Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 7
July 2023

Hardware-Oriented Block Matching with Single Precision Floating Point Division Newton-Raphson Algorithm for Fast Motion Estimation in Wire Less Networks

[1]Dr C R S Hanuman, [2]Matta Venkata Pullarao, [3]M Ravi Sankar
[1]Professor, [2,3]Assistant Professor
[1]crshanuman@sasi.ac.in, [2]mvenkat@sasi.ac.in, [3]mravi@sasi.ac.in
[1,2,3]Department of ECE, SASI Institute of Technology & Engineering
Tadepalligudem, INDIA

Abstract-In the past few decades, we have been running towards the Digital Age. Telephony networks, fiber optic networks, Internet, satellites, third-generation (3G) wireless networks, these advanced transmission technologies and network infrastructure ensure digital information can flood into every corner around us. At the same time, the traditional media format has been gradually replaced by today's digital multimedia formats. These new formats and technologies shorten the distance between people, and improve the quality of lives. In this paper we have presented a Fast Motion Estimation technique known as Quadrant based search algorithm with parallel pipeline divider (QSPD) is constructed using Newton-Raphson Algorithm. Motion vector of the video sequence is estimated with Stationary Block Prediction (SBP). Stationary Block Prediction technique is used to find the block with zero motion. Implementation results of our proposed algorithm are compared with previous methods such as ARPS and Diamond search algorithm. PSNR value and motion estimation time of our proposed algorithm are better than that of previous method. In my research work, we concentrate on extending the block matching algorithms (BMA) from single reference frame to multiple reference frames. BMAs are selected due to their successes in the past video coding standards and the simplicity of regular data structure that is favorable for both hardware and software implementations. Analysis of motion vector distribution in multiple reference frames. Analysis the searching pattern of some out-standing BMAs and Developed a novel searching strategy for multiple reference frames based on the analyzed results. A novel approach to develop the fast motion estimation algorithm results are analyzed in Xilinx ISE and implemented in Spartan low power FPGA-6.

Key Words-BMA, Peak Signal to Noise Ratio, Big Cross Search Pattern, Small Diamond Search Algorithm.

I. INTRODUCTION

For a decade, various divider concepts are used in video coding techniques have always been advancing our multimedia technology. From VCD, DVD, Internet streaming, to video conferencing, all these multimedia applications indicate a break through of video coding techniques. A wide range of standards has been developed for different multimedia applications, such as ITU-T H.261 [1], H.263 [2], ISO/IEC MPEG-1 [3], MPEG-2 [4] and MPEG-4 [5]. These standards focus on different application profiles in terms of picture quality, bit-rate, latency, network capability and complexity. They provide a common area for the interested bodies to work on and open a new market for consumer electronics. The new technology also revolutionizes the forms of different fields from time to time, for example, entertainments, teaching, medicine, geography, and meteorology.

Although today network bandwidth and storage continuously increasing, the endless demands of better image quality and faster communication can fill up the room easily. Supposing a digital video of 24-bit color depth with 720x480 resolutions is transmitted at 30 frames per second (fps), and then it requires 248Mbps bandwidth. High definition TV (HDTV), video-on-demand (VOD), video email and video conferencing and telephony are now the hottest and most demanded multimedia services. However, these services also require huge data bandwidth to provide the high quality pictures, and could easily consume a great portion of the network bandwidth of a service provider. Consequently, the providers would need to invest heavily on the network infrastructure. This is also the main concern that deters them from trying this business. As a result, many academic and industrial researchers are working on a more efficient video coding standard with different divider techniques integrated on it which is suitable for video applications in the limited bandwidth environment.

1.1 New Emerging Standard: H.264/MPEG-4 AVC

H.264/MPEG-4 AVC [6] is the latest video coding standard developed by the Joint Video Team (JVT) which is formed with ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Motion Picture Experts Group (MPEG) in 2001. VCEG was formed in 1997. It is chartered as Question 6 of Study Group 16 to establish international standards for conversational and non-conversational audio/visual applications. MPEG was formed in 1988. It is chartered as Work Group 11 of Subcommittee 29 of JTC1 to develop moving pictures coding standards for a wide variety of digital video applications like DVD (storage media) and VOD (broadcasting).

There are 3 reasons for the need of a new industry standard:

- The cost for processing power and memory has reduced. More heavy coding strategies can be supported;
- More network supports are available for coded video data;
- Video coding technology has been advanced. The previous codec is not up to date.

JVT gathers the preeminent experts from these 2 leading international standards organizations to develop a new standard that meet the ever-growing need for higher compression of various video applications like digital storage media, TV broadcasting, video conferencing and internet streaming. The standard will eventually be known as ITU-T H.264 and ISO/IEC MPEG-4 Part 10 Advanced Video Coding (AVC). It is supposed to be published in the mid 2003 [7].

This new standard significantly out-performed the existing video coding standards. The previous best video codec, H.263 and MPEG-4 Visual (Part 2), based on the video coding technology of around 1995 are mainly used for low bit-rate communication and multimedia streaming on web. MPEG-2 is the video coding format for HDTV and DVD. H.264 can save half of the bit-rate when compared with the former, and only use about quarter of that for the latter [8]. In other words, we can have 2X or 4X the video quality by renewing the current video codec while keeping the same bandwidth requirements. A recent research [9] on the average US household consumption of digital broadcast TV suggests that even a little improvement on the video compression efficiency (e.g. 10%), would reduce 20X of current Internet backbone traffic. This means that the new standard not only pursues higher quality, but also opens new opportunities for various bandwidth demanding video applications, especially for those mobile devices that have only limited bandwidth in the wireless network such as General Packet Radio Service (GPRS) and 3G.

1.2 Statement of the Problem

The significant gain of compression efficiency of H.264 is at the expense of increased computation and complexity. For examples, 4x4 DCT transform, advanced intra/inter-prediction modes, tree-structured macroblock partitioning, quarter-pixel motion compensation and multiple reference frames, all these features help to increase the compression efficiency, but introduce huge amount of computation and complexity requirements of hardware [6].

