

Performance evaluation of a new efficient H.264 intraprediction scheme

Sara HAMDY¹, Mostafa IBRAHIM¹, Mohamed OSMAN^{2,*}

¹Benha Faculty of Engineering, Benha University, Benha, Egypt

²College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt

Received: 30.12.2013

Accepted/Published Online: 01.07.2014

Final Version: 23.03.2016

Abstract: The paper presents a new efficient H.264/AVC 4×4 intraprediction scheme. The new prediction scheme is based on the best prediction matrix mode. The main idea behind the new prediction scheme is to combine the most usable intraprediction modes, {vertical - horizontal - DC}, into a new efficient prediction mode. The new prediction scheme is implemented using VHDL and hence it uses the full advantages of inherent parallelism in the hardware. We evaluate the performance of this prediction scheme in terms of compression ratio, peak signal to noise ratio, and bit rate using seven video sequences. Moreover, we analyze the power consumption, the delay, and FPGA area utilization of the implemented H.264 encoder after utilizing the new prediction scheme. The performance measures as well as the area and power consumption are compared to other best known prediction algorithms.

Key words: Audio video coding, H.264, intraprediction, field programmable gate arrays, compression efficiency

1. Introduction

Video compression systems are used in many commercial products, from consumer electronic devices such as digital camcorders and cellular phones to video teleconferencing systems. These applications make video compression hardware devices an inevitable part of many commercial products. In order to improve the performance of the existing applications, an international standard for video compression, which is named H.264 or MPEG4 Part-10, was developed. This standard significantly improves video compression efficiency [1]. Figure 1 illustrates the main building blocks of the H.264 encoder. It is clear from the block diagram that the video compression efficiency of the H.264 standard is not a result of a single feature, but a combination of a number of encoding subblocks. One of the most important factors of the improved compression efficiency of the H.264 is its intraprediction algorithm [1,2]. The intraprediction algorithm generates a prediction for a macroblock (MB) based on spatial redundancy. The H.264 intraprediction algorithm achieves better coding results than the intraprediction algorithms used in previous video compression standards [2,3].

There are nine prediction modes available for the 4×4 luminance (luma in short) block as shown in Table 1, four modes for the 16×16 luma MB, and four modes for the 8×8 chrominance (chroma in short) blocks to remove spatial redundancy within a frame. The prediction mode for each block that results in minimum difference between macroblock P and the current block is selected. The first three prediction modes (vertical, horizontal, and DC) that are used for encoding intra 4×4 blocks are the most commonly used; collectively they cover 85%–95% of the best modes [4].

*Correspondence: mbakr@ieec.org

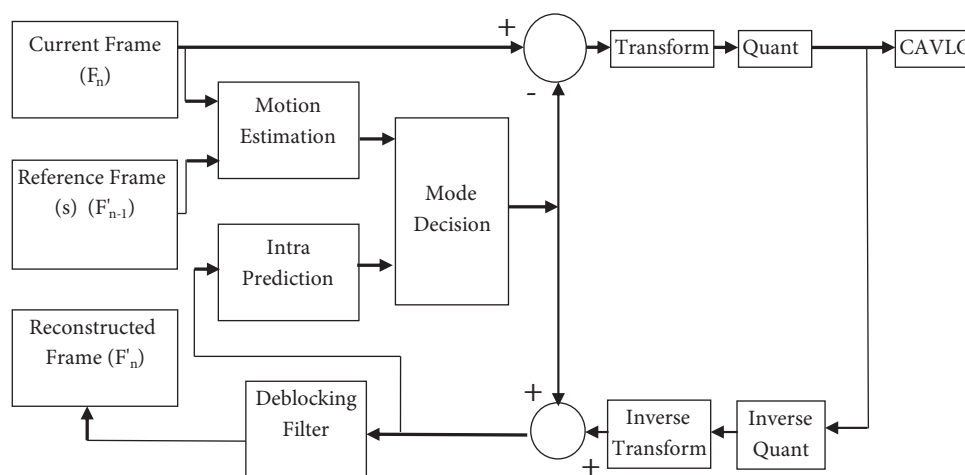


Figure 1. Block diagram of the H.264 encoder [1].

Table 1. Intra 4×4 prediction modes.

Mode #	Mode name	Mode #	Mode name
0	Vertical	1	Horizontal
2	DC	3	Diagonal-down-left
4	Diagonal-down-right	5	Vertical-right
6	Horizontal-down	7	Vertical-left
8	Horizontal-up		

Recently, many researchers aimed to simplify the intraprediction encoding in H.264. On the other hand, fewer researchers are working on improving the encoding efficiency in terms of compression ratio [5,6].

In this paper, the authors are motivated by the current interest in parallel software and hardware realizations of legacy sequential software systems to propose a new efficient mode that combines the vertical, horizontal, and DC modes. The new proposed mode is entitled best prediction matrix mode (BPMM). The performance of the BPMM is verified in terms of compression ratio, bit rate, and PSNR. Furthermore, the efficiency of the BPMM is compared with other best known intraprediction algorithms.

1.1. Related work

In this section, we present the state of the art in the efforts to implement efficient hardware realizations of H.264 encoders.

Walter et al. [7] presented the standard-cells synthesis and comparison of parallel hardware architectures for the sum of absolute differences (SAD) data path, focusing on different design points such as the tradeoff between high-performance and low-power dissipation. Multi- V_{dd} , multi- V_t , and different combinations of parallelism and pipeline architectural techniques were explored in that work. In order to generate the results, they used the IBM 65 nm standard-cells library with typical voltage of 1 V and 1.2 V, and the back-end Cadence tools, e.g., Power Analysis, for the power measurements. They achieved significant power reduction for the architectures with low-frequency and high parallelism, with High- V_t and mainly with only one pipeline stage and small power source. However, the authors did not mention the execution time, area utilization, PSNR, and compression ratio in their results.

Muralidha et al. [8] proposed an intraprediction hardware architecture where it exploits parallelism in predicting the pixels and pipelining is implemented during the calculation of the cost function. The parallelism feature includes an optimized data path, which calculates only 24 unique pixel values, and the former are assigned to the current macroblock depending on the equations for different modes as defined in the H.264 standard. Synthesis results confirmed that the proposed architecture is able to process HD 1080p at 24 fps when operating at 57 MHz for ASIC platforms and is observed to be faster than the previous methods. The main shortcoming of that paper is that the authors did not give the execution time, area utilization, PSNR, bit rate, or compression ratio.

Wei et al. [9] proposed a hardware efficient high definition television (HDTV) encoder for H.264/AVC. They used a two-level mode decision (MD) mechanism to reduce the complexity and maintain the performance, and they designed a sharable architecture for normal mode fractional motion estimation (NFME), special mode fractional motion estimation (SFME), and luma motion compensation (LMC) to decrease the hardware cost. Based on these technologies, a four-stage macroblock pipeline scheme was adopted using an efficient memory management strategy for the system, which greatly reduces on-chip memory and bandwidth requirements. The synthesized results show that the proposed encoder used about 1126 KGates with an average Bjontegaard delta peak signal-to-noise ratio decrease of 0.5 dB (*BD-PSNR* is a metric that allows to compute the average distance between two RD-curves (bit rate/quality)), compared with the JM15.0 reference design. It can fully satisfy real-time video encoding for 1080p at 30 fps of the H.264/AVC high profile. In that paper, the execution time and compression ratio were not given.

Roszkowski et al. [10] proposed hardware implementation of the intraprediction that 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 the literature. First, the architecture supports all intraprediction modes defined in the high profile of the H.264/AVC standard for all chroma formats. Second, the architecture can generate predictions for several quantization parameters. Third, the hardware cost is reduced as the same resources are used to compute prediction samples for all the modes. Fourth, the high sample-generation rate enables the encoder to achieve high throughputs. Fifth, 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 $0.13\mu\text{m}$ TSMC technologies, respectively. These frequencies allow the encoder to support 720p and 1080p video sequence at 30 fps. As noticed in the previous reviewed works, the author in that paper did not mention the execution time, PSNR, bit rate, or compression ratio.

