Efficient Adder Compressor Design for Sum of Absolute Difference Calculation

Abstract—**In video encoding Motion Estimation is the technique used to exploit temporal redundancy of video sequences. The Motion vector defines the change of previous frame from its present frame of the video in the Block Matching Algorithm. An accurate method is required to calculate the motion vector. The Sum of Absolute Differences (SAD) is the best distortion metric used to obtain the best match in Integer Motion Estimation. SAD calculation is one of the most time consumer operations of video encoders. SAD architectures employ an adder tree to accumulate the coefficients from absolute difference between two video frames. The 8:2 adder compressors implemented using combination of 4:2 adder compressor is found to be better among all the architectures based on delay and power consumption. The SAD architecture using four different structures of adder compressors are simulated using Xilinx ISE. The system performances are evaluated by comparing the power dissipation, delay and area.**

Index Terms—**Integer Motion Estimation (IME), Motion Vector (MV), Sum of Absolute Difference (SAD), Block Matching Algorithm.**

I. Introduction

Block Matching is one of the best Algorithm used for video encoding. The method helps to determine the motion vector and to predict the next frame with the information from presently coded frame. The blocks of present frame is compared with that in the previous frame. Difference between each frame are only coded to reduce the number of coding bits and thereby the bandwidth required. Motion Vector is an indicator of change in video frames from one another. Motion estimation is of two types namely, Integer Motion Estimation and Fractional Motion Estimation. The Integer Motion Estimation uses symmetric blocks of the frame to be compared with the nearby frames.

Bi-directional coding scheme uses the technique of Motion Prediction to code a video in forward or reverse direction. The Sum of Absolute Difference (SAD) is one of the best distortion metrics used to determine the change of each frame from one another. Sum of absolute difference can be formulated as the sum of absolute differences between the pixel values within blocks of the two adjacent frames. SAD is high if the blocks are different and SAD=0 for two similar blocks in the two frames. SAD helps to determine the dissimilarities easily. Another metric generally used is Mean Absolute Difference (MAD). SAD is more advantageous over MAD, because SAD generally gives integer but the MAD values are fractional. SAD also needs a faster calculation. The SAD value can be easily calculated with the help of adder compressors.

The adder compressors are the most efficient computational elements that are used for adding pixel difference for SAD calculation with an improved speed. The size of adder compressors range from 3:2 to 8:2 (number of inputs: number of outputs), where the outputs are always sum and carry. The adder compressors are compact form of full adders, which are efficient in terms of delay, area and power compared to full adders. Four different architectures are used to implement an 8:2 adder compressor and their performance parameters are compared in work [1]. The 8:2 adder compressor is implemented using different combinations of 3:2, 4:2, 5:2 and 7:2 adder compressors.

Organization of Report: The report begins with introduction and the significance of work as section 1 followed by an extensive survey on the literature in section 2. A description of the proposed system is given in section 3 along with the details of software used in section 4. The results and analysis are discussed in section 5, finally the report is wrapped up with conclusion in section 6 and challenges, advantages and applications in section 7.

II. Literature Review

The SAD architecture implementation using adder compressors are mainly used in video encoding with high speed and reduced power requirements. The work presents an overview of various adder compressors and the distortion matrices (mainly Sum of Absolute Difference) used in video coding.

In [2] the operation of 4-2 and 5-2 compressors are presented. The 4-2 adder compressor have five inputs and two outputs, all the outputs having the same weight. The 4:2 and 5:2 compressors are logically decomposed and are used to produce a new low-power circuit with good drivability. The circuit produces XOR and XNOR output simultaneously. The circuit is robust against delay driven voltage scaling. HSIM 2.0 tool with the option HSIMSPEED set to zero in HSPICE is used for the circuit simulation.

The work [3] is a simple method to implement a 7:2 adder compressor which are very useful in the field of higher order multipliers, where the partial products are to be added. The circuit simulator used is HSPICE. Mathematical calculations are used to convert the conventional compressors to the presented compressor. A carry generation module is used to obtain carry with at each level without waiting for the carry. The method has a better value of average power consumption, critical delay path and power delay product.

