

Intra Prediction for the Hardware H.264/AVC High Profile Encoder

Mikołaj Roszkowski · Grzegorz Pastuszak

Received: 6 December 2012 / Revised: 25 June 2013 / Accepted: 27 June 2013 / Published online: 23 July 2013
© The Author(s) 2013. This article is published with open access at Springerlink.com

Abstract The hardware implementation of the intra prediction described in this paper allows the H.264/AVC encoder to achieve optimal compression efficiency in real-time conditions. The architecture has some features that distinguish it from other solutions described in literature. Firstly, the architecture supports all intra prediction modes defined in High Profile of the H.264/AVC standard for all chroma formats. Secondly, the architecture can generate predictions for several quantization parameters. Thirdly, the hardware cost is reduced as the same resources are used to compute prediction samples for all the modes. Fourthly, the high sample-generation rate enables the encoder to achieve high throughputs. Fifthly, 4×4 block reordering and interleaving with other modes minimize the impact of the long-delay reconstruction loop on the encoder throughput. The architecture is verified against the JM.12 reference model and within the real-time FPGA hardware encoder. The synthesis results show that the design can operate at 100 MHz and 200 MHz for FPGA Aria II and 0.13 μm TSMC technology, respectively. These frequencies allow the encoder to support 720p and 1080p video at 30 fps.

Keywords Video coding · Intra prediction · H.264/AVC · FPGA · Very large-scale integration (VLSI)

1 Introduction

The H.264/AVC standard [1] allows more compression-efficient coding compared to its predecessors. The main coding improvement stems from the advanced prediction techniques. Particularly, some intra prediction methods are introduced. They apply the extrapolation of reconstructed

pixel samples from neighboring blocks to approximate actual pixel samples from a given block. Architectures of H.264/AVC encoders must support at least the DC intra prediction for both luma and chroma. However, the best compression efficiency can be achieved when all predictions are checked in terms of the rate-distortion (RD) criterion. This leads to a high computational burden of video encoders. As many multimedia applications require real-time coding, hardware acceleration of video encoders is indispensable, especially for High Definition Television (HDTV).

There are some challenges in design of intra prediction architectures. Firstly, the selected prediction mode should be optimal or near optimal. Secondly, the consumption of hardware resources must be as small as possible. Thirdly, the latency between processing of successive blocks must be minimized to achieve high-speed encoding. The last problem addresses the throughput of both the intra predictor and the encoder reconstruction loop. In particular, prediction samples within a block can be computed only when reconstructed samples of blocks adjacent to the left and top side are ready for the reference.

In literature, there are some architectures of the H.264/AVC encoder intra prediction [2–14]. However, none of them supports all intra prediction modes available in High Profile. Particularly, they do not support either 8×8 or Plane predictions, and some designs are dedicated only for a selected block size [10, 11]. The architectures apply separate design units to generate prediction for blocks of different sizes, which increases resource consumption. Although view-parallel MB-interleaved scheduling and open-closed loop intra prediction decomposed into two pipeline stages [14] enable a high processing speed, they have some limitations. Particularly, the first method is dedicated for multi-view or B frames, whereas the second involves quality losses for blocks predicted from original pixels. Intra-frame encoder chips described in [2–4, 6–8, 10, 12, 14] achieve high processing speed. However, they apply basic cost functions based on transformed residuals

M. Roszkowski · G. Pastuszak (✉)
Institute of Radioelectronics, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: G.Pastuszak@ire.pw.edu.pl

for mode selection, and their dataflow is optimized only for intra modes. The advanced mode selection based on actual rates (estimated according to binarization schemas of quantized coefficients and associated control data) and distortions (estimated as squared errors of reconstructed samples) would deteriorate significantly the throughput, especially when the reconstruction loop is shared with Inter modes. This limitation is caused by the fact that all prediction modes must be processed by the long-delay path including transformations, the (de)quantization, the reconstruction, the error estimation, and the cost evaluation.