1.2. Paper outline

The rest of the paper is organized as follows: Section 2 gives a theoretical background of the intraprediction in the H.264 encoder. Section 3 illustrates the hardware and software platforms that have been utilized to evaluate the proposed technique. Section 4 shows a comparison between different intraprediction mode selection algorithms to figure out the best algorithm to be used in our experiments in terms of compression ratio, PSNR, and bit rate. Section 5 explains the proposed new intraprediction approach. Section 6 evaluates the performance of the proposed algorithm, with the achieved results of applying the new proposed prediction technique. Moreover, it illustrates the FPGA power consumption and area utilization of the new proposed prediction scheme. Finally, Section 7 summarizes the main contributions of this paper and draws some directions for future work.

2. H 264 intraprediction overview

The main goal behind intraprediction is to achieve compression within a frame. The neighboring pixels within the picture frame tend to have similar values in order to exploit the spatial redundancy; the prediction is done based on the values of reconstructed pixels of the previous subblock [1]. For the luminance layer intraframe prediction, there are two possible block sizes to encode one MB. The first is the I16MB, with four possible prediction modes applied to the whole 16×16 MB. The second is the I4MB, with nine possible prediction modes applied to the sixteen blocks of 4×4 subblocks that compose the MB. For the chrominance layer there are also four possible modes to predict each 8×8 block (Cr and Cb) in the MB [11,12].

The modes of the 4×4 intraprediction are given in Table 1. The values of each 4×4 block of luma samples are predicted from the neighboring pixels above or to the left of a 4×4 block. Modes 0, 1, 3, 4, 5, 6, 7, and 8 are directional ways of performing the prediction that can be selected by the encoder. However, mode 2 is the DC prediction mode with no direction [13].

Figure 2a shows a 4×4 block containing 16 pixels labeled from a through p. A prediction block P is calculated based on the pixels labeled A–M obtained from the neighboring blocks [14]. Figures 2b and 2c show the intra H.264 standard 4×4 vertical and horizontal prediction modes [15]. If the prediction mode chosen is vertical, it means that each pixel value is vertically similar and the vertical edge is more probable than the horizontal edge in the block. In the same manner, if the prediction mode chosen is horizontal, it means that each pixel value is horizontally similar and the horizontal edge is more probable than the vertical edge in the block [16,17]. In the DC prediction mode, all pixels in the current 4×4 block are replaced by the mean value of the neighboring pixels A, B, C, D, I, J, K, and L. The DC prediction is accomplished by using the rules in Figure 2d.

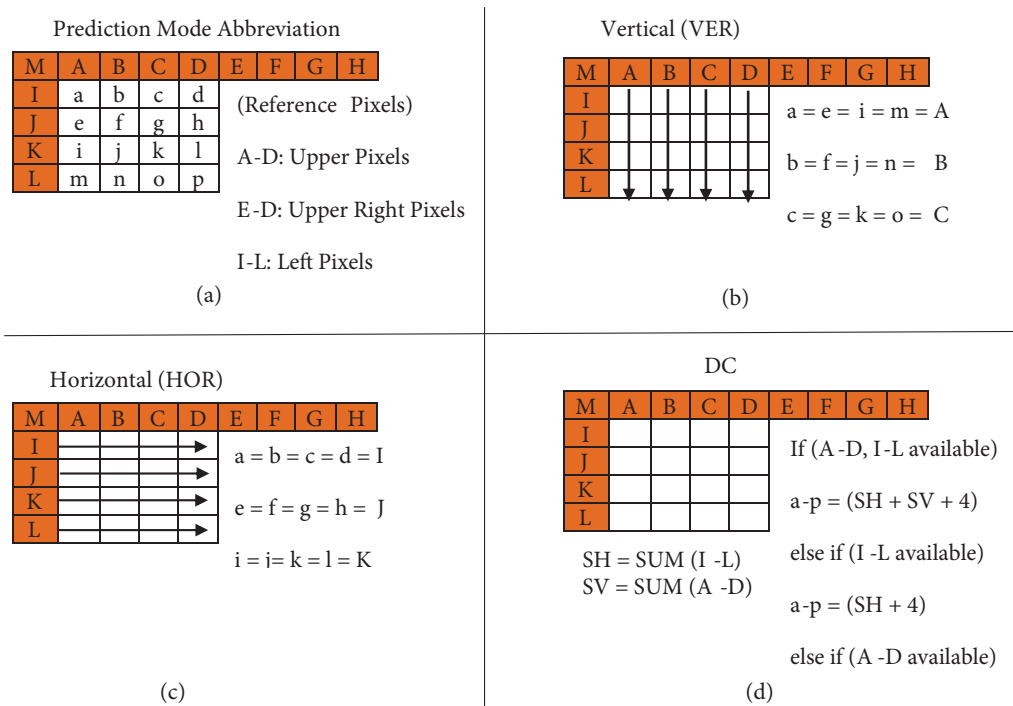


Figure 2. Equations of luma 4×4 mode prediction pixels (vertical, horizontal, DC) [15].

3. Experimental setup

In this section, we present the experimental setup utilized to implement and validate the functionality as well as the performance of the proposed algorithm.

3.1. H 264 encoder testbed

In this section, we illustrate the hardware architecture of the H.264 encoder that has been employed to test the new proposed intraprediction technique. The authors utilize the H.264 encoder testbed of Henson [18] that has been implemented using the VHDL description language. This testbed is designed as a modular system with small, efficient, and low-power components performing well-defined tasks. It exploits the intraprediction but it does not consider the interprediction. It utilizes the SAD algorithm while employing only the vertical, horizontal, and DC prediction modes as motivated by the work in [4].

The VHDL implementation supports input video sequences of CIF format (consisting of 352×288 pixels); however, it can be configured to support any other standard video format. Each frame of the input video sequence is divided into MBs. Each MB consists of $256(16 \times 16)$ pixels while each pixel is represented by 8 bits. Each MB is transferred to the encoder testbed four pixels at a time using the 32-bits-wide bus. The least significant byte of this wide bus is considered as the first pixel. The input video data are read in raster scan order and then each MB is divided into 4×4 sub-MBs.

3.2. Implementation platform

In this paper we use a low-cost FPGA that is suitable for consumer electronics, namely the Xilinx Spartan-3E-1600 board, as the hardware platform for the design. The utilized FPGA consists of 1600 KGates and supports a maximum frequency of 572 MHz [19].

For software, we used the ModelSim SE-6.2 for the functional simulation. In addition, we used the Xilinx ISE-9.2i tool for synthesis, place/route, downloading the design into the target FPGA, and finally area analysis steps. Regarding the power consumption estimations, we used of the XPower estimator provided by Xilinx.

4. Performance evaluation of different intraprediction mode selection algorithms

In this section, we present a comparison between different intraprediction mode selection algorithms, namely SAD [18–20], sum of squared difference (SSD) [21], sum of Manhattan distance (SMD) [22,23], and sum of Hamming distance (SHD) [23]. All of these algorithms utilize the vertical, horizontal, and DC modes. The aforementioned four algorithms are applied to the H.264 encoder testbed of Henson [18]. It worth mentioning here that this testbed does not consider interprediction; hence, no motion estimation process is performed.

The Mother-Daughter CIF video sequence which consists of 300 frames is encoded with the Henson [18] H.264 encoder testbed while changing the intraprediction mode selection algorithm and the quantization parameter (QP). The QP is a critical parameter that determines the quantization step size between successive rescaled values. If the step size is large, the range of quantized values is small and can therefore be efficiently represented and hence highly compressed during transmission, but the rescaled values are a crude approximation of the original signal. If the step size is small, the rescaled values match the original signal more closely, but the larger range of quantized values reduces compression efficiency. Table 2 shows the comparison results between the four previously mentioned intraprediction mode selection algorithms. The QP varies from 20 to 45 with a step size of 5. It is clear that the SAD algorithm outperforms the other algorithms from the PSNR, compression ratio, and bit rate perspectives, and hence we adopt it for our proposed technique.