In [4] motion vector calculation for video encoding is used. Current and the Reference Macro Blocks (MB) are compared to obtain the motion vector. Full search is employed. To obtain computation efficiency and to optimize complexity, search algorithms like Three Step Search (TSS) and a new Diamond Search (DS) were introduced. The paper concentrates on SAD implementation. SAD architecture is functionally divided into three stages: absolute difference calculation, accumulation of absolute differences and minimum SAD determination. The candidate block with minimum SAD value MV determines the minimum changing block. Algorithm to calculate SAD is also used.

In [5] an algorithm to calculate SAD is presented. Unsigned number comparison is done using this method. An absolute Difference Calculation Unit (ADC) called a one -bit CMP, uses the 1-bit comparators to calculate the difference between the two unsigned numbers as the input. Compression Array Unit (CAU) is implemented with the help of each ADC units. The architecture reduces area significantly at the cost of latency. Bypass channel is introduced in 2 stage pipe line, calculation in the second and third stages are omitted to reduce the latency and power consumption.

III. System Description

Sum of Absolute Difference can be implemented effectively using adder compressors. The SAD value helps to formulate the motion vector and thereby predict the next frame. Use of adder compressors make the SAD calculation process more fast as described in work [1]. There are different sized adder compressors ranging from 3:2 to 8:2. The use of adder compressor improves the speed of performance and reduces the power consumption in the process of pixel difference summation.

A. Adder Compressors (3:2 and 4:2)

The 3:2 adder compressor sums up all the three input bits A, B and C. The carry bit is selected based on the XOR sum of A and B bits. The maximum delay due to two XOR gates in the critical path.

$$S = A \oplus B \oplus C \qquad (1)$$

$$S = Sum + 2Carry \qquad (2)$$

The final sum S of the 3-2 adder compressor is given by equation 1 and 2. The 3-2 adder compressor is nearly same as a full-adder (i.e. mux-based full-adder) if the input C is used as carry input.

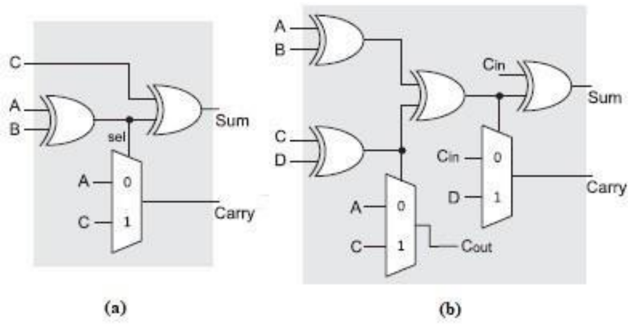


Fig. 1. Adder Compressors internal structures (a) 3-2; (b) 4-2.

The 3:2 adder compressor internal structure is as shown in Fig. 1(a). The 4:2 Adder compressor add up 4 input partial products. There are two stages of mux used in carry generation. Two sets of input and output carries are produced by a 4:2 adder compressor. The Fig. 1 (b) shows a 4:2 adder compressor internal structure with reduced critical path.

$$S = A \oplus B \oplus C \oplus D \oplus C_{in} \qquad (3)$$

$$S = Sum + 2(C_{out} + C_{carry}) \qquad (4)$$

The equation 3 and 4 gives the sum output of the of 4:2 adder compressor.

B. Adder Compressor (5:2 AND 7:2)

The 5:2 and 7:2 adder compressors are other adder compressor needed to be used in the architectures for an 8:2 adder compressor. They take 5 and 7 inputs respectively and gives out the total sum and carry.

