

# Architecture for H.264 Intra Prediction Fast Mode Decision Algorithm

Vimal Kumar Shrivastava  
Research Scholar  
NIT Raipur  
Raipur, C.G., India

P. Muralidhar  
Assistant Professor  
NIT Warangal  
Warangal, A.P., India

C. B. Rama Rao  
Associate Professor  
NIT Warangal  
Warangal, A.P., India

## ABSTRACT

The paper presents an architecture for fast mode decision algorithm in H.264/AVC for 4x4 intra-prediction. This algorithm is based on the inherent symmetry existing in the spatial prediction modes. This algorithm gives a faster way of calculating sum of absolute differences (SADs) for all modes of intra prediction. So, a significant computational savings can be achieved. Synthesis results confirmed that the proposed architecture is able to process HD videos (1280x720) at 30 fps as well as 60 fps and Full HD videos (1920x1088) at 30 fps in ASIC platform and maximum frequency achieved is 63 MHz.

## Keywords

AVC, H.264, Intra Prediction, Inherent Symmetry, mode decision, SAD, HD videos, ASIC platform.

## 1. INTRODUCTION

H.264/MPEG-4 Part10 AVC is the latest video coding standard developed jointly by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The video compression efficiency achieved in H.264 standard is not a result of any single feature but rather a combination of a number of encoding tools. One of the tools is the intra prediction algorithm used in the baseline profile of H.264 standard [1]. Intra prediction algorithm generates a prediction for a Macroblock (MB) based on spatial redundancy. H.264 intra prediction algorithm achieves better coding results than the intra prediction algorithms used in the previous video compression standards. However, this coding gain comes with an increase in encoding complexity which makes it an exciting challenge to have a real-time implementation of H.264 intra prediction algorithm [1].

## 2. H.264 INTRA FRAME PREDICTION

For the H.264 intra coding, it takes advantages of the small block size in spatial domain. These image blocks are coded based upon directional predictions, which are computed-extensive. For the luminance (luma) samples, intra prediction may be formed for each 4x4 block or for a 16x16 macroblock. There are a total of 9 prediction modes for each 4x4 luma block; 4 modes for a 16x16 luma block. The latest H.264 standard also defines 8x8 block and also has 9 prediction modes which are the same as those modes used in 4x4 block. Similarly, for chroma 8x8 block, another 4 prediction modes are used.

### 2.1 4x4 Luma Intra Prediction Modes

There are nine 4x4 luma prediction modes designed in a directional manner as shown in Fig 1(a). A 4x4 luma block consisting of the pixels a to p is shown in Figure 1(b). The pixels A to M belong to the neighboring blocks and are assumed to be already encoded and reconstructed and are

therefore available in the encoder and decoder to generate a prediction for the current MB. Each 4x4 luma prediction mode generates 16 predicted pixel values using some or all of the neighboring pixels A to M as shown in Figure 2. The arrows indicate the direction of prediction in each mode. The predicted pixels are calculated by a weighted average of the neighboring pixels A-M for each mode except Vertical, Horizontal and DC modes [2].

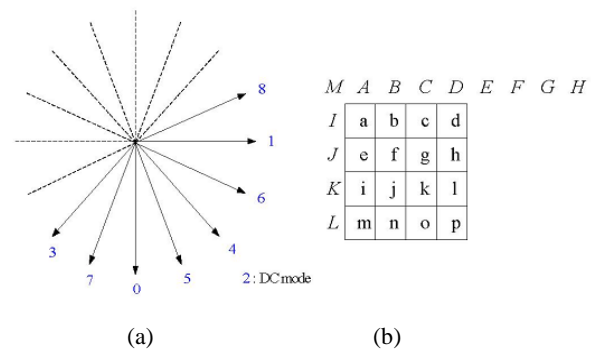


Fig 1: Directions of nine 4x4 intra prediction modes and a 4x4 block with its neighboring pixels [3]