Table 2. Comparison between different intraprediction algorithms.

Video name	QP	Algorithm	PSNR (dB)	Compression ratio	Bit rate (kbps)
Mother-Daughter	20	SAD	43.8	11	338
		SHD	43.8	10	371.25
		MD	43.8	10	371.25
		SSD	43.7	9	412.5
	25	SAD	40.5	16	232
		SHD	40.4	15	247.5
		MD	40.5	15	247.5
		SSD	40.4	14	265.17
	30	SAD	37.1	25	149
		SHD	36.9	22	168.75
		MD	37	22	168.75
		SSD	36.6	21	176.78
	35	SAD	33.7	35	106
		SHD	33.6	32	116.01
		MD	33.8	31	119.75
		SSD	33.3	30	123.75
	40	SAD	31.4	44	84
		SHD	31	40	92.81
		MD	30.7	42	88.39
		SSD	30.1	39	95.19
45	SAD	27.5	51	73	
	SHD	27.8	45	82.5	
	MD	26.8	49	75.76	
	SSD	26.9	45	82.5	

5. Proposed intraprediction approach

In this section we illustrate the proposed new technique that enhances the compression capability, PSNR, and bit rate of the H.264 encoder. Moreover, the hardware architecture of the proposed technique is presented.

It has been shown from the statistics that the vertical, horizontal, and DC prediction modes are more frequently used than other modes [4]. These modes imply higher correlation between the reference samples and the pixels to be predicted [5]. This inspired the authors to design the BPMM, which combines the vertical, horizontal, and DC modes in a single new intra 4×4 prediction mode.

The best prediction matrix of the BPMM is calculated by tradeoffs between values of the following four matrices: vertical, horizontal, DC, and upper-left corner matrices. Figure 3 shows the construction matrix, which consists of the aforementioned four matrices.

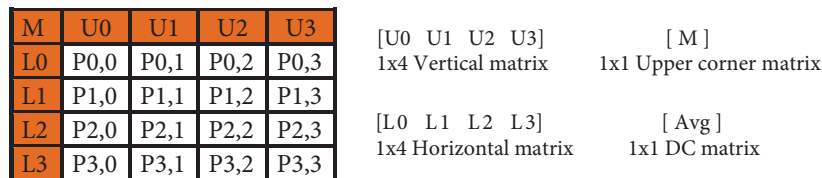


Figure 3. Construction matrices.

Figure 4 illustrates the new proposed intraprediction technique. $p_{i,j}$ denotes the pixel to be predicted in

the i th row and j th column of the current 4×4 block where $0 \leq i, j \leq 3$, and U_j and L_i denote the reference samples reconstructed from upper and left blocks, respectively. M denotes the upper-left corner pixel value, while \mathbf{avg} is calculated as shown in Figure 2d.

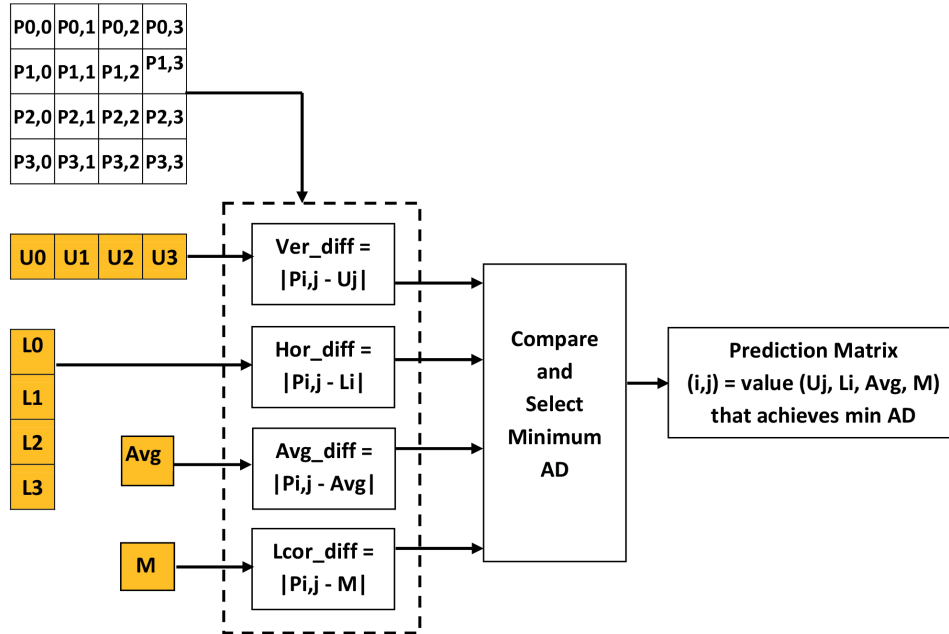


Figure 4. The proposed intraprediction technique.

Figure 5 demonstrates the proposed intraprediction technique by a numerical example while Algorithm 1 explains the new proposed intraprediction scheme. More details about the proposed algorithm can be found in [23].

Algorithm 1. The procedure of generating the best prediction matrix.

- 1: Read original intra 4×4 block;
- 2: Prepare construction matrices;
- 3: Compute the absolute difference as follows:
- 4: **for** $(i, j = 0 : 3)$ **do** \triangleright step = 1
- 5: $Ver_diff = |P_{i,j} - U_j|$; $Hor_diff = |P_{i,j} - L_i|$;
- 6: $Avg_diff = |P_{i,j} - Avg|$; $Lcor_diff = |P_{i,j} - M|$;
- 7: Find the minimum value of the absolute differences;
- 8: Put the value of one of (U_j, L_i, Avg, M) that corresponds to this minimum in the prediction matrix (i,j) .
- 9: **end for**
- 10: Get the best prediction matrix.