In this paper, the hardware architecture of intra predictor for the encoder is proposed. The design supports all prediction modes for all chroma formats. 4×4 and 8×8 modes can be computed for several quantization parameters. This feature allows more efficient compression when the advanced mode selection based on the rate-distortion criterion is used. The same logic is used for the generation of all prediction modes, leading to the reduction of hardware resources. The architecture applies three main techniques in order to maximize the processing speed. Firstly, the 16 prediction samples can be generated in one clock cycle. Secondly, predictions for 4×4 blocks are interleaved with remaining intra modes. Thirdly, the generation order of 4×4 blocks is modified to decrease the impact of their prediction dependencies on the latency. The obtained throughput allows the encoder to support 1080p 30 fps video at 200 MHz when intra and inter modes are checked based on the rate-distortion analysis. The following sections describe the proposed architecture, and the last one provides implementations results.

2 Architecture Design

The 4×4 block processing order defined by the H.264/AVC is modified to start prediction of the next block before the reconstruction of the previous one is available. The 8×8 , 16×16 , and chroma prediction modes are fully independent of 4×4 ones. Therefore, it is possible to calculate them while waiting for the reconstruction of the previously-processed 4×4 block. In the proposed module, predictions of different types are scheduled to minimize waiting for reconstructed samples when the reconstruction loop delay is long. Final block scheduling used in the design is as follows: B4(0)→B8(0)→B4(1)→L16(H)→L16(V)→B4(2)→B4(4)→B8(1)→B4(3)→B4(5)→L16(DC)→B4(8)→B4(6)→L16(PL)→B4(9)→B4(7)→C(H)→C(V)→B4(10)→B4(12)→B8(2)→B4(11)→B4(13)→C(DC)→C(PL)→B4(14)→B8(3)→B4(15); Note that B4, B8, L16, and C labels correspond to 4×4 , 8×8 , 16×16 , and chroma blocks. Parentheses contain block number or prediction mode (H-Horizontal, V-Vertical, DC-DC, and PL-plane). If the encoder checks several QPs, the generation of 4×4 and 8×8 partitions is repeated for each QP.

Figure 1 presents the block diagram of the intra predictor with distinguished buffers. The buffers include two sets of registers and the on-chip dual-port RAM module. The first set keeps 25 reference samples from left and upper macroblocks. Additionally, there are three corner registers for each color component. They prevent the loss of corner samples when reconstructed data for the left macroblock are written into the memory. The memory keeps reconstructed samples neighboring with the currently processed macroblock and inside the macroblock for up to seven QP values. The raster order of macroblocks involves keeping the whole picture line in the RAM to provide reconstructed samples from upper macroblocks. In particular, most of the memory space (6 KB) is assigned to the picture line including three color components (2 KB for each). Since both 4×4 and 8×8 predictions are computed in the interleaved manner, reconstructed samples for the two modes (and seven QPs) must be stored. They are kept in an additional 2 KB memory space. Whenever the next 4×4 or 8×8 block is started, samples neighboring with the block are fetched from the RAM and stored in registers called *LEFT* and *UPPER*, as shown in Fig. 1. In order to achieve fast transfer, four adjacent samples can be read and written through memory ports concurrently.

In the case of the 8×8 prediction modes, the first step is prefiltering of neighboring samples, which is performed using the intra prediction core. Samples needed to calculate the next prediction mode, or those that should be prefiltered, are selected and loaded into nine intermediate-level registers. The in-depth study of the standard leads to the conclusion that nine neighboring samples are enough to calculate prediction for all 4×4 and 8×8 modes with the exception of two Intra 8×8 directions (i.e. *horizontal down* and *vertical right*). For these cases, one additional prediction value should be computed. This is performed by buffering a prediction value in an auxiliary register (not shown in Fig. 1 for the simplicity) as the missing value is identical to some values calculated for the previous 4×4 block within the 8×8 one.