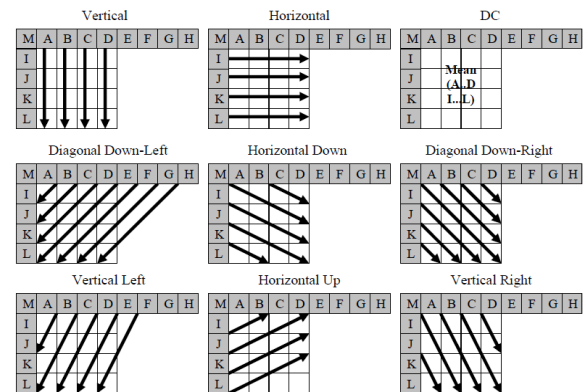


Fig 2: 4x4 luma prediction modes [4]

For horizontal and vertical modes the neighboring reconstructed pixel values are directly assigned. DC mode is a simple average of the neighboring reconstructed pixels.

### 3. IMPLEMENTED FAST MODE DECISION ALGORITHM

The proposed algorithm is based on the spatial symmetry in the intra prediction process and uses pixel to pixel correlation as the basis for obtaining a subset of candidate modes. This algorithm gives a faster way of calculating sum of absolute differences (SADs) for all modes of intra prediction. So, a significant computational savings can be achieved.

#### 3.1 Inherent Symmetry in Intra-Prediction Modes

Fig. 3 depicts the inherent symmetry in the various intra-prediction modes. Assume that this figure shows the prediction blocks of all modes of the current block indexed with respective mode numbers. From this representation the pixel to pixel correlation can be easily observed in each mode. For example, in Mode 0 - the Vertical prediction mode, due to the symmetry, we can see that the pixels in each column will have the same predicted pixel value. Similarly, in Mode 1 - the Horizontal mode, the pixels in each row will have the same predicted value. Subsequently, such symmetry observations can be extended to all the different modes [5].

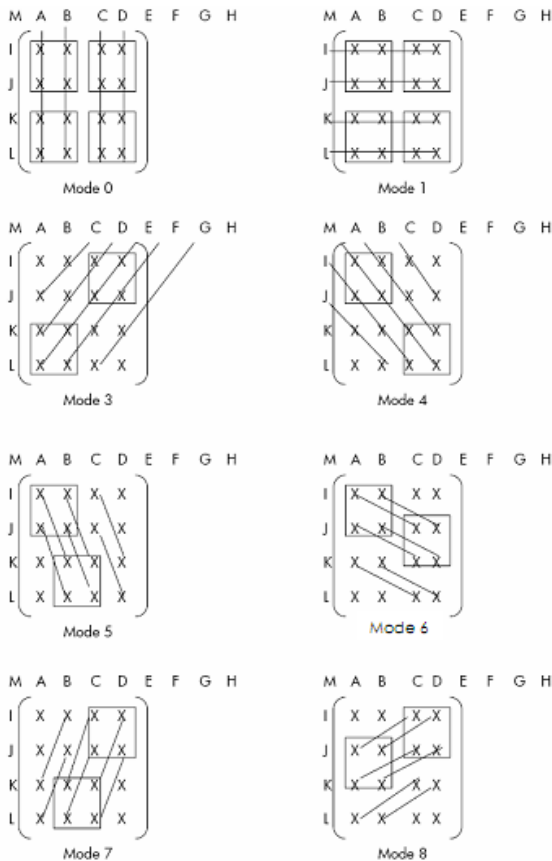


Fig 3: Inherent Symmetry and Pixel Correlation in different intra-prediction modes [5]

Now, consider eight 2x2 sub-blocks in the original 4x4 block X at different positions as shown in Fig.4. They are labeled as  $a_j$  where  $j$  is an index to position of sub-block. Let  $m_{ij}$  represent similar sub-block in  $i^{\text{th}}$  prediction mode at  $j^{\text{th}}$

position. We may consider the same notation for the sub-block average also. Thus, for Mode 0 with Vertical symmetry, we can see that  $m_{01}$  is same as  $m_{03}$ , and,  $m_{02}$  is same as  $m_{04}$ . Similar symmetry exists in all modes as summarized in Table1.