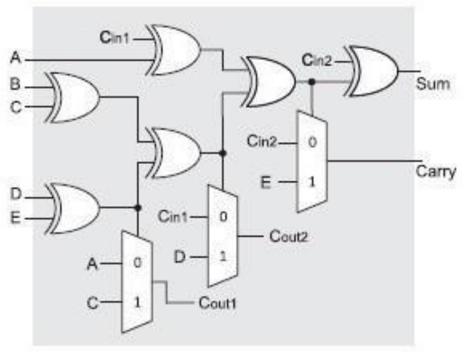


Fig. 2. Adder compressor internal structures (a) 5:2; (b) 7:2

The 5:2 adder compressor internal structure is shown in Fig. 2. The 7:2 adder compressor uses a carry generate module to obtain carry at every level of operation. It is simply the product term of each of the input terms to it.

C. Adder Compressor 8:2

An 8:2 adder compressor has 8 inputs, 4 carry in and 4 carry out respectively. An 8:2 adder compressor can be implemented using three 4:2 adder compressors. The 8:2 adder compressor can be used to add the pixel differences of two frames of an 8*8 i×mage in row or column wise. SAD value of one column/row of the image with reduced delay and improved power consumption. The combination of 8-bit adder compressor can be used for complete SAD calculation of 8 × 8 image.

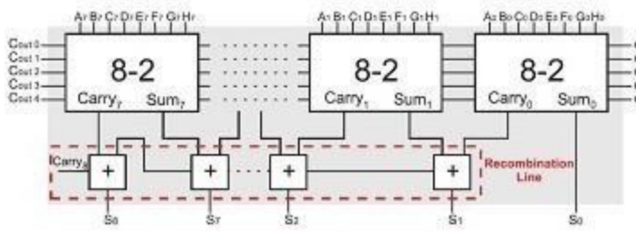


Fig. 3. SAD calculation using array of 8:2 adder compressors

The Fig.3 shows the eight-bit adder using 8:2 adder compressor. The absolute difference value obtained from the absolute difference unit in Xilinx produces the large number of binary values. These values are then needed to be added using the adder compressor. This is done on Xilinx using Verilog coding.

D. 8:2 Adder Compressor Architectures

Architecture 1

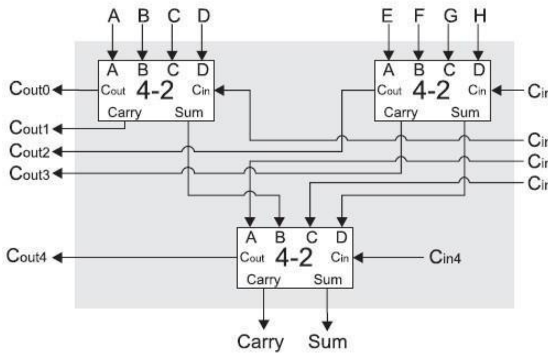


Fig. 4. Architecture 1 to implement 8:2 Adder compressor

The architecture 1 uses three 4:2 adder compressors to implement an 8:2 adder compressor. The circuit require 2 clocks to give complete output. The 8:2 adder compressor implemented by the combination of 4:2, m adder compressor is shown in Fig. 4.

Architecture 2

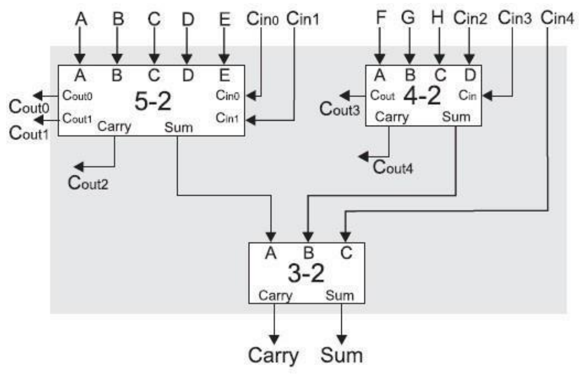


Fig. 5. Architecture 2 to implement 8:2 Adder compressor

The architecture 2 uses three 5:2,4:2 and 3:2 adder compressors to implement an 8:2 adder compressor. The circuit take 2 clocks to give complete output. The 8:2 adder compressor by the architecture2 is shown in Fig. 5.