The prediction core consists of the two levels of adders and multiplexers. The first level of adders is responsible for computations according to Eq. (1), whereas the second level supports Eq. (2). As the result of the calculation, 15 different prediction values are obtained, out of which only up to 10 are valid for a 4×4 block in a given mode. These 10 are selected by the output multiplexer (MUX).

$$\text{pred}[x, y] = (p[x_1, y_1] + p[x_2, y_2] + 1) \gg 1 \quad (1)$$

$$\begin{aligned} \text{pred}[x, y] = & (p[x_1, y_1] + 2 \times p[x_2, y_2] + \\ & + p[x_3, y_3] + 2) \gg 2 \end{aligned} \quad (2)$$

The DC mode requires the reconfiguration of the adder structure, which is accomplished by multiplexers colored

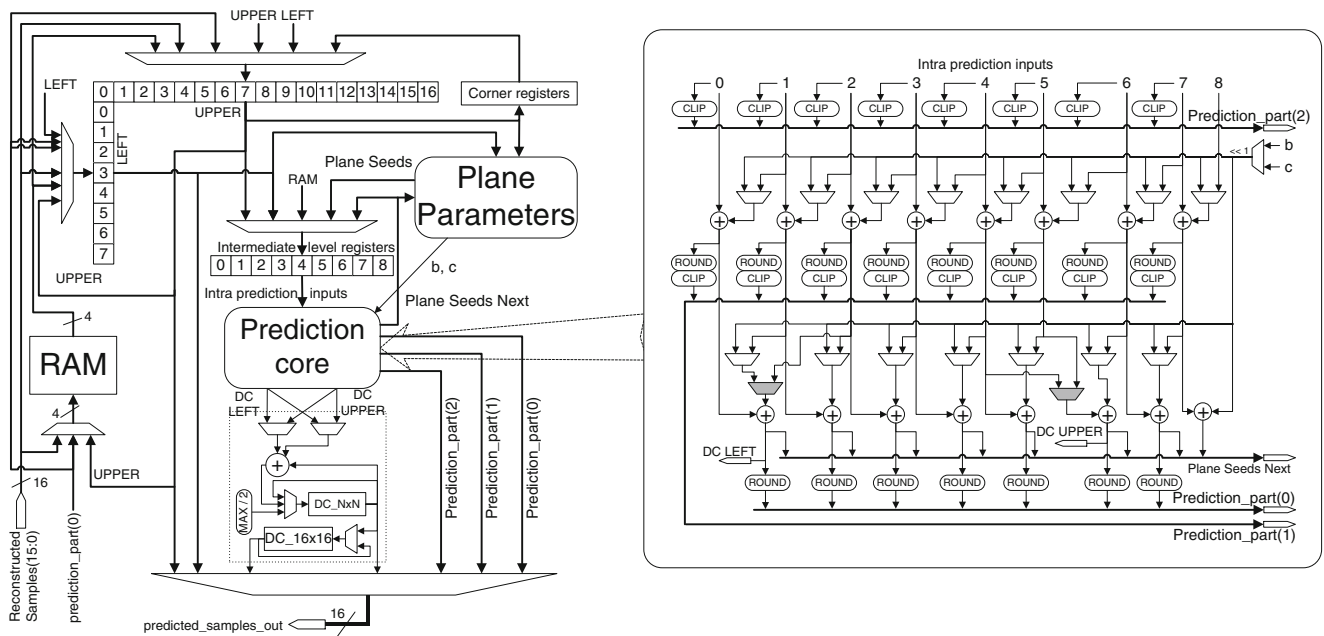


Figure 1 Intra prediction architecture.

dark grey. Particularly, sums of sample pairs obtained at the first level are directed to adders at the second level to compute two sums of four samples. The two sums correspond to the left and upper side. The new configuration, together with the extra adder (outside the core), allows the calculation of the prediction of the whole 4×4 block in one clock cycle. As a consequence, the prediction for 4×4, 8×8, and 16×16 DC modes takes one, two, and four clock cycles, respectively. The results are written or accumulated in the DC_N×N register. An additional register is used to avoid overwriting the 16×16 mode result before releasing prediction samples.