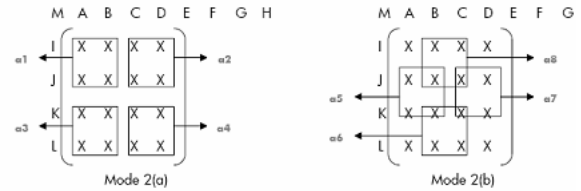


Fig 4: 2x2 Sub-blocks [5]

Table 1. Inherent Symmetry in Intra-prediction modes [5]

Mode	Symmetry	Sub-Block Cost Measures
0	$m_{01}=m_{03}$ ; $m_{02}=m_{04}$ or $m_{05}=m_{06}$	$ a_1 - m_{01}  +  a_2 - m_{02}  +  a_3 - m_{01}  +  a_4 - m_{02} $
1	$m_{11}=m_{12}$ ; $m_{13}=m_{14}$ or $m_{15}=m_{17}$	$ a_1 - m_{11}  +  a_3 - m_{12}  +  a_2 - m_{11}  +  a_4 - m_{12} $
2	$m_{21}=m_{22}=m_{23}=m_{24}$ or $m_{25}=m_{26}=m_{27}=m_{28}$	$ a_1 - m_{21}  +  a_2 - m_{21}  +  a_3 - m_{21}  +  a_4 - m_{21} $
3	$m_{32}=m_{33}$	$( a_2 - m_{31}  +  a_3 - m_{31} ) \times 2$
4	$m_{41}=m_{44}$	$( a_1 - m_{41}  +  a_4 - m_{41} ) \times 2$
5	$m_{51}=m_{56}$ or $m_{58}=m_{54}$	$ a_1 - m_{51}  +  a_6 - m_{51}  +  a_8 - m_{52}  +  a_4 - m_{52} $
6	$m_{61}=m_{67}$ or $m_{65}=m_{64}$	$ a_1 - m_{61}  +  a_7 - m_{61}  +  a_5 - m_{62}  +  a_4 - m_{62} $
7	$m_{72}=m_{76}$ or $m_{78}=m_{73}$	$ a_2 - m_{71}  +  a_6 - m_{71}  +  a_8 - m_{72}  +  a_3 - m_{72} $
8	$m_{85}=m_{82}$ or $m_{83}=m_{87}$	$ a_5 - m_{81}  +  a_2 - m_{81}  +  a_3 - m_{82}  +  a_7 - m_{82} $

#### 3.2 Prediction Process

Using the above symmetry in different prediction modes, from Table 1, we try to simplify the intra prediction process as follows. We try to obtain a close measure to SAD cost function in each mode (which is just the sum of absolute differences between 16 pixels in X and P) in terms of those sub-blocks which come under symmetry in respective prediction modes as listed in the Table 1 above.

In mode 0, the sum of absolute differences between  $a_1$  and  $m_{01}$ ;  $a_2$  and  $m_{02}$ ;  $a_3$  and  $m_{03}$ ;  $a_4$  and  $m_{04}$  gives the cost. However since  $m_{01}=m_{03}$  and  $m_{02}=m_{04}$  in mode 0 due to the symmetry we can simply calculate the cost function as  $|a_1 - m_{01}| + |a_2 - m_{02}| + |a_3 - m_{01}| + |a_4 - m_{02}|$ , where  $|x|$  stands for the absolute value for  $x$ . Similarly cost is calculated in all other modes exploiting the existing symmetry in corresponding prediction mode as listed out in Table 1. The pixels which are not covered under symmetry (like in modes 3-8) are taken care of by replicating the pixel differences that are covered under symmetry. The fact that the intensity variations within a block are not so rapid from pixel to pixel and also that these neighboring pixels are similarly predicted from the same adjacent pixels justifies the assumption [5].