Architecture 3

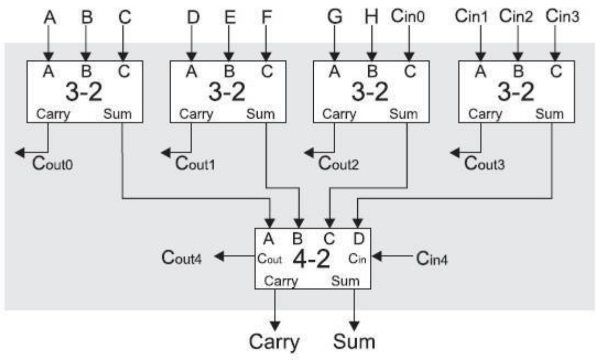


Fig. 6. Architecture 3 to implement 8:2 Adder compressor

Architecture 4

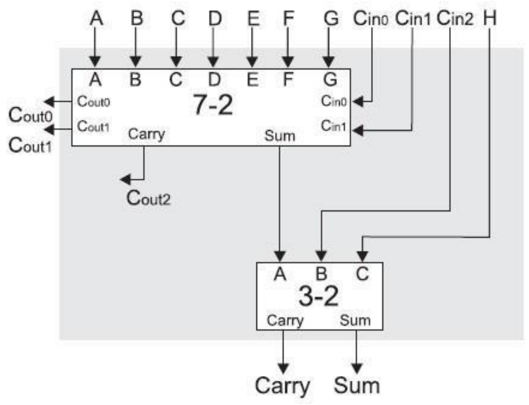


Fig. 7. Architecture 4 to implement 8:2 Adder compressor

Architecture 3 uses 3:2 and 4:2 compressors to implement 8:2 adder compressor and is shown in Fig. 6. The Fig. 7 shows the 8:2 adder compressor architecture using 7:2 and 3:2 adder compressor.

E. Sum of Absolute Difference Calculation

The SAD calculation of a video sequence has mainly four sections namely, frame generation, block division, pixel comparison and adder compressor unit. The frame generation unit divides the video into number of frames(images) whose SAD value is to be calculated. The block division unit divides the large image into different number of blocks.

Motion Estimation: The HEVC video is divided into different number of frames. The entire frame is divided into Coding Tree Units (CTU), each CTU are of block sizes ranging from 64×64 to 8×8. Prediction units are used to predict the next frame. Block Division methods are of two types: Symmetric Block partition (SMP) and Asymmetric Motion Partition (AMP). The Integer Motion Estimation (IME) uses Symmetric Motion Partition.

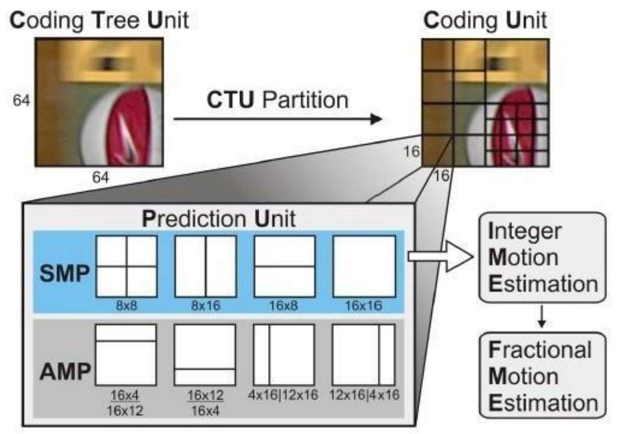


Fig. 8. Coding tree unit and the partitioning types.

The coding tree unit and the partitioning types are shown in Fig. 8.

Motion Prediction: This stage helps in predicting the next frame of the moving video with the previously calculated motion vector. Motion Vector defines the change between the two blocks during motion estimation.