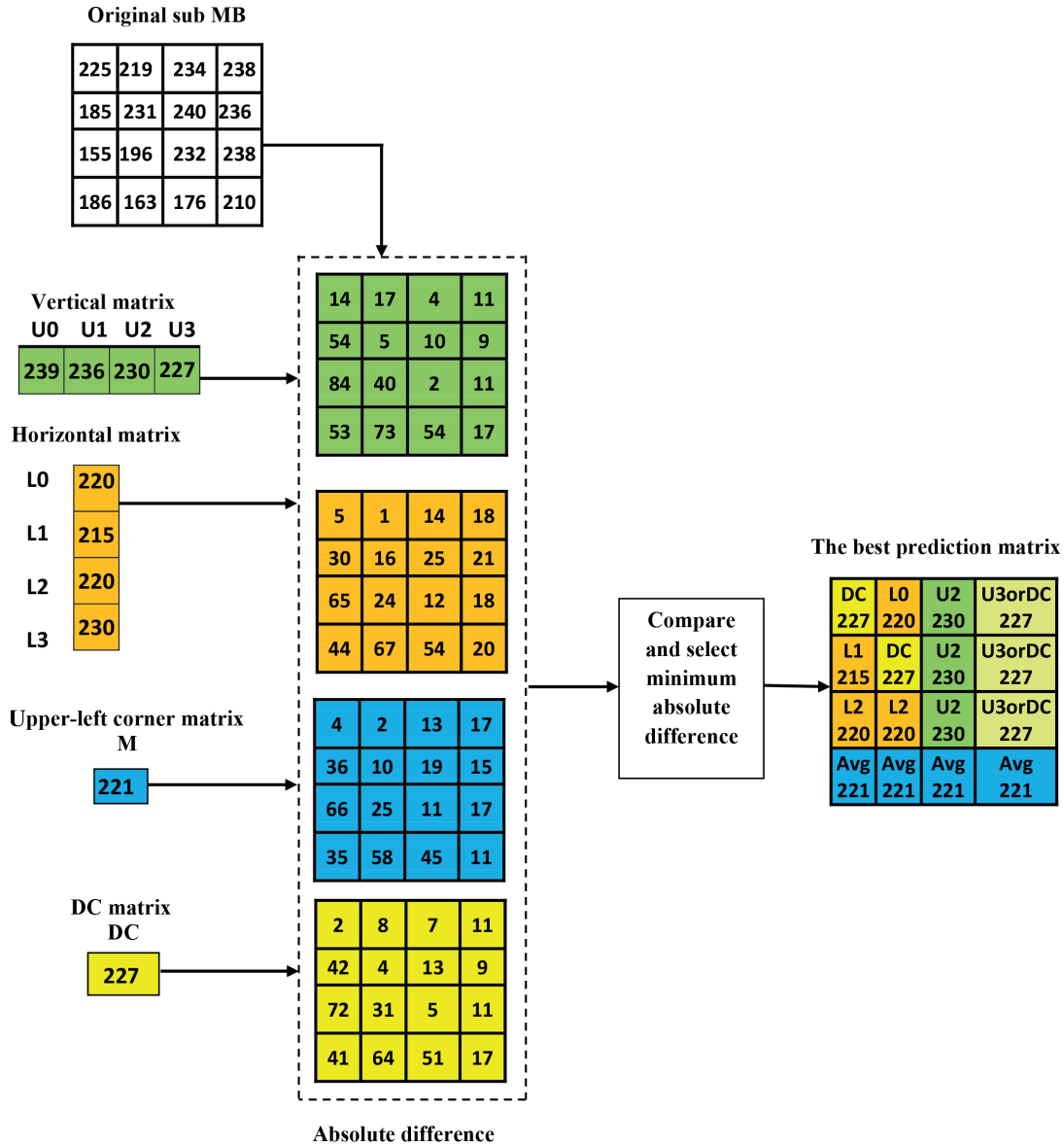


Figure 5. Numerical example of applying the BPMM.

Figure 6 presents the hardware architecture that realizes our proposed intraprediction technique. This architecture is implemented using three main modules, namely intraprediction buffers, absolute difference units, and a comparator. Intraprediction buffer modules include six buffers: one buffer for storing the original MB, another buffer for storing the original 4×4 block, and the four remaining buffers for storing the reconstructed pixel values, which are needed to generate the predicted subblock (upper, left, upper-left corner, and average). The absolute difference module is responsible for calculating the absolute difference values. Finally, the comparator module finds the minimum value of the absolute differences and then selects a value between (upper, left, upper-left corner, and average) that corresponds to this minimum and stores it in the best prediction matrix [24].

In the case that that two minimum values are returned, we give the priority as follows: horizontal, vertical, corner, and then DC.

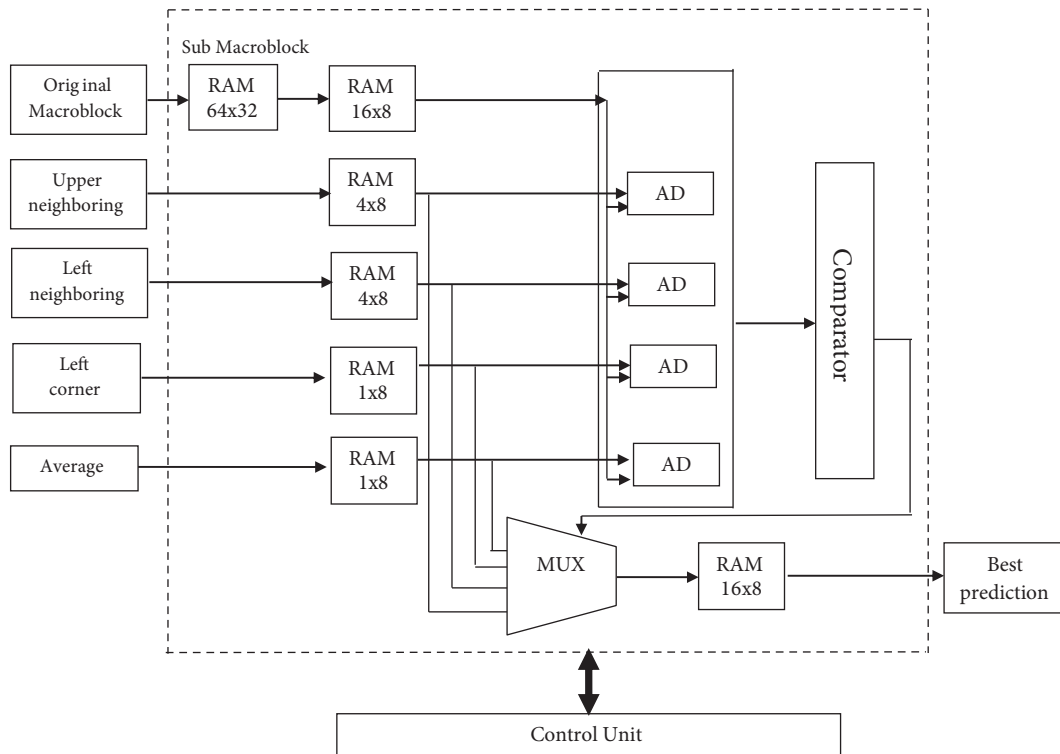


Figure 6. Hardware architecture of the BPMM.

6. Evaluation of the proposed intraprediction scheme

In this section we evaluate the efficiency of the proposed intraprediction scheme from different perspectives. First, we explore the results of evaluating our prediction scheme in terms of compression ratio, PSNR, and bit rates. Second, we introduce a power analysis of the hardware implementation of the H.264 encoder utilizing our BPMM intraprediction scheme. Third, we present FPGA area analysis in terms of memory elements and configurable blocks.

6.1. Performance evaluation of our proposed approach

In this section, we present the experimental results of the new proposed algorithm compared to the Henson implementation [18] that used the SAD algorithm as proved in Section 4. The comparison is based on five CIF video sequences, namely Mother-Daughter, Container, Foreman, Silent, and News, which are composed of 300 frames. The evaluation process is accomplished as follows: each video is encoded with the H.264 intraframe coder while utilizing the new BPMM. The QP varies from 20 to 45 with a step size of 5

Figure 7 shows the evaluation results of the new algorithm compared to the SAD using the Mother-Daughter CIF video sequence. The results show that the proposed prediction technique enhances the compression ratio on average by 28.24% and consequentially the bit rate decreases on average by 22.5% while the PSNR is slightly increased on average by 0.6dB for the Mother-Daughter CIF video.

Seeking for a much more accurate judgment of the new proposed prediction technique, we test it with the Container, Foreman, Silent, and News CIF video sequences as shown in Figures 8–11, respectively. The results indicate that it enhances the compression ratio on average by 13.3% and correspondingly the bit rate decreases on average by 13.14% while the PSNR is slightly increased on average by 0.116 dB for the Container CIF video.

Regarding the Foreman CIF video, the results indicate that the new prediction scheme enhances the compression ratio on average by 34.86% and correspondingly the bit rate decreases on average by 25.51% while the PSNR is slightly increased on average by 0.5 dB. For the Silent CIF video the results indicate that it enhances the compression ratio on average by 34.5% and correspondingly the bit rate decreases on average by 25.3% while the PSNR is slightly increased on average by 0.533 dB. Regarding the News CIF video, the results indicate that the new prediction scheme enhances the compression ratio on average by 16.2% and correspondingly the bit rate decreases on average by 14% while the PSNR is slightly increased on average by 0.033 dB.