3 Plane Parameter Generator

The plane mode is described by the following equations:

$$H = \sum_{x'=0}^7 (x' + 1) \cdot (p[8 + x', -1] - p[6 - x', -1]) \quad (3)$$

$$V = \sum_{y'=0}^7 (y' + 1) \cdot (p[-1, 8 + y'] - p[-1, 6 - y']) \quad (4)$$

$$a = 16 \cdot (p[-1, 15] + p[15, -1]) \quad (5)$$

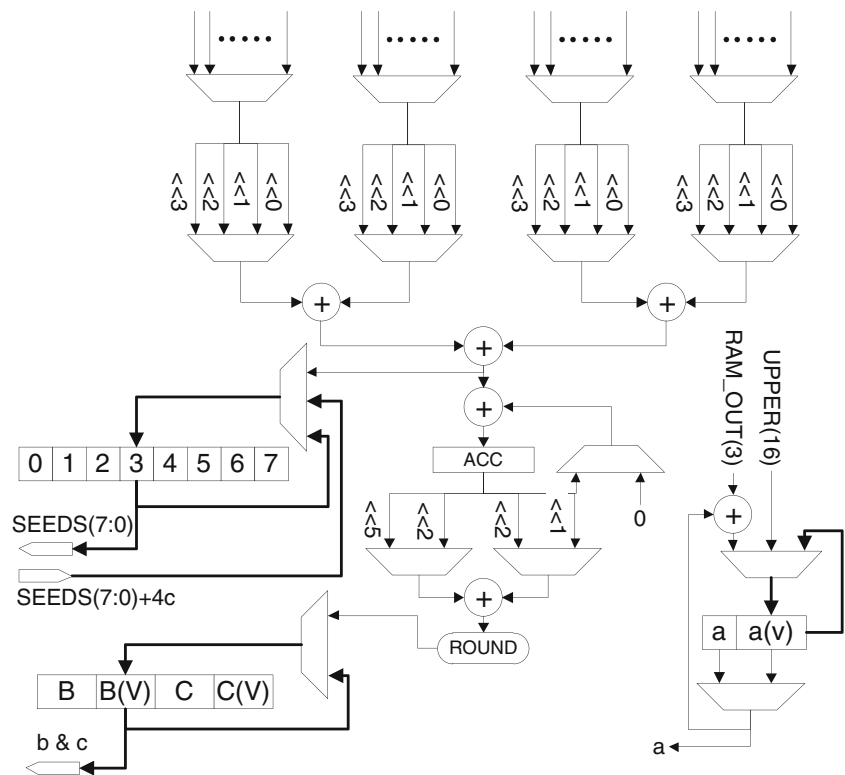
$$b = (5 \cdot H + 32) \ggg 6 \quad (6)$$

$$c = (5 \cdot V + 32) \ggg 6 \quad (7)$$

$$\text{pred}[x, y] = \text{Clip}(a + b \cdot (x - 7) + c \cdot (y - 7) + 16) \ggg 5 \quad (8)$$

Plane prediction mode parameters H , V , a , b , and c are calculated in a separate sub-module in parallel with the calculation of 16×16 luma (or chroma) vertical and horizontal prediction modes. This allows a significant complexity reduction of the calculations of plane mode parameters as the multiplication that appears in (3) and (4) can be replaced by the series of shifts, additions, and accumulations. The sub-module is shown in Fig. 2. It can compute the sum of four selected shifted values in one clock cycle. The shifting and the selection is controlled by a dedicated Plane-mode Finite State Machine (FSM) synchronized with the main FSM of the intra predictor. The shifting replaces the multiplication by powers of two, and the addition of results for different shifts enables the multiplication by all required values. The selection using input multiplexers is performed on the set of reconstructed samples of adjacent macroblocks. They are kept in the corner, left, and upper registers. The addition result can be either accumulated (ACC) or written to seed registers (seeds are plane-predicted samples before clipping). Accumulated values H and V are multiplied by 5 by adding shifted values according to Eqs. (6) and (7). If a subsampled chroma is processed in a given dimension, the accumulated value is multiplied by 34 (the equations are modified according to the H.264/AVC specification). The result is rounded and directed to an appropriate register (b or c). A separate circuit is employed to compute the a value according to (5). The circuit accumulates only two samples when they are available. Values a , b , and c are computed and registered in two versions for both chroma components, and they can also be selected by