#### 3.3 Pre-calculation Cost

Note that all of the  $m_{ij}$  measures can be calculated from 13 pixels A-M from neighboring blocks using the corresponding pixel values in respective positions in each prediction block depending on the mode. The cost function is simplified by using averages of sub-blocks instead of pixels directly. It is further simplified by the symmetry observed in Table 1, which brings down the number of 'm' measures that are to be pre-computed in each mode. As an example mode 0 requires only  $m_{01}$  and  $m_{02}$  to be computed as  $(A+B)/2$  and  $(C+D)/2$  respectively instead of 4 measures  $m_{01}$  to  $m_{04}$ . Modes 2-4 require only one such measure to be computed and the rest of the modes requires at most two such measures. So a total of

15 measures are required to be computed in addition to 8 averages  $a_j$  in the original block to calculate the cost function for all modes. This is the total pre-calculation cost for a block. To further cut down the pre-calculation cost, we avoid divisions by considering only additions of pixels in sub-blocks rather than averages. This final pre-calculation cost is insignificant compared to the computational gain achieved [5].

#### 4. PROPOSED ARCHITECTURE

The block diagram for spatial symmetry based algorithm of intra prediction process is shown in fig. 5.

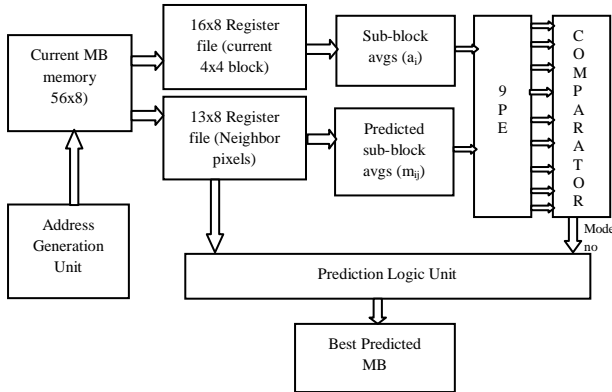


Fig 5: Block Diagram of intra prediction unit

The proposed H.264 intra frame coder hardware is divided into two main parts; the mode decision part and prediction part. The mode decision hardware and the prediction hardware work in a parallel manner. There is one memory of size 256x 8 and one address generation unit, which generate addresses to access one 4x4 block (16 pixels) and its corresponding neighboring block (13 pixels) from memory. After the current 4x4 block and its neighboring pixels are loaded to the input register files (16x8), mode decision hardware starts to work on determining the best mode out of 9 modes based on the minimum SAD value and in parallel with mode decision hardware, prediction logic unit also starts to predict the 4x4 luma block by using equations specified in h.264 standard draft [7]. Finally, the output of mode decision unit selects the best predicted mode from all the 9 modes processed by prediction logic unit.

#### 4.1 Architecture for Sub-Block Averages Block

Fig. 6 shows the architecture for sub-block averages block, which calculates the 8 sub-block averages ( $a_i$ ) as shown in fig. 4. The input for this block comes from 16 registers each of 8-bits, which holds the value of 16 pixels of current 4x4 block.

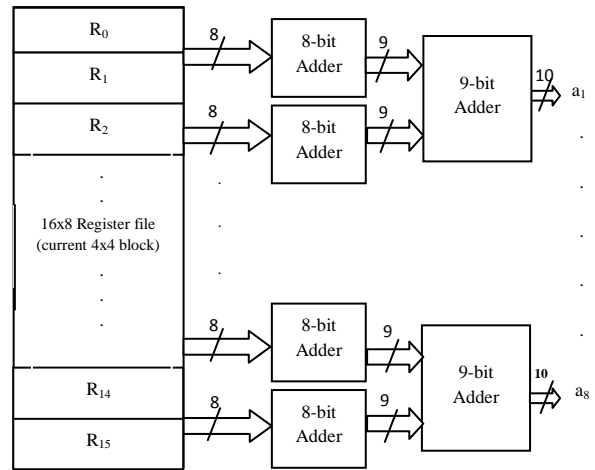


Fig 6: Architecture for sub-block averages block

#### 4.2 Architecture for Predicted Sub-Block Averages Block

Fig. 7 shows the architecture for predicted sub-block averages block, which calculates the 15 predicted sub-block averages ( $m_{ij}$ ) as shown in Table 1. The input for this block comes from 13 registers each of 8-bits, which holds the value of 13 neighboring pixels of current 4x4 block.