Sum of Absolute Difference: Sum of Absolute Difference between two blocks the original and reference block respectively are calculated by using the equation as follows:

$$SAD = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |O_{i,j} - R_{i,j}| \qquad (5)$$

The general equation for calculating SAD is given by equation 5. Here, O represents the original block (block in the current frame) and R represents the Reference block (block in the reference frame). m and n are the dimensions of the block in samples.

F. Working

An adder compressor adds up all the input sequences and provides sum and carry output. The addition is done as in a full adder and carry is with a reduced critical path. The adder compressor is the best tool used to obtain the sum of absolute pixel differences between two images. The SAD of two ×8 8 images are obtained by taking difference between each pixel in same position and further adding up the absolute values of differences between them. An image pixel has four image information associated with it namely, luminance, red, blue and green values of the pixels. The pixel values are obtained as hexadecimal data from a monochrome video. Each pixel values are selected and are given to the absolute difference unit. The absolute difference value is then given to the adder compressor unit to add it up.

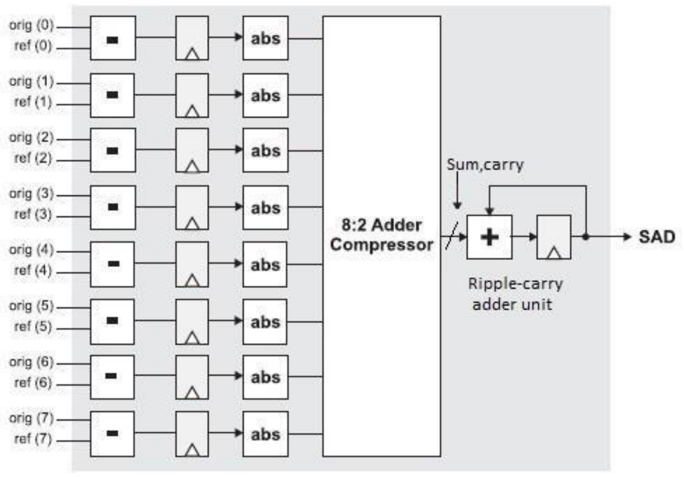


Fig. 9. System diagram of SAD architecture using 8-2 adder compressor.

The input image differences are first stored in register and are then given to an absolute difference unit. The Sum of Absolute Differences is obtained giving all the inputs to adder compressor. The sum and carry outputs of the adder compressor represent the SAD value and carry of the column/row of the image respectively. The Fig. 9 shows the functional diagram for SAD calculation.

G. Software Section

MATLAB (matrix laboratory) is a numerical computing environment. It allows plotting of functions and data, implementation of algorithms and interfacing with programs written in other languages, including C, C++, C, Java etc. The Matlab15b is to read, display and process the input test video. The test video is converted into monochromatic image frames and the hexadecimal pixel values are saved into .txt file using MATLAB.

Xilinx is a hardware depended simulation software that helps to view the system performance through program codes written in Verilog, VHDL etc. The Hexadecimal values obtained from the .txt file is read and processed in Xilinx.