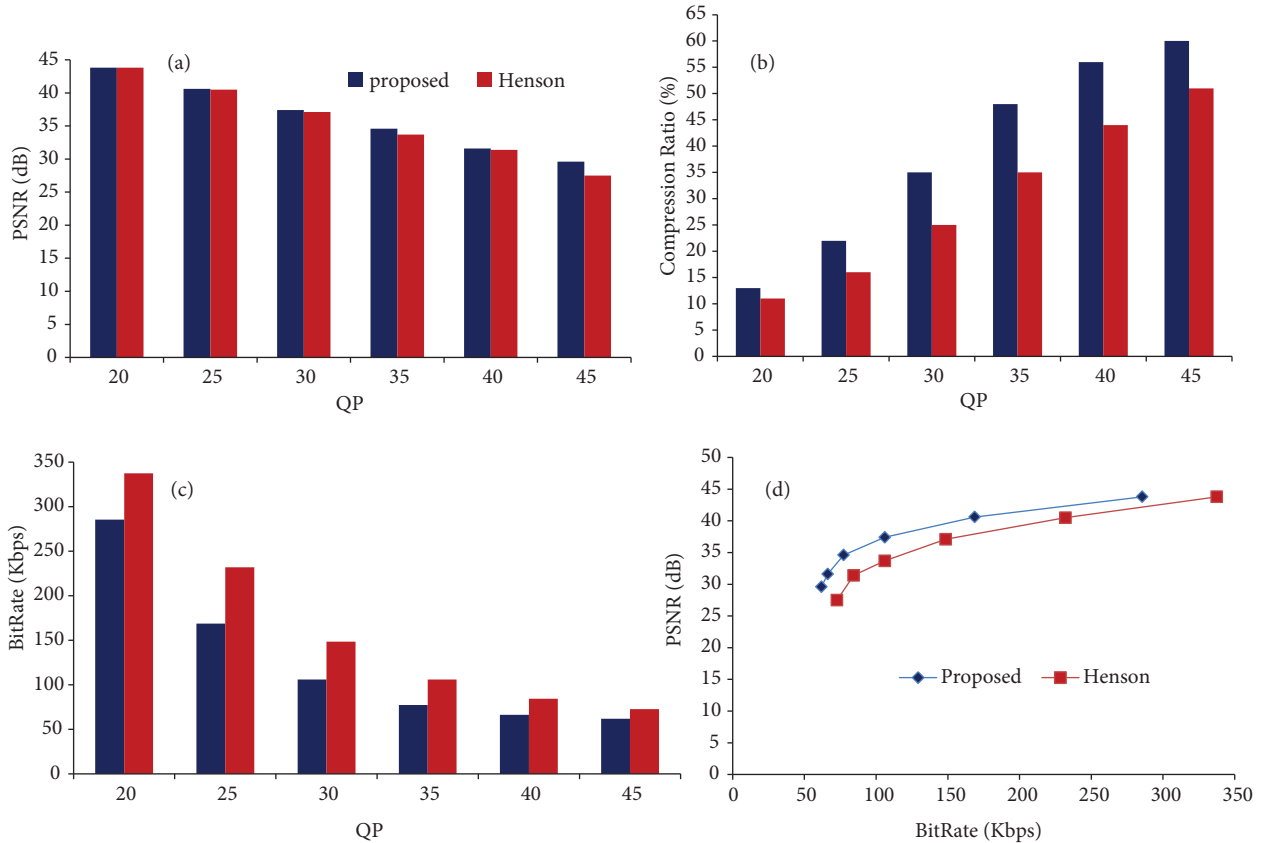


Figure 7. Evaluation of BPMM using Mother-Daughter CIF video.

Table 3 shows the experimental results of the BPMM; we achieve a better compression ratio by an average of 25.42%, bit rate is decreased by an average of 20.09% and PSNR is increased by an average of 0.365 dB for the five CIF videos and over all the QP values under test. The previously mentioned results have been achieved while the execution time is slightly increased on average by 0.0475% for the five CIF videos.

Table 3. Experimental results of BPMM compared to Henson [18].

	PSNR (dB)	Compression ratio increase %	Bit rate increase %	Time increase %
Mother-Daughter	+0.6	28.24	22.5	-0.0015
Container	+0.116	13.3	13.14	-0.052
Foreman	+0.5	34.86	25.51	+0.1
Silent	+0.533	34.5	25.3	-0.219
News	+0.033	16.2	14	+0.22
Average	+0.356	25.42	20.09	+0.0475

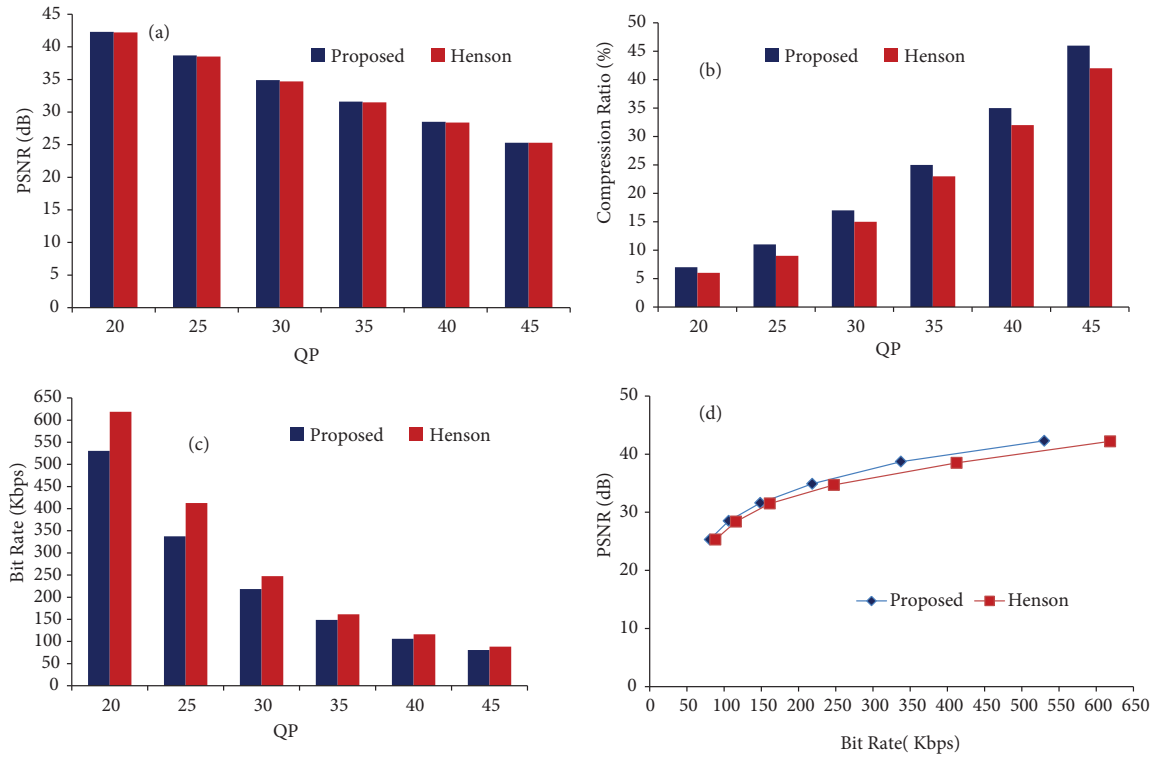


Figure 8. Evaluation of BPMM using Container CIF video.