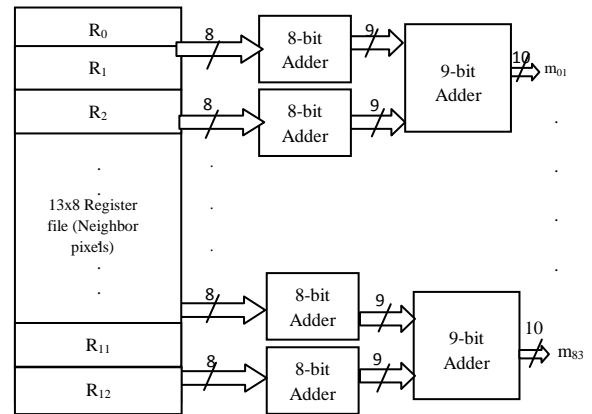


Fig 7: Architecture for predicted sub-block averages block

#### 4.3 Processing Element (PE)

In fig. 8, architecture for processing element (PE) is shown, which calculate 9 SADs for 9 prediction modes of one 4x4 luma block. The one of the input for this is 8 sub-block averages  $a_j$  where  $j$  is an index to position of sub-block and the other input is 15 predicted sub-block averages  $m_{ij}$  where  $m_{ij}$  represent similar sub-block in  $i^{th}$  prediction mode at  $j^{th}$  position. ABS block gives the absolute value of output of subtractor block.

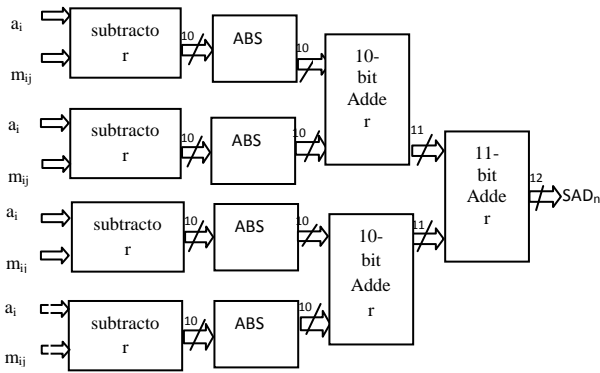


Fig 8: Architecture for processing element (PE)

Example: for mode 0, the sum of absolute differences between  $a_1$  and  $m_{01}$ ;  $a_2$  and  $m_{02}$ ;  $a_3$  and  $m_{03}$ ;  $a_4$  and  $m_{04}$  gives the cost. However since  $m_{01}=m_{03}$  and  $m_{02}=m_{04}$  in mode 0 due to the symmetry we can simply calculate the cost function as  $|a_1 - m_{01}| + |a_2 - m_{02}| + |a_3 - m_{01}| + |a_4 - m_{02}|$ , which is nothing but the  $SAD_0$ . Similarly cost is calculated in all other modes by this PE block.

#### 4.4 Prediction Logic Unit

The prediction logic block consists of two blocks, equation generation block and equation assignment block. Input for this block comes from the 13 neighboring pixels of current 4x4 block and the outputs are prediction equations for all 9 modes. Architecture for this block is shown in fig. 9.

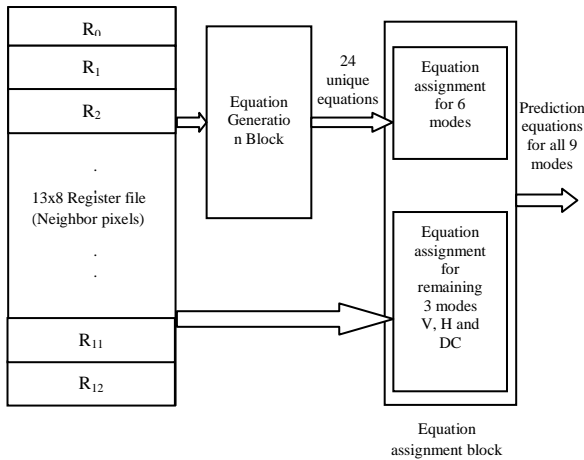


Fig 9: Architecture for prediction logic unit

##### 4.4.1 Equation Generation Block

The standard formulae for different modes are transformed into pixel processing equations specified in h.264 standard draft [8]. For horizontal and vertical modes the neighboring reconstructed pixel values are directly assigned. DC mode is a simple average of the neighboring reconstructed pixels. For the remaining six modes all the equations are sorted to find out the unique set of equations by equation generation block, which result in 24 equations only [6].