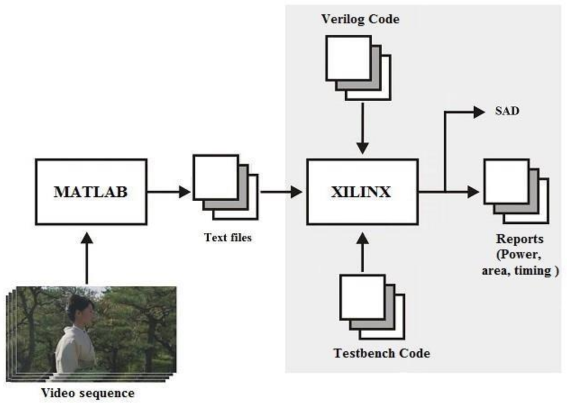


Fig. 10. Simulation workflow

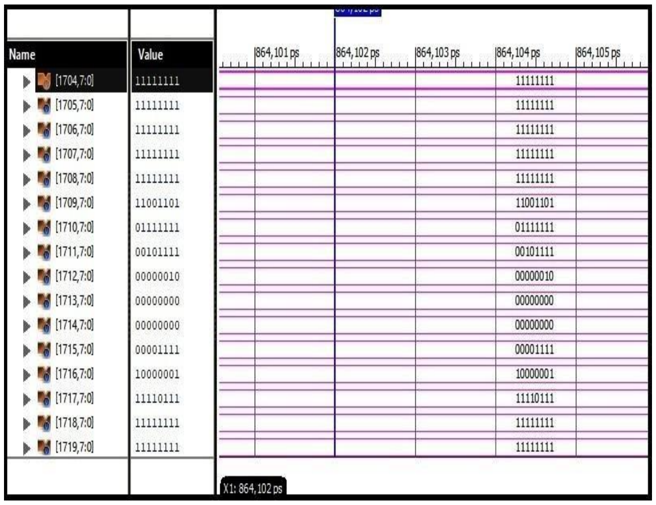
Verilog coding is used with the Xilinx 14.7 software for the programming. The simulation workflow is shown in Fig. 10. Each hexadecimal value in the image corresponds to 8-bit binary value and are used to be given into absolute difference and adder compressor unit simulated in Xilinx. The final output SAD is then saved into a .txt file.

Cadence is the software that helps in simulation of electronic circuits coded in Verilog or VHDL. It also helps to simulate analog circuits by drawing the circuit. The clock and timing details and be set in the script file. The power area and delay reports are obtained for each architecture of the 8:2 adder compressor using cadence software at 45 nm Technology.

IV. Results & Analysis

The results and analysis of SAD architectures for an 8 × 8 and 64 × 64image frames are depicted in this section.

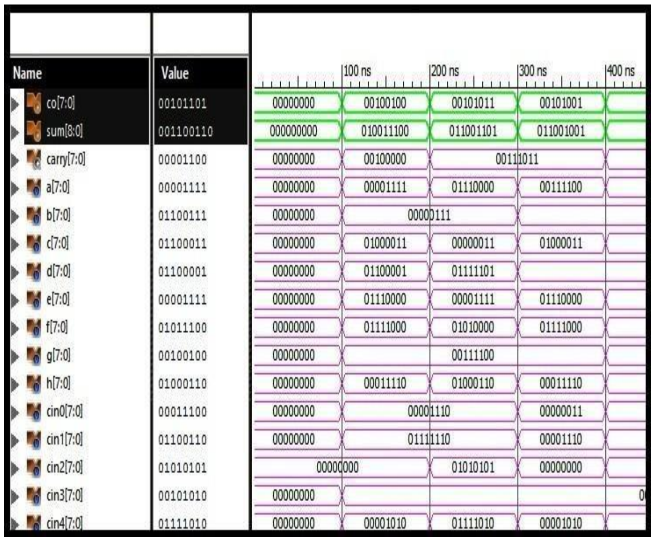
A. *Sum of Absolute Difference Simulation Outputs*



Name	Value
[1704,7:0]	11111111
[1705,7:0]	11111111
[1706,7:0]	11111111
[1707,7:0]	11111111
[1708,7:0]	11111111
[1709,7:0]	11001101
[1710,7:0]	01111111
[1711,7:0]	00101111
[1712,7:0]	00000010
[1713,7:0]	00000000
[1714,7:0]	00000000
[1715,7:0]	00001111
[1716,7:0]	10000001
[1717,7:0]	11110111
[1718,7:0]	11111111
[1719,7:0]	11111111