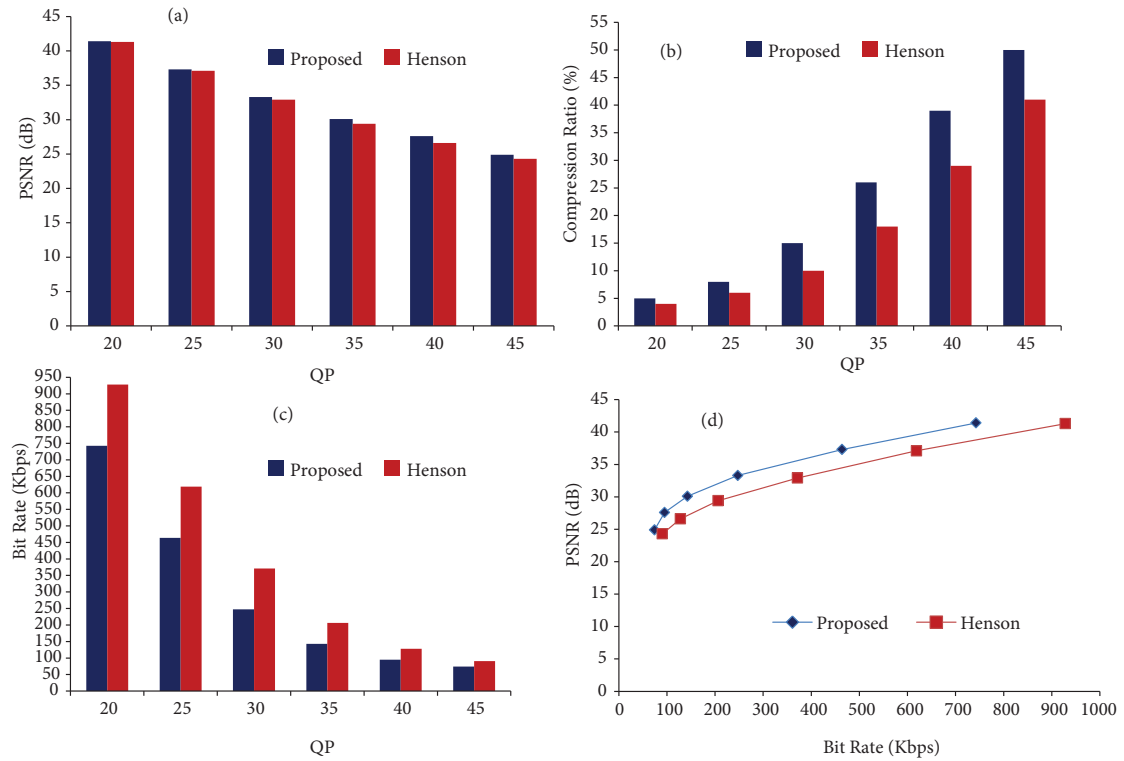


Figure 9. Evaluation of BPMM using Foreman CIF video.

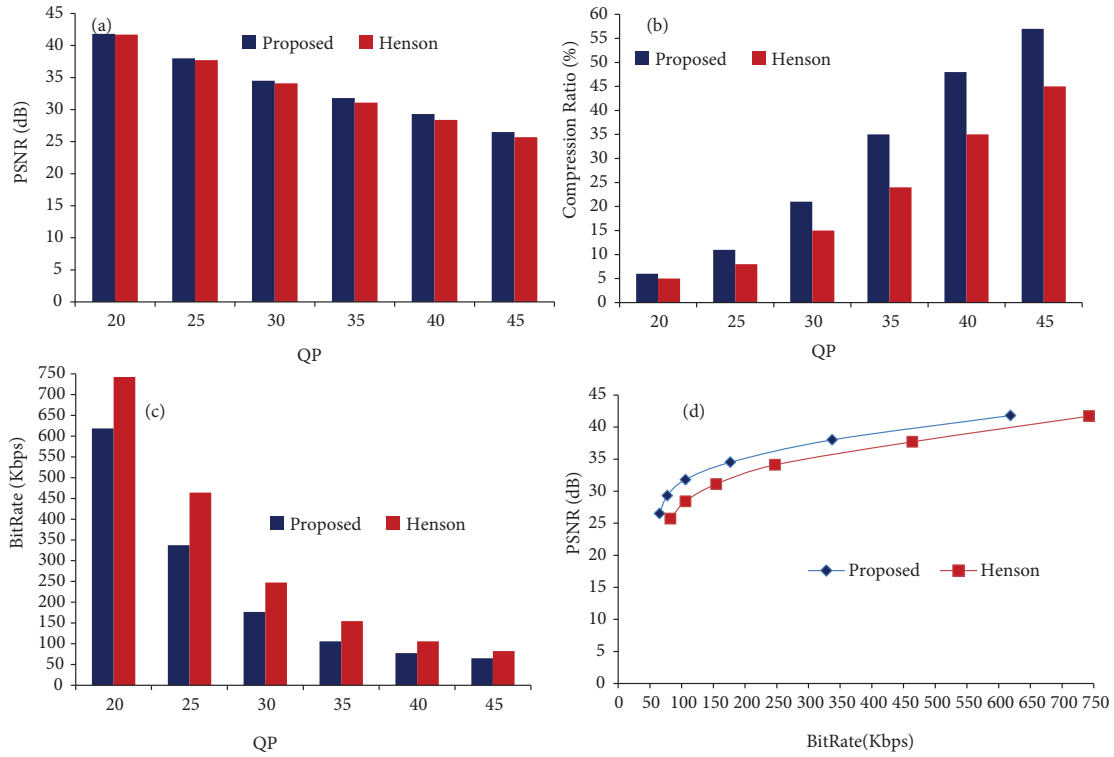


Figure 10. Evaluation of BPMM using Silent CIF video.

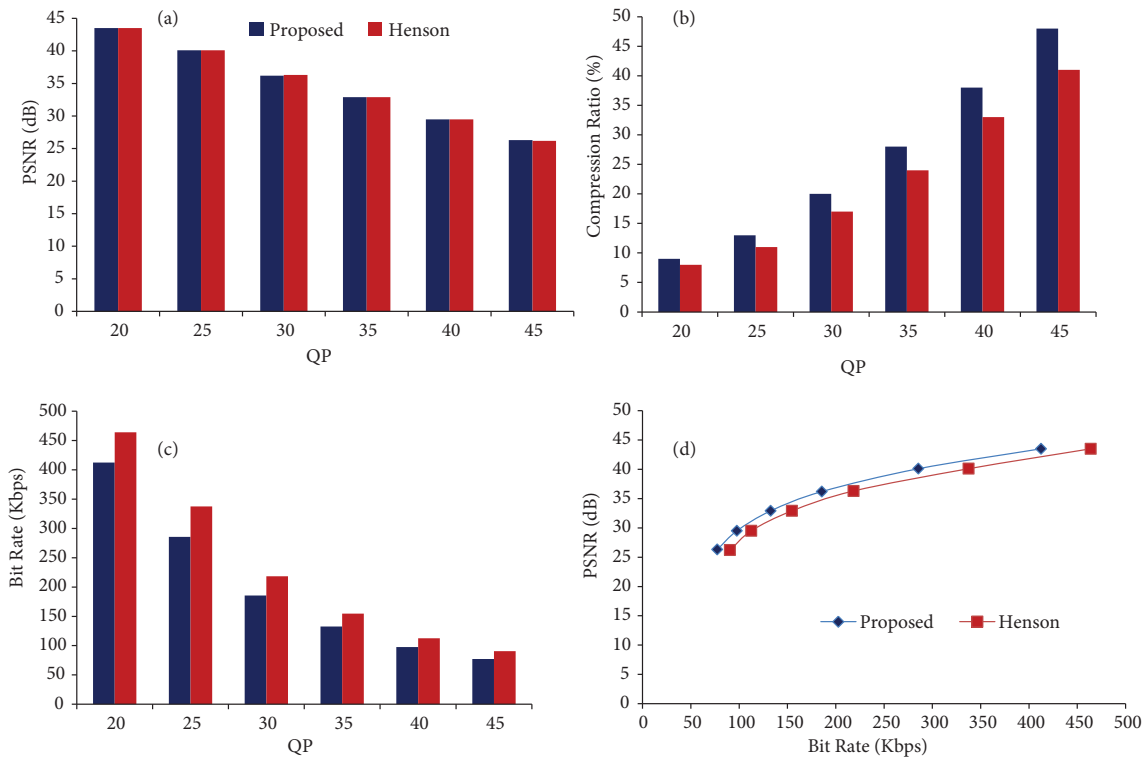


Figure 11. Evaluation of BPMM using News CIF video.