Unique equations:

$$\begin{aligned} D1 &\leq L & D2 &\leq (K+L+L+L+2) \\ D3 &\leq (K+L+1) & D4 &\leq (J+K+K+L+2) \end{aligned}$$

$$\begin{aligned} D5 &\leq (J+K+1) & D6 &\leq (I+J+J+K+2) \\ D7 &\leq (I+J+1) & D8 &\leq (M+I+I+J+2) \\ D9 &\leq (M+I+1) & D10 &\leq (A+M+M+I+2) \\ D11 &\leq (M+A+1) & D12 &\leq (M+A+A+B+2) \\ D13 &\leq (A+B+1) & D14 &\leq (A+B+B+C+2) \\ D15 &\leq (B+C+1) & D16 &\leq (B+C+C+D+2) \\ D17 &\leq (C+D+1) & D18 &\leq (C+D+D+E+2) \\ D19 &\leq (D+E+1) & D20 &\leq (D+E+E+F+2) \\ D21 &\leq (E+F+1) & D22 &\leq (E+F+F+G+2) \\ D23 &\leq (F+G+G+H) & D24 &\leq (G+H+H+H+2) \end{aligned}$$

##### 4.4.2 Equation Assignment Block

The predicted pixels are calculated by a weighted average of the neighboring pixels A-M for each mode except Vertical, Horizontal and DC modes. The prediction equations used in 4x4 Diagonal Down Left mode is shown in Fig. 10, where  $[y,x]$  denotes the position of the pixel in a 4x4 block (the top left, top right, bottom left, and bottom right positions of a 4x4 block are denoted as  $[0, 0]$ ,  $[0, 3]$ ,  $[3, 0]$ , and  $[3, 3]$ , respectively) and  $\text{pred}[y,x]$  is the prediction for the pixel in the position  $[y,x]$ .

$$\begin{aligned} \text{pred}[0, 0] &= A + 2B + C + 2 \gg 2 \\ \text{pred}[0, 1] &= B + 2C + D + 2 \gg 2 \\ \text{pred}[0, 2] &= C + 2D + E + 2 \gg 2 \\ \text{pred}[0, 3] &= D + 2E + F + 2 \gg 2 \\ \text{pred}[1, 0] &= B + 2C + D + 2 \gg 2 \\ \text{pred}[1, 1] &= C + 2D + E + 2 \gg 2 \\ \text{pred}[1, 2] &= D + 2E + F + 2 \gg 2 \\ \text{pred}[1, 3] &= E + 2F + G + 2 \gg 2 \\ \text{pred}[2, 0] &= C + 2D + E + 2 \gg 2 \\ \text{pred}[2, 1] &= D + 2E + F + 2 \gg 2 \\ \text{pred}[2, 2] &= E + 2F + G + 2 \gg 2 \\ \text{pred}[2, 3] &= F + 2G + H + 2 \gg 2 \\ \text{pred}[3, 0] &= D + 2E + F + 2 \gg 2 \\ \text{pred}[3, 1] &= E + 2F + G + 2 \gg 2 \\ \text{pred}[3, 2] &= F + 2G + H + 2 \gg 2 \\ \text{pred}[3, 3] &= G + 3H + 2 \gg 2 \end{aligned}$$

Fig 10: Prediction Equations for 4x4 Diagonal Down-Left Mode

As shown above, the prediction equations for Diagonal Down Left mode, other modes also have prediction equations. The equation generation block generates all those equations in the form of 24 unique equations except for vertical, horizontal and DC modes, because for horizontal and vertical modes the neighboring reconstructed pixel values are directly assigned and DC mode is a simple average of the neighboring reconstructed pixels. So, equation assignment block gives prediction equations for all 9 modes, for 6 modes with the help of 24 unique equations and for other 3 modes V, H and DC, directly by 13 neighboring pixels of current 4x4 block [5].