Figure 2 Architecture of the plane parameter generator.



input multiplexers to compute seeds. Eight seeds for the first block (upper-most 4×4 block) are computed in the plane parameter generator (see Fig. 2). In order to calculate the prediction for the whole macroblock, the seeds need to be updated. In particular, the addition of $+4b$ or $+4c$ in successive clock cycles gives seed values of successive 4×4 blocks to the left or bottom side, respectively (see Fig. 3). All these updates, as well as proper prediction values, are calculated by the intra prediction core. Before processing blocks in one row, one clock cycle is utilized to compute seed for the first block in the row below ($+4c$). The result is written back to eight seed registers in the plane parameter generator.

Most multiplexers in the prediction core (white in Fig. 1) are used to reconfigure the core for the plane computations. The plane mode employs eight of nine intermediate-level registers (numbered from 0 to 7 in Fig. 1) to keep seeds for two adjacent columns. The seeds are clipped (clip modules) to limit the range. To get seed values for positions shifted by two to the right, $2b$ parameter value is added to values kept in registers, the results are rounded, and then clipped as well. This way, all prediction values within the 4×4 block are determined. Seeds obtained at the first level (increased by $2b$ to correspond to two right columns) are finally increased by $2b$ in the second level of adders. This operation yields eight seeds in the 4×4 block located to the right. If it is necessary to obtain the seeds located in the block below, b is replaced by c .

4 Implementation Results and Conclusion

The intra prediction unit for the encoder is described using VHDL. The design is validated through the comparison with

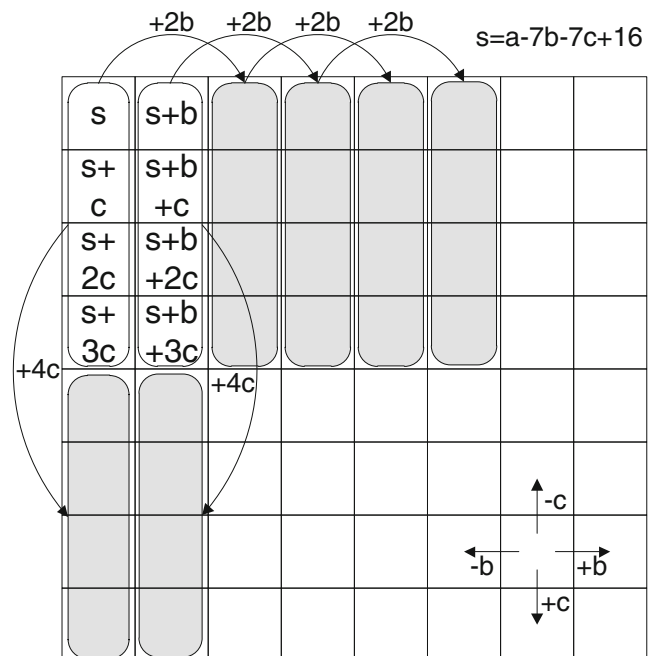


Figure 3 Computations of seeds in the prediction core.

data produced by the JM.12 reference model. The module is synthesized for ASIC and FPGA technologies: TSMC 0.13 μm and Altera Arria II GX. The FPGA synthesis is performed with Altera Quartus II whereas the ASIC synthesis uses Synopsys Design Compiler. The intra prediction module is integrated with other parts of the hardware video encoder, and the whole system operates in the Arria II GX device on a development board. The design can work at the frequency of 100 MHz for the speed grade equal to 5. Smaller speed grades (e.g. 4 for Arria II GX) allow better performance. The implementation in the TSMC 0.13 μm technology allows better performance as the maximal frequency can be set to 200 MHz. Table 1 provides the resource consumption for different configurations of supported modes. Note that luma 16×16 and chroma modes except the Plane prediction are always available. If all modes are supported, the design consumes 23658 gates. The exclusion of particular modes provides moderate savings for the sake of the resource sharing. In particular, the hardware cost of the Plane mode is about 4.8 K. gates. Removing either Intra 4×4 or Intra 8×8 provides smaller savings of gates as these modes shares the same resources. The need for the filtering causes Intra 8×8 requires more resources as compared to Intra 4×4 .