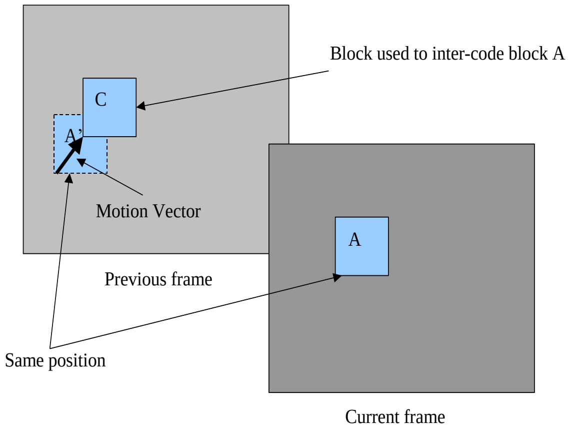


Figure 1.1: Motion compensation

In a hybrid-coding [10] video encoder, most of the computation is spent on motion estimation. Hybrid-coding is virtually the basis of all video coding standards, in which intra-frame coding and inter-frame coding are combined to use to reduce the spatial and temporal redundancy. Motion estimation (ME) and motion compensation (MC) are the two major components of inter-frame coding for video compression. As depicted in Figure 1.1, motion compensation is a kind of predictive coding that explores the data correlation in temporal domain by compensating for the displacement of objects between the reference frame and current frame. On the other hand, motion estimation is the process to find out the motion vector (MV) which represents the displacement of the objects. A good prediction can substantially increase the compression efficiency. Nevertheless, such kind of motion-compensated prediction technique is computationally intensive. Up to 80% computational power of an encoder can be consumed by motion estimation [11], and this is the case for exhaustive searching of all candidate blocks in a single reference frame only. In H.264, motion estimation is allowed to search on multiple reference frames with multiple partition sizes at quarter-pixel accuracy, and its reference software adopts full search scheme. Certainly, the computational load due to motion estimation must be increased greatly and directly proportional to the number of reference frames. VideoLocus, one of the leading developers of video compression technology, has recently developed a video evaluation platform for H.264. It shows that, without hardware acceleration, a 2GHz P4 computer requires 10 seconds to encode a single frame for H.264 [12]. This is absolutely a challenging requirement for implementing a software encoder with today computational power, especially for mobile computing devices. Obviously, it indicates there is an eager need for faster motion estimation strategy. In all the above mentioned optimization is done with respect to algorithmic level where as we are concentrating on reconfigurable design through optimizing divider and by introducing parallel pipelining architecture. In Section 2, an overview of the H.264 coding standard will be introduced. We will explain the impact of the advanced prediction features on the compression efficiency of the new codec. In Section 3, a summary of some well-know motion estimation algorithms and two recently proposed multiple-frame-selection algorithms will be presented with divider approach. In Section 4, a wholesale analysis of motion vectors distribution on multiple reference frames of various sequences will be shown with divider approach based on Newton raphston method. Based on the result, two novel fast motion estimation algorithms for multiple reference frames are proposed and the possible outcome will be discussed. Finally, our research will be concluded in Section 5.

2. EXISTING METHOD OVERVIEW OF H.264

H.264 is the most up-to-date video coding technology. Many advanced features are assembled in the standard. These features include 4x4 DCT transform which replaces the 8x8 DCT transform in previous standards; advanced intra-prediction which has up to 13 prediction modes for luminance blocks; advanced inter-prediction which support motion compensation in multiple reference frame with multiple macroblock partition size at quarter-pixel accuracy. In this chapter, we mainly focus on discussing about the prediction part of the new codec and how it essentially increases the video compression efficiency.

2.1 H.264 Codec Design

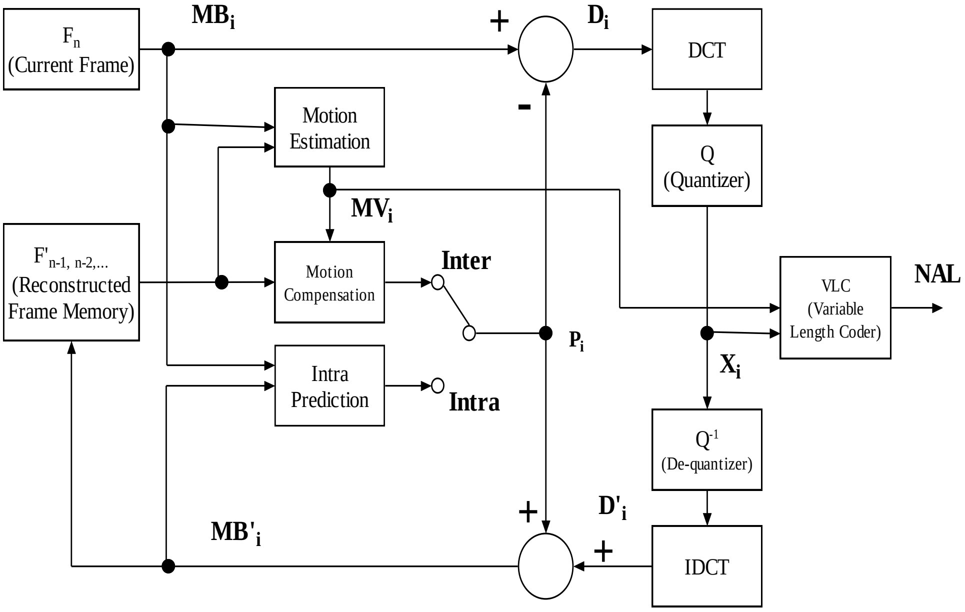


Figure 2.1: H.264 Video Encoder

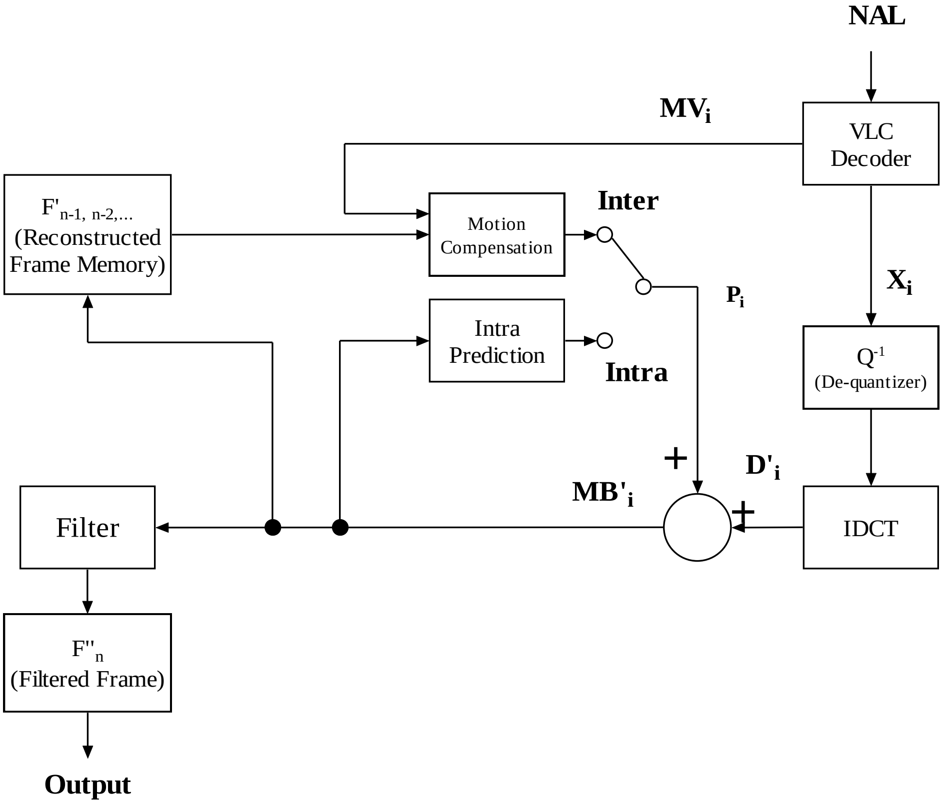


Figure 2.2: H.264 Video Decoder

Similar to other standards, the H.264 codec (encoder and decoder pair) consists of some basic functional elements like transformer, quantizer, variable length coder, and their counterparts. The encoder is shown in Figure 2.1. The encoding path starts from left to right. A raw frame F_n is input to the encoder. Each frame is partitioned into blocks with 16x16 pixels that they are called macroblocks. The encoder processes the macroblocks one by one. A macroblock MB_i is encoded either in intra or inter mode. For both modes, a prediction P will be produced. In inter mode, P_i is formed by motion compensation of the reference frames, $F_{n-1,\ n-2,\ \ldots}$, stored in frame memory. The motion estimation finds the best match of the current block from one or more reference frames and generates the motion vectors MV_i which represent the displacement of the block. In intra mode, P_i is predicted from the previous reconstructed neighboring macroblocks. The prediction is then subtracted from MB_i to produce the residual data D_i. The residual data is transformed and quantized for lossy compression to give X_i which contains the quantized DCT coefficients. These coefficients and motion vectors are entropy encoded to form a bitstream with other side information like quantization factor and prediction mode. It is sent to the Network Abstraction Layer (NAL) for storage and transmission. In return path, X_i is de-quantized and de-transformed. As quantization is a lossy process, the reconstructed residual data D'_i is not exactly the same as original. It is then added to P_i to form a reconstructed macroblock, and stored in the frame memory for further reference.