## 5. IMPLEMENTATION

The architecture is defined in a hardware description language (VHDL) and synthesized by the Synopsys Design Compiler with 0.13 $\mu\text{m}$  standard cell library. The design specifications are shown in Table 2.

**Table 2. Design Specifications**

Algorithm	Fast Mode Decision
Technology	0.13 $\mu$ m
Application	HD videos
Frame per second	30
Memory Requirement	(256x8) to store current MB
Pixel parallelism	4 pixels

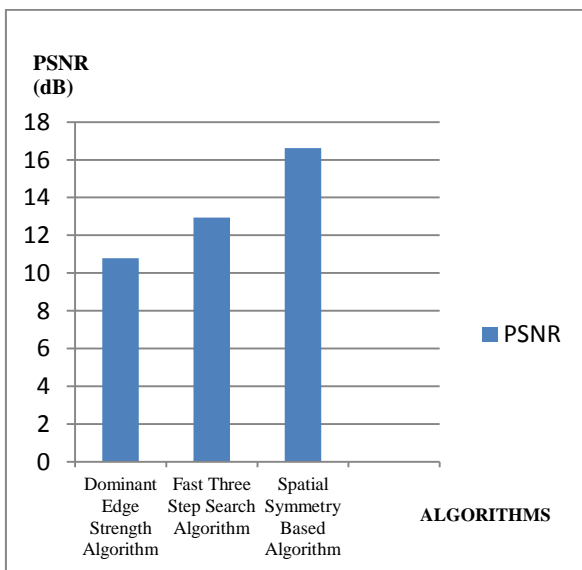
The performance analysis after compiling using the Synopsys Design Compiler is given in Table 3.

**Table 3. Performance Analysis**

Maximum Frequency	63 MHz
Critical Path Delay	15.10ns
Total cell area	219.54 K
Total dynamic power	1.4118 mW
Throughput	1059.60 x 10 <sup>6</sup> Luma samples/sec

## 6. EVALUATION AND RESULTS

The proposed algorithm is compared with two other algorithms three step search algorithm [6] and dominant edge strength algorithm [7] in matlab and observed that proposed algorithm is giving the best PSNR value as shown in fig. 11. Architecture for the proposed algorithm is implemented on FPGA Altera stratix II device and maximum frequency achieved is 153 MHz and also synthesized and simulated on synopsys tool and maximum frequency achieved is 63 MHz. In table 4, the proposed approach is compared with previous approaches for FPGA based implementation and in table 5, the proposed approach is compared with previous approaches for synopsys based implementation. Observation shows that the proposed architecture is able to process HD videos (1280x720) at 30 as well as 60 fps and Full HD videos (1920x1088) at 30 fps.



**Fig 11: Comparison of PSNR for three algorithms**

**Table 4. Comparison with previous works for FPGA based Implementation**

Design feature	Our approach	[9]	[10]	[6]
Device used	Altera Stratix II	Altera Stratix II	Xilinx Vertex II Pro	Xilinx Vertex II Pro
Maximum frequency	153 MHz	103 MHz	110 MHz	90 MHz
Maximum Target Size	Full HD (1920 x 1088)	Full HD (1920 x 1088)	Full HD (1920 x 1088)	VGA frame (640x480)

**Table 5. Comparison with previous works for synopsys based Implementation**

Design feature	Our approach	[11]	[12]
Technology	0.13 $\mu$ m	0.13 $\mu$ m	0.18 $\mu$ m
Maximum frequency	63 MHz	54 MHz	61 MHz
Total cell area	219 k	84 k	72 k
Maximum Target Size	Full HD (1920 x 1088)	SDTV (720x480)	HDTV (1280x720)