Figure 4 depicts the comparison of the number of clock cycles required for one macroblock versus the delay of the reconstruction loop for three configurations of intra 4×4 and 8×8 modes. Note that plots assume that intra 16×16 and chroma are always processed. Numbers of clock cycles in Figs. 4 and 5 are obtained in successive simulations by the introduction of a constant delay between the intra predictor output and its input for reconstructed samples. If the mode selection is based on SAD and the throughput of the reconstruction loop is 64 samples per clock cycle, the delay is about 10 clock cycles (e.g. 10 pipeline stages: residuals(1), forward transform(2), quantization (2), dequantization(2), inverse transform(2), reconstruction (1)). In this case, the number of clock cycles required for one macroblock is about 340/318 and 448 for the configuration supporting one (either 4×4 or 8×8) and both block sizes, respectively. The encoder

Table 1 Synthesis results for different configurations.

Configuration			Arria II GX [ALUT]	TSMC 0.13 μm [gate]
4×4	8×8	Plane		
×	×	×	4920	23658
	×	×	4640	22980
×		×	4148	21503
		×	3264	19950
×	×		4131	18871
			2449	12333

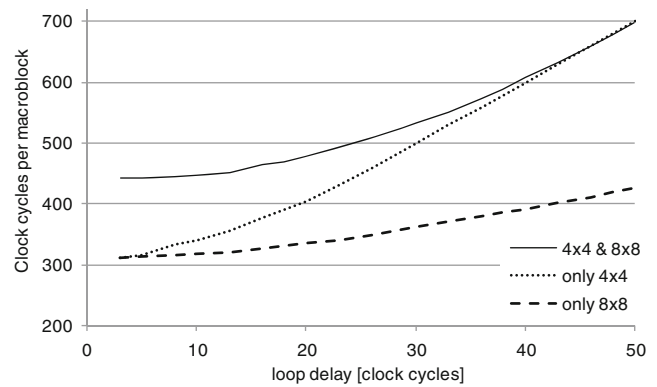


Figure 4 Number of clock cycles per macroblock vs. the delay of the reconstruction loop for different mode configurations.

used for validation employs the advanced mode selection based on the rate-distortion criterion. All inter and intra modes are multiplexed in the common reconstruction loop [15]. It is measured that in such conditions the delay is equal to 40, on average. The corresponding number of clock cycles per macroblock increases to about 392, 599, and 608 for intra 8×8 , intra 4×4 , and both, respectively. Figure 5 depicts the number of clock cycles required for one macroblock versus the delay of the reconstruction loop for checking different QPs. If their number increases, the impact of the delay becomes smaller, and the main limitation results from the generation rate of the intra predictor. The module can also support smaller resolutions than 1080p 30 fps. If the generation rate were eight samples per clock cycles, 384 additional clock cycles would be needed to produce all predictions. In general, the reordering and the interleaving enable the reduction of the number of clock cycles in dependence on the delay of the reconstruction loop. If the delay is close to zero, the gain is small. The reordering removes six of 16 iterations of the reconstruction loop for 4×4 . For delays greater than 40, the interleaving reduces the number of clock cycles close to that needed only for 4×4 blocks (see Fig. 4).

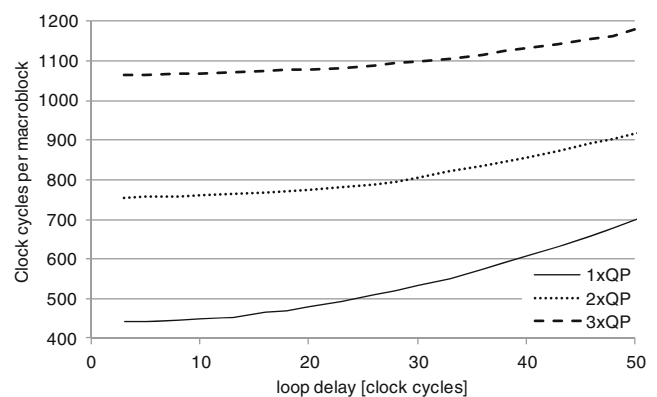


Figure 5 Number of clock cycles per macroblock vs. the delay of the reconstruction loop for different numbers of QP. All modes are supported.