In the decoder shown in Figure 2.2, it is quite similar to the reconstructing path of the encoder. The decoding path starts from right to left and up to down. The data received from NAL are processed inversely as in the encoder. Eventually, the reconstructed frame is filtered to alleviate the artifacts due to lossy compression. One should notice the codec design that both the encoder and decoder use the identical references for prediction. Otherwise, error will accumulate on successive decoded frames. That is also why we have a reconstructing path in the encoder. Newton-Raphson computational division algorithm with parallel pipelining approach is introduce in the proposed and existing method in order to optimize the area overhead and delay with less power utilization as discussed below.

2.2 Intra Prediction of Macroblocks

The intra coding of H.264 is quite different to that of the previous standards. In H.264, if a macroblock is coded in intra mode, a prediction block is always generated from the neighboring decoded samples in spatial domain. Whereas, in MPEG-4 Visual and H.263+, only some of the transformed coefficients can be predicted from neighboring samples in frequency domain, and there is even no prediction in intra mode for the older standards like H.261, MPEG-1 and MPEG-2.

There are three types of intra coding supported in the standard, two for luma blocks and another one for chroma blocks. No matter which type of macroblocks, intra prediction is not allowed performing across slice boundaries to keep each slice independent of decoding.

2.2.1 Intra_4x4 prediction for luma samples

Q	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Figure 2.3: Label of samples used for intra prediction

3. MATERIALS A AND METHODS OF PROPOSED AND CONVENTIONAL FAST MOTION ESTIMATION ALGORITHMS

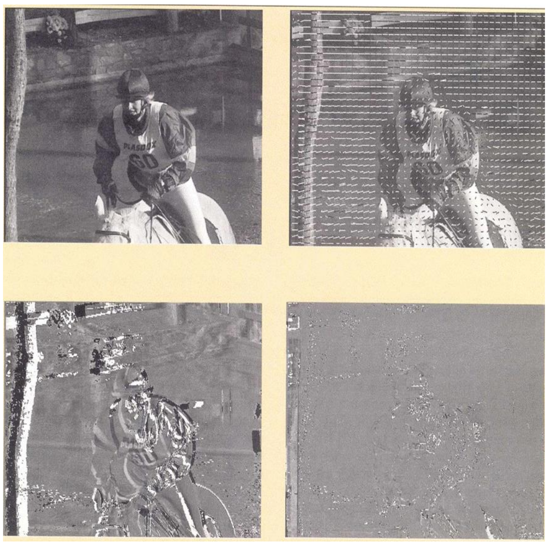

(a) Original raw frame F_n
(b) MV from frame F_{n-1}
(c) Prediction Error without using MC (i.e. all MVs are 0)
(d) Prediction Error if MC is applied

Figure 3.1: Prediction with / without motion compensation

Motion estimation plays an important role in video compression. In general, it can improve the compression efficiency by utilizing the temporal redundancy between adjacent pictures. As depicted in Figure 3.1, the motion compensated picture 3.1(d) has much less error than the one without it 3.1(c). It means that the necessary coded information is reduced. In H.264, as well as all other video coding standards, only the syntax elements of motion estimation is specified while the algorithm is left open. Researchers and vendors can have their room for development and competition. As a result, many kinds of motion estimation algorithms have been proposed [10], [14], [15]. Among those algorithms, one that commonly adopted by different communities and standards as a reference is the block matching algorithm with Newton-Raphson computational division algorithm.

3.1 Block Matching Motion Estimation

Most of the block matching motion estimation (BMME) algorithms rely on the following assumptions:

- The uniform illumination is along the motion axis;
- The uncovering background problem is not present;
- The pixels inside a block have the same shift between successive frames.

The first assumption states the problem that illumination may introduce different optical flows that are not related to the motion. The second assumption addresses the problem of scene change and uncovered objects that they are not exist in the previous frame. The last assumption points out that different motions may present within a block.

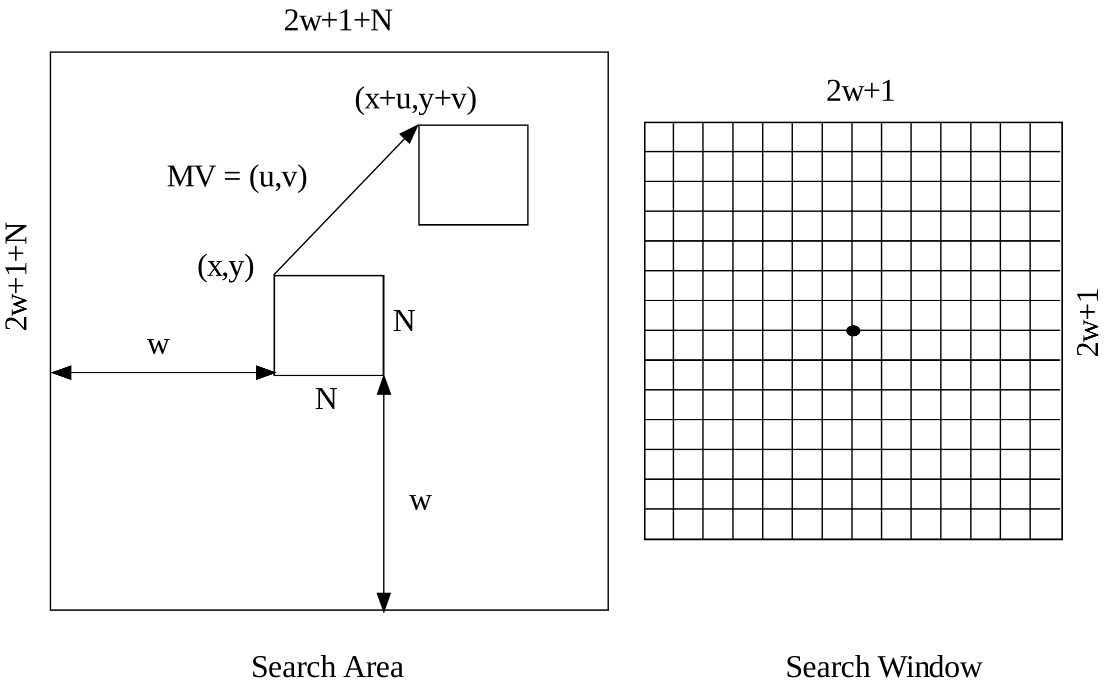


Figure 3.2: Search area and search window for block matching

A block matching algorithm finds the best match of the current block from the search area of the reference frame. Each frame is segmented into blocks of 16x16 pixels (macroblock) for processing. These blocks are given a search window where they are located in the center. Figure 3.2 shows a general case. Supposing the window size is w, then there is total $(2w+1)^2$ candidate blocks to be searched. The actual searched area on the reference frame should be to $(2w+1+N)^2$ if the block has a size of NxN. The best match is usually evaluated by a cost function which based on the block distortion measure (BDM) (e.g. Mean Square Error (MSE), Mean Absolute Error (MAE) and Sum of Absolute Difference (SAD)) or other criteria like bit-rate and number of motion vectors. The motion vector (MV) is then defined as the displacement of the best match from the current block. In all the reference models of the standards, full search (FS) scheme is adopted. In this case, we need to search through all the candidate blocks as mentioned above. The exhaustive FS can typically have 60% to 80% [11] computation of an encoder. Thus, fast searching algorithms are desirable to speed up the searching time and reduce the loading by optimizing divider approach.

3.2. Overview of our conventional hardware architecture

Figure.3.1(a) shows the hardware architecture of the conventional Quadrant based search algorithm with parallel pipelining divider and stationary motion prediction (SBP). It consists of current data buffer, reference data buffer, process element array, SAD comparator and also SBP block. From the figure the current data buffer and reference data buffer used to store the pixels values of current blocks and reference blocks. Processing element (PE) array is used to calculate the SAD value between the current and reference data. SAD comparator used to find the minimum SAD value from the calculated SAD values. SBP block has predicted threshold value and it used to compare the SAD of the centre point in the QSPD. In this architecture they have used single pipelining approach for divider concept is used as shown in the Figure.3.1.(b) .

