Conference* (DATE), pp. 1434–1439, 2009.

[13] X. Wang, X. Cui, and D. Yu. "A Parallel Prediction Architecture for H.264 Video Decoding". In *International Conference on ASIC*(ASICON), pp. 859–862, 2009.