Fig. 11. Absolute difference obtained between two 8 × 8 image frames

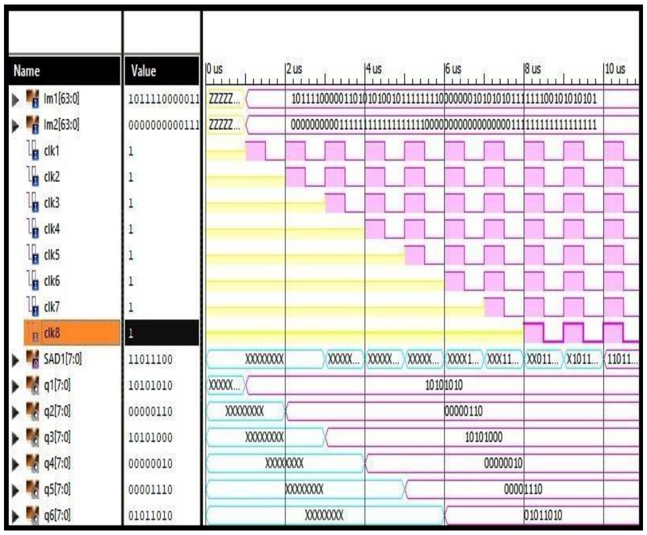
The absolute difference value obtained from each image pixel is eight bit long binary numbers, since the input image pixels are hexadecimal and is shown in Fig. 11.



Name	Value
co[7:0]	00101101
sum[8:0]	001100110
carry[7:0]	00001100
a[7:0]	00001111
b[7:0]	01100111
c[7:0]	01100011
d[7:0]	01100001
e[7:0]	00001111
f[7:0]	01011100
g[7:0]	00100100
h[7:0]	01000110
cin0[7:0]	00011100
cin1[7:0]	01100110
cin2[7:0]	01010101
cin3[7:0]	00101010
cin4[7:0]	01111010

Fig. 12. Output of 8:2 adder compressor using architecture1

Each output of SAD value for the eight rows of the input (one row at one clock cycle) is obtained within 3 cycles. So the final SAD calculation takes total 11 cycles to complete. The frequency of operation depending on the clock cycle is obtained as 610MHz. Since the architecture uses two levels of clock for operation, we assume the frequency to be half of the original, so the operating frequency is merely equal to 305Mhz. Output of an 8:2 adder compressor is shown in Fig. 12.



Name	Value
Im1[63:0]	1011110000011
Im2[63:0]	0000000000111
clk1	1
clk2	1
clk3	1
clk4	1
clk5	1
clk6	1
clk7	1
clk8	1
SAD1[7:0]	11011100
q1[7:0]	10101010
q2[7:0]	00000110
q3[7:0]	10101000
q4[7:0]	00000010
q5[7:0]	00001110
q6[7:0]	01011010

Fig. 13. SAD output for random 8 × 8 input sequence of two image values in 11 clock cycles

The Fig. 13 shows the SAD output for random 8 × 8 input sequence of two image values in 11 clock cycles using the first architecture.

B. 8:2 Adder Compressor Architectures

The power consumption and delay results are obtained using cadence software at 45nm technology. The method of power extraction is done with the help of some files named constraint, TCL, run. TCL etc.

	Adder-Tree Combination			
Recombination Adder	Architecture1	Architecture2	Architecture3	Architecture4
Mux-based ripple-carry adder	35111.127	44915.998	34178.484	35473.211

TABLE I
POWER DISSIPATION (N W) RESULTS @ 45NM TECHNOLOGY

	Adder-Tree Combination			
Recombination Adder	Architecture1	Architecture2	Architecture3	Architecture4
Mux-based ripple-carry adder	2.042	2.240	2.858	2.149

TABLE II
DELAY (NS) RESULTS @ 45NM TECHNOLOGY

The table 1 and 2 shows the delay and power consumption comparison between each architecture and architecture1 is found to be better among them.

	Adder-Tree Combination			
Recombination Adder	Architecture1	Architecture2	Architecture3	Architecture4
Mux-based ripple-carry adder	469	469	476	476

TABLE III
CELL AREA (μm^2) COMPARISON RESULTS @ 45NM TECHNOLOGY

The table 3 shows the cell area comparison result for the adder compressor architectures. The comparison of all the three parameters shows that the architecture 1 is better architecture for 8:2 adder compressor implementation.