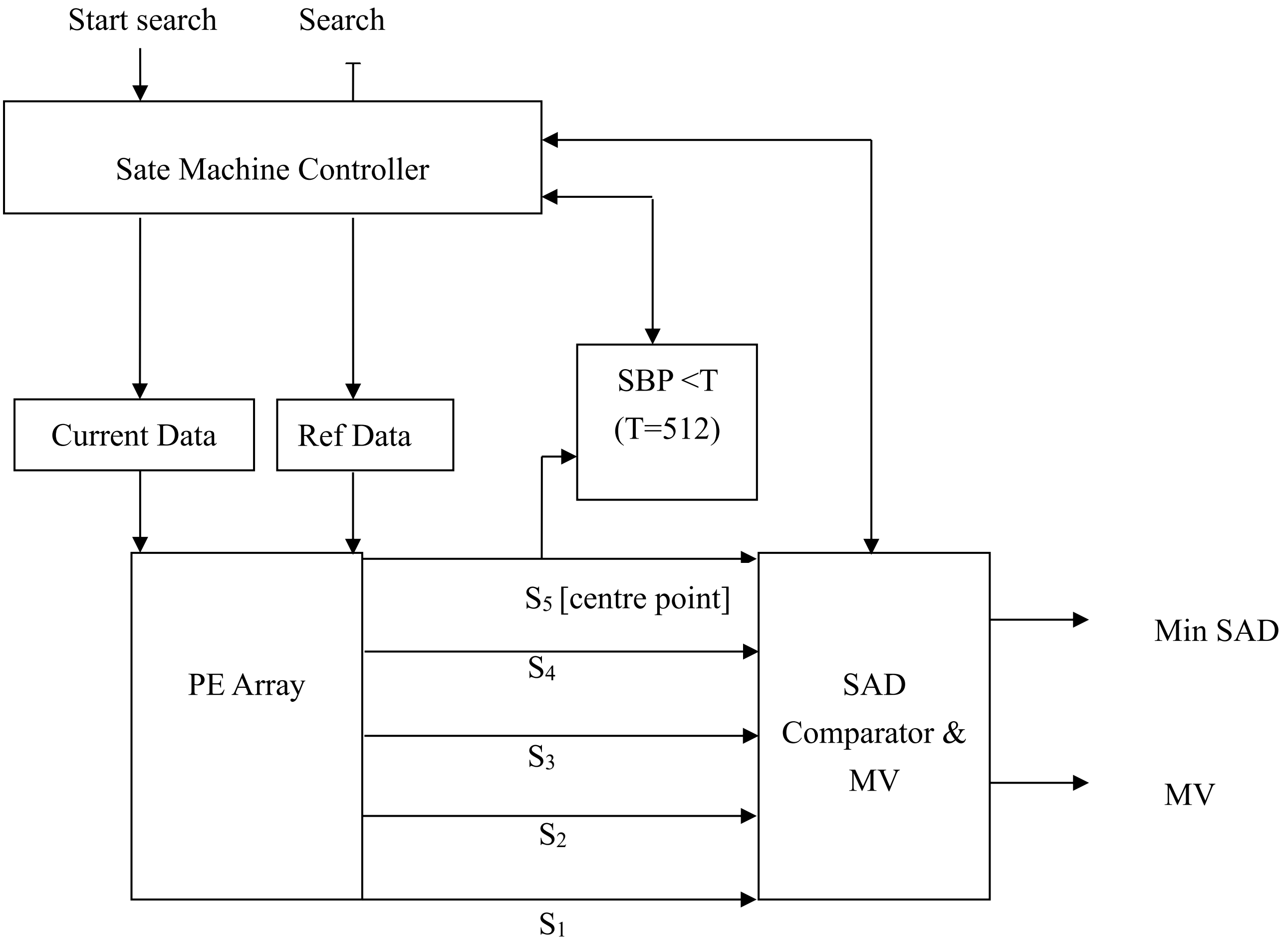


Figure.3.1. (a) Hardware architecture of Conventional method

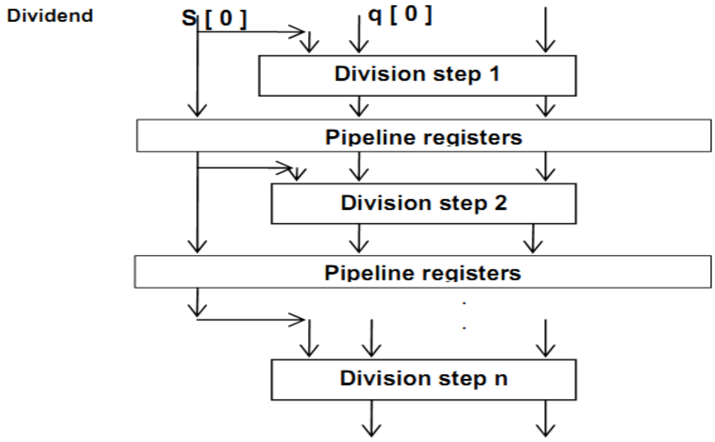
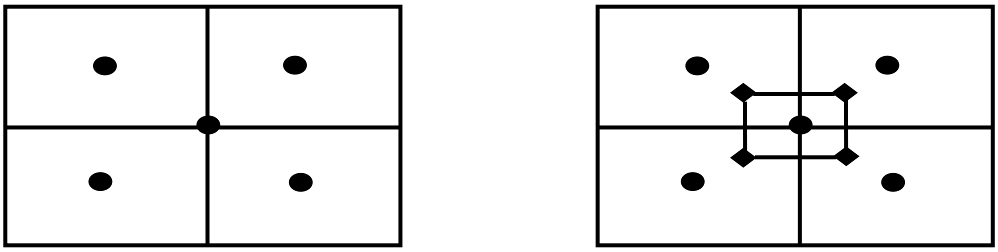


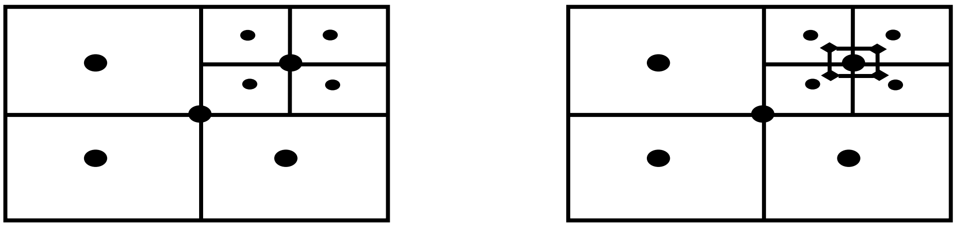
Figure.3.1. (b) Conventional single pipelining Method Flow diagram

In order to further speed up the search, Stationary Block Prediction is introduced, which is also known as Zero Motion Prejudgment. For zero motion blocks, the block distortion between the current block and reference block is very low as compared to moving blocks. Therefore to find a stationary block first step is to find the block distortion for current block with the collocated block in the reference frame. Second step is to compare it with a predetermined threshold. The current block is declared as a stationary block if the block distortion is below a predicted threshold (T). The block is assigning motion vector (0, 0) and search is terminated. The success and accuracy of this prediction depends upon the accuracy in prediction of T. In [14] author suggested a fixed T value as 512. As shown in the figure.3.2.