Table 2 Comparison of encoder intra predictors for ASIC technologies.

Design	[2]	[3]	[4]	[5]	[6]	[7]	Proposed
Technology [μm]	TSMC 0.13	0.18	Dongbu 0.13	TSMC 0.18	TSMC 0.18	TSMC 0.13	TSMC 0.13
Clock [MHz]	140	100	116	157	150	114	200
Gate count [K]	18.02	17.193	15.35	28.513	45.7	29.3	23.658
Memory	Off-chip	Off-chip	Off-chip	Off-chip	5.4 KB	Off-chip	8 KB
Video format	1080p@30fps	1408×960@30fps	1080p@30fps	1080p@50fps	1080p@61fps	1080p@30fps	1080p@30fps
Plane Mode	No	Yes	No	Yes	Yes	Yes	Yes
Intra 8×8	No	No	No	No	No	No	Yes
Sample Rate	8	8	8	10	4/16	16	16
Cycle/MB	584	556	464 (max)	256	300 (max)	464(max)	448/608

The comparison of the synthesis results of the designed module with other works is presented in Table 2. The comparison shows that the proposed intra predictor is slightly larger than some other designs [2–4]. However, the proposed supports both the Plane mode and intra 8×8 whereas other designs support only one [3, 5–7] or neither of them [2, 4]. Moreover, the proposed architecture supports all chroma formats and multi-QP coding for 4×4 and 8×8 modes. Its sample-generation rate is equal to 16, which is the highest among the designs. Figure 4 shows that if we remove either 4×4 or 8×8 intra predictions as in [5, 6], the number of clock cycles per macroblock for low-delay reconstructions loop decreases close to the state-of-the-art. The architecture described in [9] needs only 160 cycles per macroblock. However, the design support only some prediction modes (all 8×8, three 16×16, and one chroma). Although the proposed architecture can work at 200 MHz, 1080p@30fps sequences can be encoded at lower frequencies. In particular, 448 (rate-distortion off) and 608 (rate-distortion on) clock cycles correspond to 110 and 168 MHz. In practice, it is indispensable to apply a little higher frequencies to take into account encoder-level control and inter mode processing. Apart from [6], the reference designs cooperate with off-chip memories to save pixel line from the upper macroblocks, and some of them [2–4] assume the controller external to the intra predictor. These features involve the use of additional resources not taken into account in Table 2.

The proposed architecture for the H.264/AVC encoder supports all intra prediction modes specified in High Profile of the H.264/AVC standard. Additionally, it supports coding for several QPs and all chroma formats. The fact that all High Profile intra prediction modes can be generated should result in more efficiently encoded H.264/AVC streams compared to other designs. The architecture is able to generate predictions in real time for 1080p@30fps processing at 200 MHz. The high throughput is achieved through the high degree of the parallelism. Particularly, the architecture calculates a prediction for the whole 4×4 block in one clock cycle for each mode. Moreover, the 4×4 block reordering and the interleaving with other block sizes enable prediction computations in parallel with the long-delay reconstruction loop used for the RD-based mode selection. The design reuses the same combinational unit for processing of 4×4, 8×8, 16×16 modes.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. ITU-T Recommendation H.264 and ISO/IEC 14496–10 MPEG-4 Part 10 (2005) Advanced Video Coding (AVC)