CONCLUSION

The four architectures of 8:2 adder compressors were compared in terms of power, area and delay. The 8:2 adder compressor using three 4:2 adder compressor is found to be the best among all the four architectures. The comparison is obtained using the report from cadence software at 45 nm technology and Xilinx report. The Sum of Absolute Difference of frames in a grayscale test video of different resolutions are obtained with the help of adder compressors using Xilinx and MATLAB. The SAD of each row of an 8*8 image are obtained at each clock and final value is obtained within the eleven clocks. The SAD values obtained using four architectures are easily comparable to identify the change of nearby frames. The SAD using 8:2 adder compressor built using the 4:2 adder compressor is more power and delay efficient.

REFERENCES

[1] Bianca Silveira, Guilherme Paim, Brunno Abreu, Mateus Grellert, Cludio Machado Diniz, Eduardo Antonio Cesar da Costa and Sergio Bampi, ”Power-Efficient Sum of Absolute Dierences Hardware Architecture Using Adder Compressors for Integer Motion Estimation Design”, *IEEE Transaction on Circuits and SystemsI: Regular Papers*, vol. 5, pp. 1–11, July 2017.

[2] Chip-Hong Chang, Jiangmin Gu and Mingyan Zhang,”Ultra Low-Voltage Low-Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits”, *IEEE Transaction on Circuits And SystemsI: Regular Papers*, vol. 51, no. 10, pp. 1985–1996, October 2004.

[3] Mahnoush Rouholamini, Omid Kavehie, Amir-Pasha Mirbaha, Somaye Jafarali Jasbi and Keivan Navi, ”A New Design for 7:2 Compressors”, *IEEE/ACS International Conference on Computer Systems and Applications*, vol. 21, no.12, pp. 847–858, 2007.

[4] Jarno Vanne, Eero Aho, Timo D. Hmlinen, and Kimmo Kuusilinna, ”A High-Performance Sum of Absolute Difference Implementation for Motion Estimation”,*IEEE Transaction on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp.876–883, July 2006.

[5]C.LiYufei, FengXiubo and WangQin, ”A High-Performance Low Cost SAD Architecture for Video Coding”, *IEEE Transactions on Consumer Electronics*, vol. 53, no. 7, pp.535–541, 2007.

[6] Bianca Silveira, Guilherme Paim, Claudio M. Diniz, Eduardo and A. C. da Costa, ”Power-Efficient Sum of Absolute Differences Architecture using Adder Compressors”,*IEEE International Conference on Electronics, Circuits and Systems(ICECS)*, vol. 36, pp. 290–301 2016.

[7] Fbio L. Walter, Cludio M. Diniz, Sergio Bampi, ”Synthesis and Comparison of Low-Power High-Throughput Architectures for SAD Calculation”, *IEEE International Conference on Electronics, Circuits and Systems*, vol. 42, pp. 1028–1033, 2011.

[8] Purnachand Nalluri1, Luis Nero Alves and Antonio Navarro,”High Speed SAD Architectures for Variable Block Size Motion Estimation In HEVC Video Coding”, *IEEE International Conference on Image Processing(IPCP)*, vol. 4, pp. 1233–1236, 2014.

[9] Zhenyu Liu, Satoshi Goto and Takeshi Ikenaga,” Optimization of Propagate Partial SAD and SAD Tree Motion Estimation Hardwired Engine for H.264”, *IEEE/ACS International Conference on Computer Systems and Applications*, vol. 18, pp. 328–333, 2008.

[10] Gustavo Sanchez, Marcelo Porto and Luciano Agostini ” Hardware Friendly Motion Estimation Algorithm for Emergent HEVC Standard and its Low Power Hardware Design”, *IEEE International Conference on Image Processing*, vol. 11, pp. 1991–1994, 2013.