## 7. CONCLUSIONS

In this paper, we present a fast mode decision algorithm for intra prediction in H.264 video coding. The proposed algorithm is based on the spatial symmetry in the intra prediction process and uses pixel to pixel correlation as the basis for obtaining a subset of candidate modes. The proposed hardware architecture for h.264/AVC intra prediction supports all the intra modes, calculates the sum of absolute differences (SADs) and decides the best intra 4x4 predicted macro block. This algorithm is also compared with two other fast mode decision algorithms in matlab and observed that this algorithm is giving the best PSNR value. Architecture is developed for this algorithm and synthesized and simulated on synopsys tool. Synthesis results confirmed that the proposed architecture is able to process HD videos (1280x720) at 30 as well as 60 fps and Full HD videos (1920x1088) at 30 fps and maximum frequency achieved is 63 MHz.

## 8. REFERENCES

- [1] Iain E. G. Richardson, "H.264 and MPEG-4 Video Compression – Video Coding for Next-generation Multimedia", John Wiley & Sons, ISBN: 978-0-470-84837-1, 2003.
- [2] Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A.; "Overview of the H.264/AVC video coding standard," Circuits and Systems for Video Technology, IEEE Transactions on , vol.13, no.7, pp.560-576, July 2003.
- [3] Jia-Ching Wang; Jhing-Fa Wang; Jar-Ferr Yang; Jang-Ting Chen; "A Fast Mode Decision Algorithm and Its VLSI Design for H.264/AVC Intra-Prediction," Circuits and Systems for Video Technology, IEEE Transactions on , vol.17, no.10, pp.1414-1422, Oct. 2007.
- [4] Shafique, M.; Bauer, L.; Henkel, J.; "A parallel approach for high performance hardware design of intra prediction

- in H.264/AVC Video Codec," *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE '09.*, pp.1434-1439, 20-24 April 2009.
- [5] Sairam, Y.N.; Nan Ma; Sinha, N.; "A Novel Partial Prediction Algorithm for Fast 4x4 Intra Prediction Mode Decision in H.264/AVC," *Data Compression Conference, 2008. DCC 2008.*, pp.232-241, 25-27 March 2008.
- [6] Sahin, E.; Hamzaoglu, I.; "An Efficient Hardware Architecture for H.264 Intra Prediction Algorithm," *Design, Automation & Test in Europe Conference & Exhibition, 2007. DATE '07.*, pp.1-6, 16-20 April 2007.
- [7] Chao-Chung Cheng; Tian-Sheuan Chang; "Fast three step intra prediction algorithm for 4x4 blocks in H.264," *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pp. 1509- 1512 Vol. 2, 23-26 May 2005.
- [8] Jhing-Fa Wang; Jia-Ching Wang; Jang-Ting Chen; An-Chao Tsai; Anand Paul; "A novel fast algorithm for intra mode decision in H.264/AVC encoders," *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pp.4, pp., 21-24 May 2006..
- [9] La-Gou Wu; Duo-Li Zhang; Gao-Ming Du; Yu-Kun Song; Ming-Lun Gao; "A 4x4 pipelined intra frame decoder for H.264," *Anti- , Security, and Identification in Communication, 2009. ASID 2009. 3rd International Conference on*, vol., no., pp.332-335, 20-22 Aug. 2009.
- [10] Diniz, C.M.; Zatt, B.; Agostini, L.; Susin, A.; Bampi, S.; "A real time H.264/AVC intra frame prediction hardware architecture for HDTV 1080P video," *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, vol., no., pp.1138-1141, June 28 2009-July 3 2009.
- [11] Yu-Wen Huang; Bing-Yu Hsieh; Tung-Chien Chen; Liang-Gee Chen; "Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol.15, no.3, pp. 378- 401, March 2005.
- [12] De-Wei Li; Chun-Wei Ku; Chao-Chung Cheng; Yu-Kun Lin; Tian-Sheuan Chang; "A 61MHz 72K Gates 1280X720 30FPS H.264 Intra Encoder," *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. no. 2, pp.II-801-II-804, 15-20 April 2007.