a) Quadratic search pattern (QSP)

b) Small Square Search Pattern (SSSP)



c) If any of one quadrant point has minimum

d) The process continue until find the motion

Figure.3.2 (a, b, c & d): Quadrant search pattern algorithm

3.3 Fast BMA for Single Reference Frame

In this section, five fast BMAs will be briefly described. All of them apply the sub-sampling technique to reduce the number of search points. They commonly have an assumption that *the BDM decreases monotonically towards the global optimal point* [16]. Therefore their searching strategies gradually converge in the direction that the minimum BDM is detected. Although this assumption does not always hold for the real world sequences, low BDM sub-optimal points can still be found if suitable strategies are used to prevent being trapped by some local minima

3.2.1 Three-Step Search Algorithm (3SS)

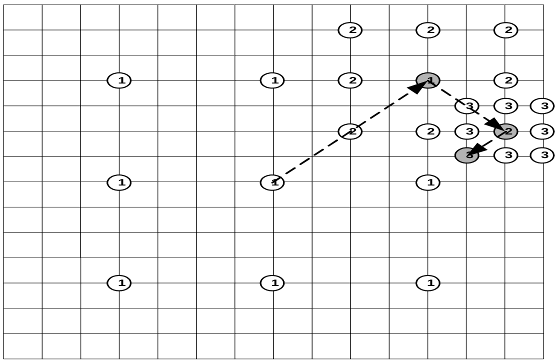
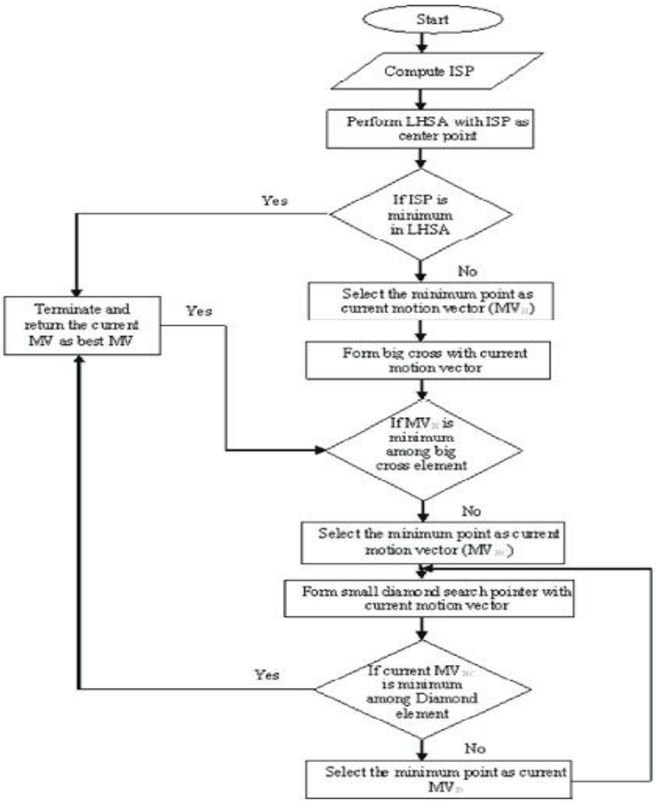


Figure 3.3: Example of 3SS

3SS was proposed by T. Koga in 1981 [17]. It applies a coarse-to fine approach that assumes the motion vectors of video sequences are uniformly distributed. As shown in Figure 3.3, the search starts from the center of the search window. It uses a nine searching point's grid for each step. The initial step size is equal to $\lceil w/2 \rceil$ where w is the window size and $\lceil\ \rceil$ is a round up function. Each successive step has half the size of the last step. The minimum BDM point is defined as the center of the next step. The search will be terminated when the step size is equal to one. Proposed algorithmic flow is shown in the figure.3.4.Further the algorithm is implemented in FPGA as shown in the figure.4.5,4.6 and 4.7

Figure 3.4. Proposed Algorithmic flow.



Algorithm Flow:

Step 1: Determine the ISP centre point from the reference frame and provided to searching window of current frame block.

Step 2: find the SAD1 value for current frame by using Large Hexagon Search Pattern (LHSP) and compare with ISP value.

Step 3: If the SAD1 of the centre point in the QSPD is less than the ISP, diminished SAD1 value is considered to be the last value or else determine the minimum SAD value and set the centre point as ISP2 value.

Step 4: find the SAD2 value for current frame by using Big Cross Search Pattern (BCSP) and compare with ISP2 value.

Step 5: If the SAD2 of the centre point in the QSPD is less than the ISP2, diminished SAD2 value is considered to be the last value or else determine the minimum SAD value and set the centre point as ISP3 value.

Step 6: find the SAD3 value for current frame by using Small Diamond Search Algorithm (SDSA) and compare with ISP3 value.

Step 7: If the SAD3 of the centre point in the QSPD is less than the ISP3, diminished SAD3 value is considered to be the last value or else determine the minimum SAD value and set the centre point as ISP4 value and repeated same procedure from step6 to step7 until to get efficient motion vector(MV)value .Further optimization in divider algorithm is shown below Figure.3.5. Newton-Raphson computational division algorithm with parallel pipelining approach is introduce in the proposed and existing method in order to optimize the area overhead and delay with less power utilization as discussed below.

A. Newton-Raphson computational division algorithm:

In paper describes a single precision floating point division based on Newton-Raphson computational division algorithm. The Newton-Raphson computational algorithm is implemented using 32-bit floating point multiplier and subtractor. The salient feature of this proposed design is that the module for computing mantissa in 32-floating point multiplier is designed using a 24-bit Vedic multiplication (Urdhva-triyakbhyam-sutra) technique. 32-bit floating point multiplier, designed using Vedic multiplication technique, yields a higher computational speed, hence, is efficiently used in floating point divider. Another important feature is the efficient use of device utilization parameters and reduced power consumption. An advantage of the Newton-Raphson algorithm is the higher versatility and precision. As shown in the Figure.2.3.

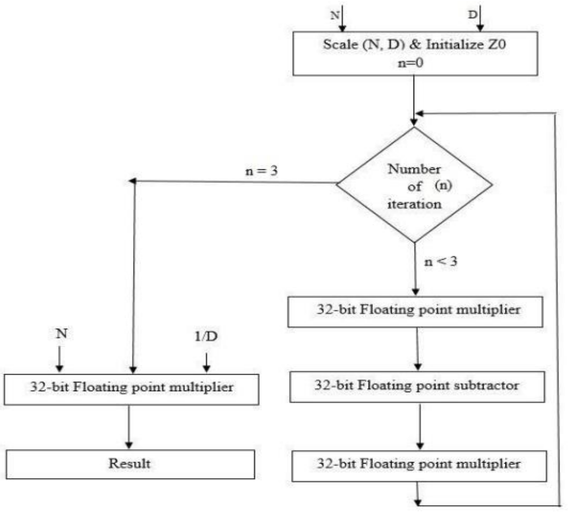


Figure.3.5.A Flowchart of the 32-bit floating point division using Newton-Raphson method.

4. RESULTS AND SIMULATION OF MULTI-FRAME MOTION VECTOR DISTRIBUTION

In this chapter, two novel fast BMAs are proposed for motion estimation of multiple reference frames. As the motion vector distributions give significant information about the real-world motion properties, which in turn affect the searching strategies in motion estimation. Some analyses on the motion vector distributions and rate distortion are conducted beforehand. Based on the results, the detailed steps of our methods are developed.[18,19]

4.1 Experiments and Analyses

Sequence	**Format**	**Motion**
Claire	CIF (352x288)	Small
Miss America	CIF (352x288)	Small
Sales	CIF (352x288)	Small
Garden	SIF (352x240)	Large
Football	SIF (352x240)	Large
Tennis	SIF (352x240)	Large

Table 4.1: Six testing video sequences

Three kinds of analyses are conducted on a series of experiments to find out the distribution of motion vectors and importance of number of reference frames. In these experiments, full search motion estimation is simulated on six CIF/SIF video sequences, Table 4.1. Three of them contain small motions, and the others contain large motions. Motion estimation is performed for 80 frames at fixed 16x16 block size with integer pixel accuracy. The search window size is set to w=7, w=15 and w=30. Ten sets of experiments are done for 1 to 10 allowed number of reference frames.