In addition, Table 4 shows the evaluation results of the proposed algorithm compared to Henson [18] when applied to two HD videos; namely ducks_ take_ off (size of 1280 × 720) and park_ joy (size of 3840 × 2160). The results show that the proposed prediction technique enhances the compression ratio and the PSNR for the two HD videos.

Table 4. Experimental results of BPMM applied on HD video sequences.

Algorithm	Performance metrics	Video sequences	
		Ducks_ take_ off 1280 × 720	Park_ joy 3840 × 2160
BPMM	PSNR (dB)	25.1	29.2
	Compression ratio increase (%)	55	60
	Bit rate (kbps)	613.6	5062.5
Henson [18]	PSNR (dB)	24.4	28.6
	Compression ratio increase (%)	39	50
	Bit rate (kbps)	865.38	6075

Seeking for a much more accurate judgment of the proposed prediction technique, we compared it with the hardware implementations of Bharathi and Nagabhushana [3], which used the Lena CIF video, and Loukil et al. [25], which used the Foreman CIF video, as shown in Table 5. The comparison is performed in terms of PSNR only as the considered papers for comparison did not include information about compression ratio or bit rate. In addition, the proposed prediction technique is compared with the software implementations of Tajdid et al. [6] and Joint Model (JM) version 18.2 [26] using the Mother-Daughter QCIF video sequence as shown in Table 6.

Table 5. PSNR (in dB) comparison between the proposed approach and recent hardware implementations.

Sequence	BPMM	Bharathi and Nagabhushana [3]	Loukil et al. [25]
Lena	42.56	37.3	-
Foreman	41	-	38.08

Table 6. Comparison between the proposed approach and recent software implementations.

	PSNR (dB)	Compression ratio	Bitrate (kbps)
BPMM	42.4	5	185.625
Tajdid et al. [6]	39.679	2.32	400
Tourapis et al. [26]	37	2.32	400

It is important to mention that these results are based on the assumption that the standard H.264 headers are used. These headers do not support sending the prediction mode (2 bits per pixel) for each pixel as it supports sending the prediction mode for each block (4 bits per block). In [27], it was mentioned that in the standard H.264 encoder, the encoder sends a flag for each 4 × 4 block. If the flag is ‘1’, the most probable prediction mode is used. We think that this mechanism should be extended for our proposed algorithm to reduce the frame overhead, hence allowing the researchers to achieve the calculated compression ratio (and hence bit rate). This extension is planned for future work.

6.2. Power consumption analysis

Power consumption is a major concern for reconfigurable architecture users and vendors. High power and energy leads to decreased battery life and increased costs for packaging and cooling. This is especially important if these reconfigurable architectures are used in hand-held mobile devices [28].

Total power in FPGAs (the targeted configurable architecture) as in any other semiconductor device is the sum of two components, namely static and dynamic power.

Static power results primarily from transistor leakage current in the device. Leakage current is the small current that ‘leaks’ either from source to drain or through the gate oxide, even when the transistor is logically ‘off’.

The design implemented on a FPGA has a greater effect on FPGA dynamic power consumption than on FPGA static power.

The dynamic power is the power consumed during switching. The dynamic power dissipation depends on the capacitive loading of each transistor, the voltage applied, and the frequency with which the transistor is switched. It represents the additional power resulting from the design activity. This power varies over time with the design activity. It also depends on voltage level and logic and the used routing resources. The simple equation governing dynamic power consumption is given in Eq. (1):

$$DynamicPower = 0.5CV^2f\alpha, \quad (1)$$

where C is the capacitance of the node switching, V is the supply voltage, f is the switching frequency, and α is the activity rate.

We estimate power consumption for the BPMM by using the XPower estimator available inside Xilinx ISE [29]. To obtain accurate estimates, we used postsynthesis estimation methodology that loads the synthesized design that has already been fully placed and routed into the power analysis tool (Xpower). Xpower then used the maximum clock frequency obtained from the place-and-route synthesis tool along with a commonly used activity rate of 12.5% [30].

This methodology represents a tradeoff between spreadsheet-based early power estimators that do not consider the synthesis results of the design, and hence produce inaccurate estimate [30], and simulation-based estimation, which is more accurate than our adopted methodology as it extracts the needed information (specifically the operating frequency and activity rate) from the completely synthesized design and then applies some test sequences using logic simulators to capture nodes’ switching activities. However, these simulation cycles need a huge time (days in normal test sequences) to finish one complete test sequence; in addition, their results depend on the test sequences used in the simulations and hence produce somewhat biased results.

Table 7 demonstrates the evaluation results of the BPMM new algorithm compared to Henson [18] regarding power consumption. The results show that the total power is increased on average by 4% and the dynamic power is increased on average by 8.3% which is an acceptable increase.

Table 7. Power consumption analysis for both BPMM and Henson algorithms.

Attributes	BPMM	Henson [18]
Total power (W)	0.39	0.375
Dynamic power (W)	0.181	0.167

6.3. Area analysis

The synthesis of the proposed algorithm is achieved using Xilinx ISE 9.2i software. The complete H.264 encoder including our new prediction algorithm is loaded to Xilinx Spartan-3E FPGA to calculate the area usage. Table 8 illustrates the area utilization results of the new proposed BPMM algorithm compared to that obtained with the plain H.264 implementation by Henson [18]. From the table, it can be observed that the total number of

occupied slices is increased by 12.41%, LUTs area is increased by 20.89%, select RAMs area is decreased by 0.76% and FFs count is increased by 9.63% while all other area components are not changed. These changes are acceptable in terms of cost increase.

Table 8. FPGA area analysis for both BPMM and Henson algorithms.

Attributes	BPMM	Henson [18]
Number of occupied slices	4890	4350
LUTs	6261	5179
Select RAMs	2085	2101
FFs	2800	2554
Shift registers	4	4
IOBs	89	89
Multipliers	2	2
BRAMs	4	4

Regarding the performance of the algorithm, synthesis results show that the proposed algorithm runs with a clock speed of 70.4 MHz, which is comparable to that obtained by Henson [18] (70.7 MHz).

The results indicated that the new intraprediction scheme, the so-called BPMM, achieves better compression ratios, bit rates, and PSNR for the five CIF videos and two HD videos and over all the QP values under test while maintaining almost the same level of execution time and power consumption. Regarding the silicon area, the results show that our prediction scheme involves almost 12% increase in the number of occupied FPGA slices.

7. Conclusions

Recently, video compression has become a very hot research topic due to the tremendous increase in video sizes and the limitations of the available bandwidth for transferring data. Among several video encoding standards, H.264 is the most recent and efficient one. The intraprediction stage in H.264 significantly contributes to its compression efficiency. In this paper, a new efficient H.264 intra 4×4 prediction scheme that relied on the combination of the most utilized intraprediction modes was proposed. First, several intraprediction algorithms were compared. The results showed that the SAD algorithm outperformed the others. Hence, it was utilized as a reference algorithm to be compared with the proposed prediction scheme. Second, a performance evaluation of the new intraprediction scheme was accomplished. The results indicated that the new intraprediction scheme, the so-called BPMM, achieves better compression ratios, bit rates, and PSNR for the considered CIF and HD videos and over all the QP values under test while maintaining almost the same level of execution time, hardware area utilization, and power consumption.

It is worth mentioning that although we focused on 4×4 prediction for luma blocks, our technique can be used in luma 16×16 and chroma 4×4 blocks, as well. However, because the intra 4×4 prediction is suitable for the parts with significant details, while the intra 16×16 was applied to the smoother areas, we did not consider the 16×16 luminance blocks. In addition, as the human visual system is less sensitive to color than to luminance (brightness), chroma 8×8 intra prediction did not receive researchers' attention in previous research efforts [5,14] and hence we did not focus on it.