2. Lin, Y.-K., Ku, C.-W., Li, D.-W., & Chang, T.-S. (2009). A 140-mhz 94 k gates hd1080p 30-frames/s intra only profile H.264 encoder. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(3), 432–436.
3. Lin, H.-Y., Wu, K.-H., Liu, B.-D., & Yang, J.-F. (2010). An efficient VLSI architecture for transform-based intra prediction in H.264/AVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(1), 894–906.
4. Jung, J.-S., Jo, Y.-J., & Lee, H.-J. (2011). A fast H.264 intra frame encoder with serialized execution of 4×4 and 16×16 predictions and early termination. *Journal of Signal Processing Systems*, 64(1), 161–175.
5. Hsia, S.-C., Wang, S.-H., & Chou, Y.-C. (2007). A configurable IP for mode decision of H.264/AVC encoder. *Second NASA/ESA conference on adaptive hardware and systems* (pp. 146–152).
6. Diniz, C., Zatt, B., Thiele, C., Susin, A., Bampi, S., & Sampaio, F., et al. (2011). A high throughput H.264/AVC intra-frame encoding loop architecture for HD1080p. *IEEE international symposium on circuits and systems* (pp. 579–582).
7. Kuo, H.-C., Wu, L.-C., Huang, H.-T., Hsu, S.-T., & Lin, Y.-L. (2011). A low-power high-performance H.264/AVC intra-frame encoder for 1080pHD video. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(6), 925–938.
8. Chen, J.-W., Chang, H.-C., Wang, J.-S., & Guo, J.-I. (2011). A dynamic quality-adjustable H.264 intra coder. *IEEE Transactions on Consumer Electronics*, 57(8), 1203–1211.
9. He, G., Zhou, D., Zhou, J., & Goto, S. (2010). Intra prediction architecture for H.264/AVC QFHD encoder. *Picture coding symposium* (pp. 450–453).
10. Shi, Y., Tokumitsu, K., Togawa, N., Yanagisawa, M., & Ohtsuki, T. (2010). VLSI implementation of a fast intra prediction algorithm for H.264/AVC encoding. *IEEE Asia Pacific conference on circuits and systems* (pp. 1139–1142).
11. Kim, T. S., Rhee, C. E., & Lee, H.-J. (2011). Prediction mode reordering and IDCT direction control for fast intra 8×8 prediction. *IEEE International midwest symposium on circuits and systems*.
12. Shafique, M., Bauer, L., & Henkel, J. (2007). An optimized application architecture of the H.264 video encoder for application specific platforms. *IEEE/ACM/IFIP workshop on embedded systems for real-time multimedia* (pp. 119–124).
13. Lin, Y.-K., Li, D.-W., Lin, C.-C., Kuo, T.-Y., Wu, S.-J., & Tai, W.-C., et al. (2008). “A 242mW 10mm² 1080p H.264/AVC high-profile encoder chip”, *IEEE International solid-state circuits conference* (pp. 313–315).
14. Ding, L.-F., Chen, W.-Y., Tsung, P.-K., Chuang, T.-D., Hsiao, P.-H., Chen, Y.-H., et al. (2010). A 212MPixels/s 4096 \times 2160p multiview

video encoder chip for 3D/Quad full HDTV applications. *IEEE Journal of Solid State Circuit*, 45(1), 46–58.

15. Pastuszak, G. (2008). Transforms and quantization in the high-throughput H.264/AVC encoder based on advanced mode selection. *IEEE annual symposium on VLSI* (203–208).



related algorithms for the embedded systems.

Mikolaj Roszkowski received the M.S. degree in radio-communication and multimedia technology from the Warsaw University of Technology in 2009, where he is currently pursuing the Ph.D degree. He is currently a Research Assistant with the Institute of Radioelectronics, Warsaw University of Technology. His current research interests include video and image processing and compression technologies, computer vision and efficient hardware implementations of the



processing and compression, high-performance digital ICs.

Grzegorz Pastuszak received the M.S. degree in microelectronics in 2001 and the Ph.D degree in multimedia technology in 2006, both from Warsaw University of Technologies, Warsaw, Poland. From 2001 to 2002, he was an ASIC designer in FFC, Tokyo, and Fujitsu Devices, Yokohama. Currently, he is with Institute of Radioelectronics Warsaw University of Technology. His areas of interest include VLSI architectures and algorithms, image/video/audio processing and compression, high-performance digital ICs.