4.1.1 Distortion Gain

r	MAE per frame with used no. of reference frames = r									
Sequences	1	2	3	4	5	6	7	8	9	10
Claire	105024	102651	101288	100747	100310	99791	99434	99197	98899	98705
missA	231422	221222	218201	216380	215231	214290	213610	213081	212691	212333
Sales	287478	153003	152434	150201	150026	148979	148865	148119	148066	147539
Garden	713569	689149	685035	682286	678514	675776	674721	674346	673926	673652
Football	581333	564103	557374	551346	549514	547516	546517	545114	544445	543332
Tennis	520248	492739	484518	479861	474579	472149	470124	468311	466570	465014
Avg MAE per pixel	4.607	4.229	4.183	4.149	4.124	4.106	4.096	4.087	4.080	4.072

Table 4.2: MAE per frame for each sequence with w=7

r	MAE gain (%) per frame with used no. of reference frames = r									
Sequences	1	2	3	4	5	6	7	8	9	10
claire	N/A	2.26	3.56	4.07	4.49	4.98	5.32	5.55	5.83	6.02
missA	N/A	4.41	5.71	6.50	7.00	7.40	7.70	7.93	8.09	8.25
sales	N/A	46.78	46.98	47.75	47.81	48.18	48.22	48.48	48.49	48.68
garden	N/A	3.42	4.00	4.38	4.91	5.30	5.44	5.50	5.56	5.59
football	N/A	2.96	4.12	5.16	5.47	5.82	5.99	6.23	6.35	6.54
tennis	N/A	5.29	6.87	7.76	8.78	9.25	9.63	9.98	10.32	10.62

(i) Average	N/A	10.85	11.87	12.60	13.08	13.49	13.72	13.94	14.11	14.28
(ii) Unit of load	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00
Efficiency = (i)/(ii)		5.43	3.96	3.15	2.62	2.25	1.96	1.74	1.57	1.43

Table 4.3: MAE gain in various no. of reference frames with w=7

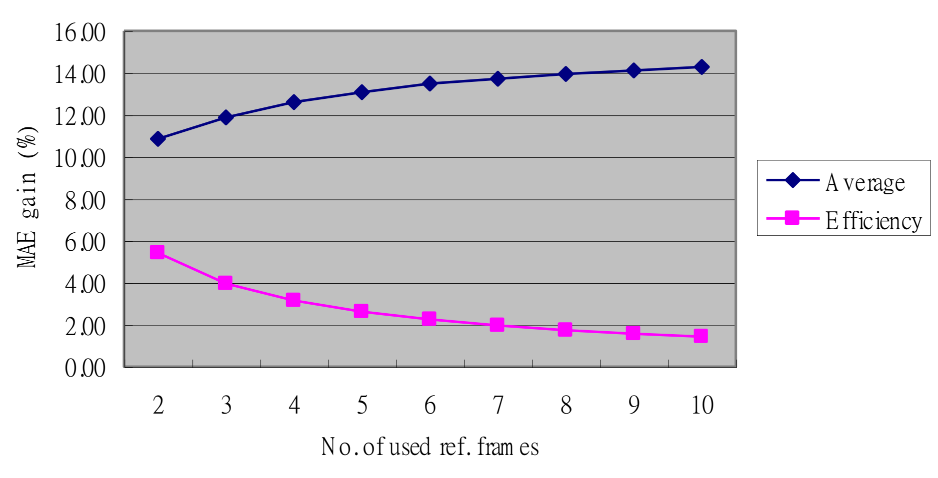


Figure 4.1: Change of MAE gain against no. of reference frames

Two curves are shown in Figure 4.1. The blue one is the average MAE gains of the six sequences. The MAE gain is defined as the relative reduction of MAE to the value counted using single reference frame. In order to represent the efficiency of increasing the number of reference frames, the MAE reduction is divided by the computational load of each frame can be optimized because of Newton-Raphson computational division algorithm with parallel pipelining approach because As the load increases linearly for increment of reference frames, the efficiency in fact decreases and tends to constant after 5 to 6 frames which can be optimized with this method. The efficiency is shown by the pink curve. The analysis reveals that the improvement is insignificant when searching more than about 5 reference frames. In contrast, the system would be burdened with the complexity. Implementation on FPGA results are shown in figure 4.5 and 4.6.

4.1.2 Motion Vector Density

n	MV distributions (%) in 10 ref. frames F(t-n)									
Sequences	1	2	3	4	5	6	7	8	9	10
Claire	64.00	8.00	9.00	4.00	3.00	4.00	2.00	2.00	2.00	1.00
missA	22.00	18.00	11.00	9.00	8.00	8.00	6.00	7.00	6.00	6.00
Sales	16.00	37.00	1.00	15.00	0.00	11.00	0.00	11.00	0.00	10.00
garden	77.00	12.00	3.00	2.00	2.00	2.00	1.00	0.00	0.00	0.00
football	53.00	14.00	5.00	12.00	2.00	4.00	1.00	5.00	1.00	4.00
tennis	61.00	19.00	6.00	2.00	4.00	2.00	2.00	1.00	1.00	2.00
Average	48.83	18.00	5.83	7.33	3.17	5.17	2.00	4.33	1.67	3.83
Accumulation	49	67	73	80	83	88	90	95	96	100

Table 4.4: Motion vector distribution in 10 reference frames

Table 4.5 shows that device utilization of our proposed approach is compared with that of previous method Full Search (FS) and Diamond search algorithm (DS). QSPD used 306 LUT's respectively than our proposed approach which uses 413 slices. Our proposed approach uses 239 slice flip flops and 186 LUTs. Maximum operating frequency of our proposed approach is 415.317MHz. Besides our proposed approach is good in total on chip power consumption as shown in the table 4.5 after integrated with parallel pipeline approach.

Attributes	FS[24]	DS[25]	QSPD	BMA
No.of slices	**14.7K**	**7.8K**	**186**	**239**
No.ofFlipFlops	**-**	**11.3K**	**226**	**153**
No.of 4			**306**	**413**
Minim			**2.408ns**	**12.205ns**
Max **(MHz)**			**415.317MHz**	**81.933Mhz**
Power(mw)	**-**	**-**	**0.187**	**0.052**
Number of PE	**16**	**16**	**5**	**6**
Search range	±4 *pixels*	±16 *pixels*	±8 *pixels*	±25 *pixels*

Table 4.5: Motion vector distribution in 10 reference frames

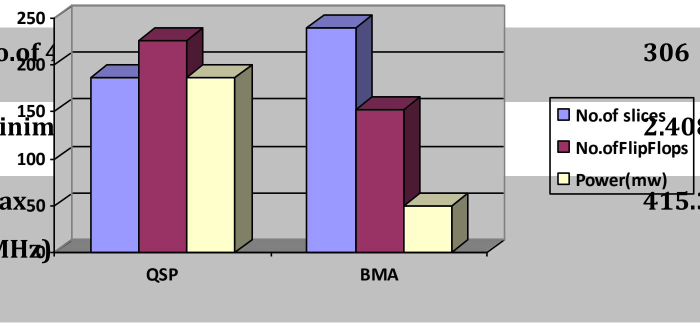


Figure.4.6 Optimization results of the existing and proposed approach.