As a future extension to the research in this paper, we plan to propose a pipelined architecture implementation to increase the processing speed of the H.264 encoder. Moreover, we are working on the necessary modifications to the H.264 decoder that account for our proposed H.264 encoder utilizing the new BPMM intraprediction technique in order to standardize our proposed prediction scheme.

Another important issue to be considered is the overhead due to sending the prediction mode for each pixel instead of each block. In [27], it was mentioned that in the standard H.264 encoder, the encoder sends a flag for each 4×4 block. If the flag is '1', the most probable prediction mode is used. We think that this mechanism is helpful in terms of reducing the frame overhead. However, it is necessary to modify the standard H.264 encoder to support using this feature as well as modify the decoder to support the new frame header format.

As it is hard to compare different hardware implementations due to the variations in the utilized FPGAs as well as synthesis tools, another future work direction is to compare our modification to Henson's work [18] with other efficient hardware implementations such as the implementation proposed by Sahin and Hamzaoglu [31].

Finally, software implementations in multicore and GPU machines will be investigated to reduce the cost of using extra hardware to implement the heavy computational operations.

References

- [1] Hamzaoglu I, Tasdizen O, Sahin E. An efficient H.264 intra frame coder system. *IEEE T Cons Elect* 2008; 54: 1903-1911.
- [2] Vimal KS, Muralidhar P, Rama Rao CB. Architecture for H.264 intra prediction fast mode decision algorithm. *Int J Comput Appl* 2013; 68: 1-6.
- [3] Bharathi SH, Nagabhushana KR. Verilog realization of diagonal-down-left intra prediction for H.264 video encoder. *Int J Eng Res Technol* 2012; 1: 1-13.
- [4] Fritts JE, Zhang H, Steiling FW. Fast intra-frame mode selection for H.264. In: *SPIE Electronic Imaging - Image and Video Communications*; January 2005; San Jose, CA, USA.
- [5] Wang L, Po LM, Uddin YMS, Wong KM, Li S. A novel weighted cross prediction for H.264 intra coding. In: *IEEE 2009 International Conference on Multimedia and Expo*; 28 June–3 July 2009; New York, NY, USA. pp. 165-168.
- [6] Tajdid UA, Jafor I, Touhid UA. Fast DC mode prediction scheme for intra 4x4 block in H.264/AVC video coding standard. *Int J Adv Comput Sci Appl* 2012; 3: 90-94.
- [7] Walter F, Bampi S. Synthesis and comparison of low-power architectures for SAD calculation. In: *IEEE Second Latin American Symposium on Circuits and Systems*; 2011; Bogota, Columbia. pp. 1-4.
- [8] Muralidha P, Devi RV, Rao CBR, Murthy NS. An efficient architecture for H.264 intra prediction mode decision algorithm. In: *Proceedings of the 10th WSEAS International Conference on Signal Processing, Robotics and Automation*; 2011; Wisconsin, USA. pp. 113-116.
- [9] Liang W, Dan-dan D, Juan D, Bin-bin Y, Lu Y. An efficient hardware design for HDTV H.264/AVC encoder. *J Zhejiang Univ-Sc C* 2011; 12: 499-506.
- [10] Mikołaj R, Grzegorz P. Intra prediction for the hardware H.264/AVC high profile encoder. *J Signal Process Sys* 2014; 76: 11-17.
- [11] Palomino D, Corrêa G, Diniz C, Bampi S, Agostini L, Susin A. Algorithm and hardware design of a fast intra-frame mode decision module for H.264/AVC encoders. In: *Proceedings of the 24th Symposium on Integrated Circuits and Systems Design*; 2011. pp. 143-148.
- [12] Bahri N, Werda I, Samet A, Ayed MAB, Masmoudi N. Fast intra mode decision algorithm for H264/AVC HD baseline profile encoder. *Int J Comput Appl* 2012; 67: 8-13.
- [13] Elyousfi A, Tamtaoui A, Bouyakhf ELH. Fast mode decision algorithm for intra prediction in H.264/AVC video coding. *Int J Comput Sci Netw Sec* 2007; 7: 356-364.
- [14] Sairam YN, Nan M, Neelu S. A novel partial prediction algorithm for fast 4x4 intra prediction mode decision in H.264/AVC. In: *Proceedings of the IEEE Data Compression Conference*; 2008; Snowbird, UT, USA. pp. 232-241.

- [15] Lin YLS, Kao CY, Kuo HC, Chen JW. VLSI Design for Video Coding: H.264/AVC Encoding from Standard Specification to Chip. 1st ed. Berlin, Germany: Springer, 2010.
- [16] Lee YL, Han KH, Sim DG, Seo J. Adaptive scanning for H.264 intra coding. ETRI J 2006; 28: 664-667.
- [17] Kim DY, Kim DK, Lee YL. Adaptive scanning using pixel similarity for H.264/AVC. IEICE T Fund Electr 2007; 90A: 1112-1114.
- [18] Henson A. H.264 Hardware Encoder in VHDL. Technical Report. Oxford, UK: Zexia Access Ltd, 2008.
- [19] Xilinx Corp. Spartan-3E FPGA Family Data Sheet, DS312. San Jose, CA, USA: Xilinx, 2013.
- [20] Olivares J, Hormigo J, Villalba J, Benavides I. Minimum sum of absolute differences implementation in a single FPGA device. In: Proceedings of the 14th International Conference on Field Programmable Logic and Application; September 2004; Leuven, Belgium. pp. 986-990.
- [21] Sarwer MG, Wu QMJ. Improved intra prediction of H.264/AVC. In: Radhakrishnan S, editor. Effective Video Coding for Multimedia Applications. Rijeka, Croatia: InTech, 2011. pp. 3.1-3.18.
- [22] Kong W, Li WJ, Guo M. Manhattan hashing for large-scale image retrieval. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2012; Portland, OR, USA. pp. 45-54.
- [23] Sara H, Mostafa EAI, Abdelhalim MB. Efficient H.264 intra prediction scheme based on best prediction matrix mode. In: Proceedings of the International Conference on Electronics, Computer and Computation; November 2013; Ankara, Turkey. pp. 367-371.
- [24] Hamdy S, Ibrahim MEA, Zekry A. VHDL realization of efficient H.264 intra prediction scheme based on best prediction matrix mode. Int J Comput Appl 2013; 77: 1-7.
- [25] Loukil H, Werda I, Masmoudi N, Ben Atitallah A, Kadionik P. FPGA design of an intra 16 × 16 module for H.264/AVC video encoder. SCIRP J Circuit Syst 2010; 1: 18-29.
- [26] Tourapis AM, Leontaris A, Suhring K, Sullivan G. H.264/14496-10 AVC Reference Software Manual. Geneva, Switzerland: Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, 2009.
- [27] Richardson IEG. H.264 and MPEG-4 Video Compression. 2nd ed. New York, NY, USA: Wiley, 2010.
- [28] Jamieson P, Luk W, Wilton SJE, Constantinides GA. An energy and power consumption analysis of FPGA routing architectures. In: Proceedings of the International Conference on Field-Programmable Technology; 2009; Sydney, Australia. pp. 324-327.
- [29] Xilinx Corp. Xilinx XPower Estimator User Guide, UG440. San Jose, CA, USA: Xilinx Corp., 2012.
- [30] Abdelhalim MB, Habib SED. Fast hardware upper-bound power estimation for a novel FPGA-based HW/SW partitioning scheme. In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI; April 2008; Montpellier, France. pp. 393-398.
- [31] Sahin E, Hamzaoglu I. An efficient hardware architecture for H.264 intra prediction algorithm. In: Proceedings of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools; August 2007; Lubeck, Germany. pp. 448-454.