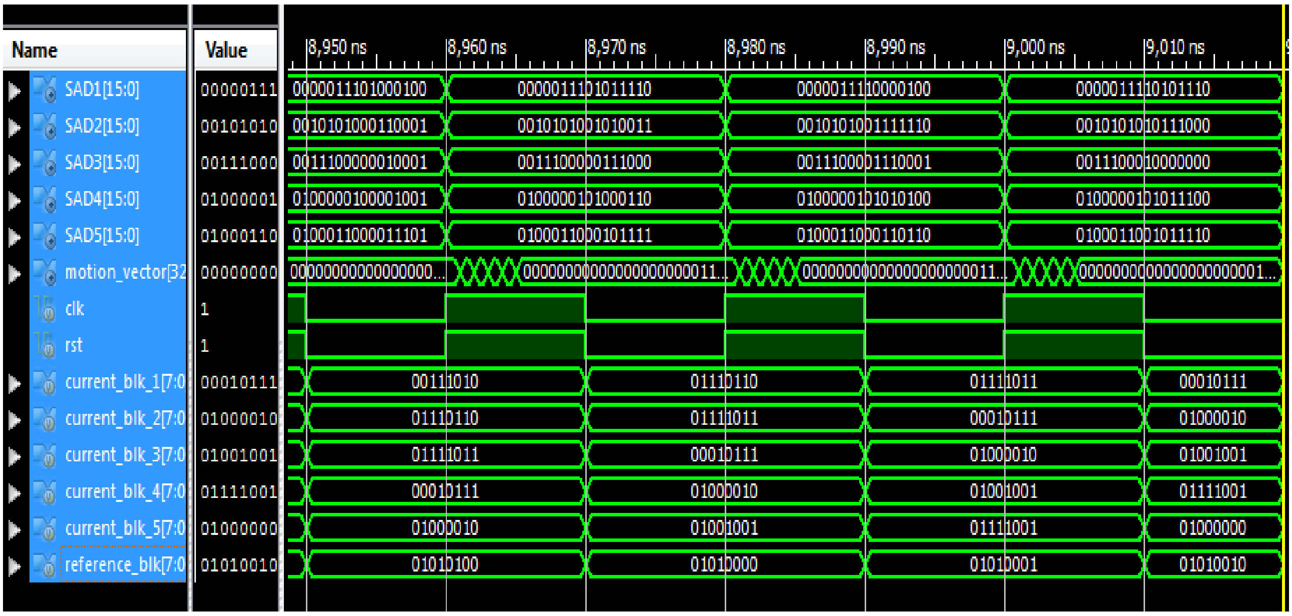
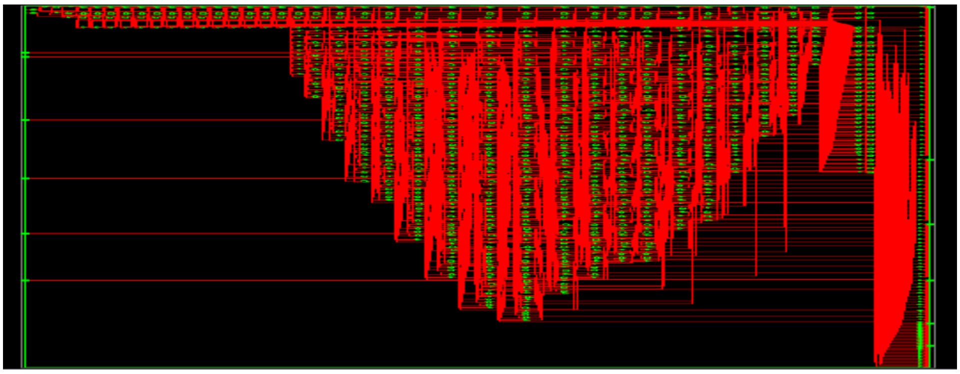


Figure 4.7: Simulation result of the conventional algorithm

Figure 4.8: Simulation result of the proposed algorithm



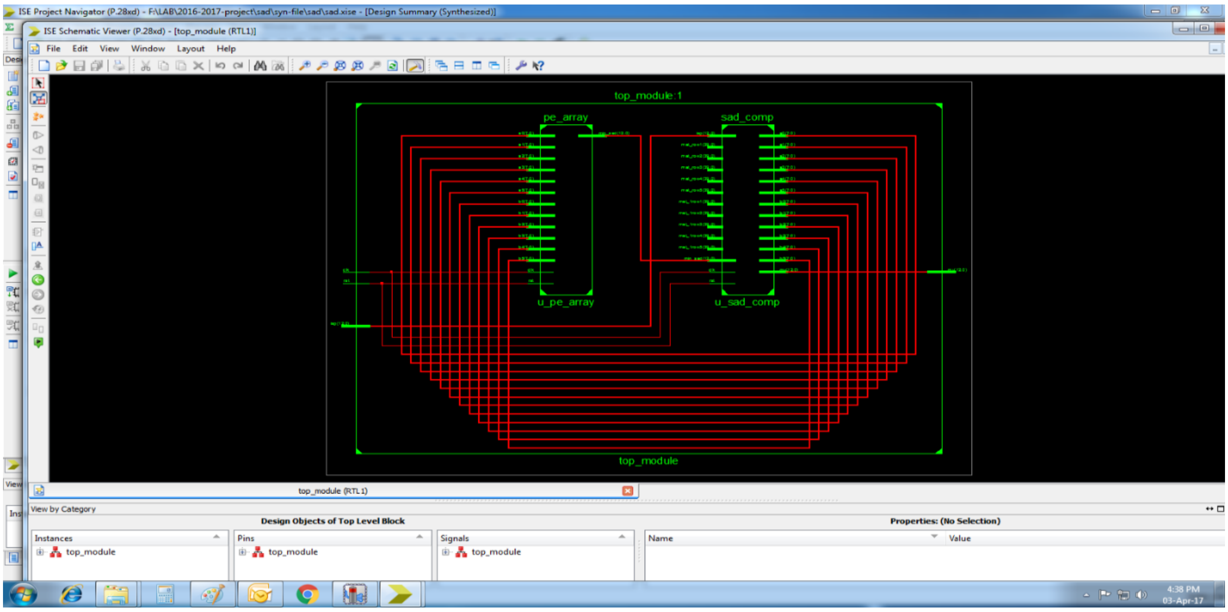
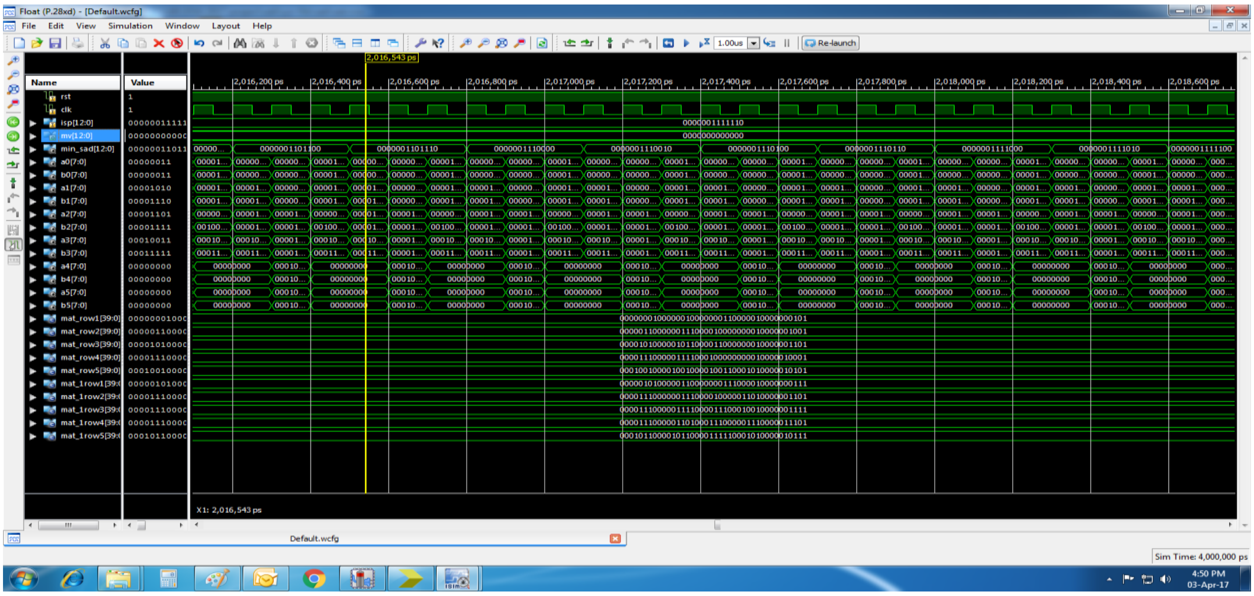


Figure.4.9: RTL schematic diagram of our conventional architecture

Figure.4.10: RTL schematic diagram of our conventional architecture



Name	Value
rst	1
clk	1
isp[12:0]	00000011111
mv[12:0]	00000000000
min_sad[12:0]	00000011011
a0[7:0]	00000011
b0[7:0]	00000011
a1[7:0]	00001010
b1[7:0]	00001110
a2[7:0]	00001101
b2[7:0]	00001111
a3[7:0]	00010011
b3[7:0]	00011111
a4[7:0]	00000000
b4[7:0]	00000000
a5[7:0]	00000000
b5[7:0]	00000000
mat_row1[39:0]	00000001000
mat_row2[39:0]	00000110000
mat_row3[39:0]	00001010000
mat_row4[39:0]	00001110000
mat_row5[39:0]	00010010000
mat_1row1[39:0	00000101000
mat_1row2[39:0	00001110000
mat_1row3[39:0	00001110000
mat_1row4[39:0	00001110000
mat_1row5[39:0	00010110000

The power report produced by the Xilinx XPower Analyzer instrument appeared in Table.4.5. Our proposed architecture devours **0.052** total on-chip power Figure 4.7 ,4.8,4.9 and 4.10 demonstrates the simulation results and RTL (Register Transfer Logic) of diagram of our proposed architecture as well demonstrates the reproduction aftereffects of the foreseen engineering. In the above mentioned table (4.1 to 4.4), motion vectors of the information outlines indicated relating to the square address of the casing furthermore the clock flag. Motion vector has been found for the input video frames by comparing the number of current frames with one reference frame as shown in the above table comparison.(4.1 to 4.4).

5. CONCLUSION

In my research, video coding techniques and the advanced prediction features of H.264 are studied. With the multiple reference frames motion compensation, a wholesale motion vector distributions analysis is conducted in both spatial and temporal domains. The experimental results show that (1) cross-center-biased remains in multiple reference frames; (2) five reference frames is an optimal for complexity and prediction quality; (3) search window size does not affect motion vector distributions. Based on the characteristics we found, two novel 3D fast motion estimation algorithms are proposed to exploit the cross-center-biased behavior in real-world sequences with Newton-Raphson computational division algorithm with parallel pipelining approach. As the rationales are supported by the analysis results, it is supposed that our algorithms will have a significant gain in the speed while keeping similar PSNR quality to FS. Implementation results of our proposed algorithm are compared with previous methods such as conventional UM Hexagon algorithm and shows higher results due parallel pipelining approach in divider. In our proposed method, we consider the Initial Search Point (ISP) to determine the motion vector efficiently. Conventional UM Hexagon technique is used to find the block with zero motion. The efficiency of the proposed algorithm is evaluated in terms of Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR). Our future work is the simulation in H.264 reference model.

References

[1]. ITU-T SG15, "Video codec for audiovisual services at px64 kbit/s", ITU-T Recommendation H.261 Version 2, Mar. 1993.

[2]. ITU-T SG16, "Video coding for low bitrate communication", ITU-T Recommendation H.263 Version 2, Feb. 1998.

[3]. JTC1/SC29/WG11, "Information technology – coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbit/s – part 2: Video", ISO/IEC 11172-2 (MPEG-1 Video), 1993.

[4]. ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information – Part 2: Video", ITU-T Recommendation H.262 - ISO/IEC 13818-2 (MPEG-2), Nov. 1994.

[5]. JTC1/SC29/WG11, "Information technology – coding of audio-visual objects – Part 2: Visual", ISO/IEC 14469-2 (MPEG-4 Visual) Version 1, April 1999.

[6]. Joint Video Team of ITU-T and ISO/IEC JTC 1, "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)", Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, Mar. 2003.

[7]. ITU, "New Video Coding Standard Delivers Major Advances in Image Quality – ITU standard

opens new opportunities for entertainment & video telecommunication", ITU, Press Release, Geneva, 23 Dec. 2002.

[8]. Shaifali M. Arora, NavinRajpal, RavinderPurwar, "Dynamic Pattern Search Algorithm with Zero Motion Prejudgment for Fast Motion Estimation", In: 2015 Fifth International Conference on Advanced Computing & Communication Technologies, pp: 138-142, 2015

[9]. Yao Nie and Kai-Kuang Ma, "Adaptive rood pattern search for fast block-matching motion estimation", IEEE Transactions on Image Courseing, vol. 11, no. 12, pp. 1442-1449, 2002.

[10]. B. Biswas, R. Mukherjee and I. Chakrabarti, "Efficient architecture of adaptive rood pattern search technique for fast motion estimation", Microprocessors and Microsystems, vol. 39, no. 3, pp. 200-209, 2015.

[11]. B. Biswas, R. Mukherjee, P. Saha and I. Chakrabarti, "An Efficient VLSI Architecture of the Enhanced Three Step Search Algorithm", J. Inst. Eng. India Ser. B, 2015.

[12]. C. Yan, Y. Zhang, J. Xu, F. Dai, L. Li, Q. Dai and F. Wu, "A Highly Parallel Framework for HEVC Coding Unit Partitioning Tree Decision on Many-core Processors", IEEE Signal Processing Letters, vol. 21, no. 5, pp. 573-576, 2014.

[13]. T. Senoh, T. Mishina, K. Yamamoto, R. Oi and T. Kurita, "Viewing-Zone-Angle-Expanded Color Electronic Holography System Using Ultra-High-Definition Liquid Crystal Displays With Undesirable Light Elimination", Journal of Display Technol., vol. 7, no. 7, pp. 382-390, 2011.

[14]. P. Helle, S. Oudin, B. Bross, D. Marpe, M.O. Bici, K. Ugur, J. Jung, G. Clare and T. Wiegand, "Block Merging for Quadtree-Based Partitioning in HEVC", IEEE Trans. on Circuits and Systems for Video Technol., vol. 22, no. 12, pp. 1720-1731, 2012.

[15]. S. Baskar,"ERROR DETECTION AND CORRECTION ENHANCED DECODING OF DIFFERENCE SET CODES FOR MEMORY APPLICATION," International Journal of Advanced Research in Computer and Communication Engineering, IJARCCE, Issue 10, 2012, 816-820.

[16]. S.Baskar, "Error recognition and correction enhanced decoding of hybrid codes for memory application" at Devices, Circuits and Systems (ICDCS), 2014 2nd IEEE Conference 1-6.

[17]. S.Baskar, "RELIABILITY ORIENTED PLACEMENT AND ROUTING ANALYSIS IN DESIGN OF LOW POWER MULTIPLIER FOR WIRELESS SENSOR NETWORKS" at International Journal of Applied Engineering Research, Volume 10,issues 44, 31384-31390.

[18]. Ambigaipalan, P., de Camargo, A. C., & Shahidi, F. (2016). Phenolic compounds of pomegranate byproducts (outer skin, mesocarp, divider membrane) and their antioxidant activities. Journal of agricultural and food chemistry, 64(34), 6584-6604.

[19]. Milano, F., & Ortega, A. (2017). Frequency divider. IEEE Transactions on Power Systems, 32(2), 1493-1501