

# **A Low Power and Error Tolerant Structure for Motion Estimation Core in H.264/AVC Standard**

M.H. Sargolzaei  
University of Sistan and Baluchestan  
Zahedan, Iran

## **ABSTRACT**

Motion estimation is widely used for removing temporal data redundancy in many video coding systems. Motion estimation core is one of the biggest and the most complex cores in many of video coding standards. Nowadays, video systems are embedded into many of portable devices; hence, they need have a trade of between their power consumption and quality of output. Hardware failure is the most important reason of quality degradation in the RT level. In this paper, two RT level low power error tolerant methods are proposed for this important video compression core. The proposed methods can be used in conjunction with other higher or lower level error tolerant and power reduction methods that have been previously proposed for this component. Experimental results show that our methods have smaller area overhead, higher reliability, and lower power consumption than the existing methods for motion estimation.

## **General Terms**

VLSI system design, Fault tolerant systems, Video coding cores.

## **Keywords**

Motion estimation core, error tolerant systems, low power design.

## **1. INTRODUCTION**

By introducing the emerging H.264/AVC video coding standard the usage of various types of video processing systems (such as digital camera, digital video broadcast (DVB), video-conferencing systems, portable video systems, Mobile TV, etc) is become more popular these days. The new compression benefit of this new standard is along with more computational needs. In addition a small fault may have a considerable effect in video quality. As a consequence, fault tolerant (and in a higher level, error tolerant,) and low power encoder design become two important objectives in designing the video coding cores. Motion estimation (ME) is the most important component in the video coding systems. That is why, many researchers are working on this module, especially on its power consumption and error resiliency.

In multimedia applications generally there are two major modules, normative and non-normative modules [1]. The normative modules are the ones that are defined in a standard structure. The output of these modules has to conform to the standard and any error in their output will cause major decoding problems. The entropy module, transform and quantization modules are in this group. On the other hand there are some modules that may affect the quality of the coded video. Although the presence of faulty components in these structures may cause some quality degradation in these modules, but the final generated bit-stream can match the standard definition. Integer motion estimation, fractional

motion estimation, mode decision, rate control are of this type that are non-normative modules.

In fault tolerant (FT) methods, effect of occurred fault will be masked completely. But, in error tolerant (ET) methods, the fault may be ignored, if its effect on video quality is less than a predefined threshold. Clearly, ET methods are well suited for the non-normative modules where erroneous output can be tolerated. The motion estimation module has a significant effect on the quality of the encoded video. Hence, an error tolerated motion estimation module will prevent drastic quality degradation in presence of faults.

Many researchers proposed methods for reducing power consumption of motion estimation cores ([2]-[8]). In their works, power consumption is reduced by decreasing the number of reference frames via input sub-sampling, fixed or adaptive search area [2], prediction based methods [3], and conservation approximation [4]. Some other works have tried to reduce power consumption with dynamic search window size [5] and dynamic pixel resolution [6] and adaptive pixel truncation [7]. In [8], authors proposed a VLSI level method to reduce power consumption of motion estimation core for real-time applications. Richmond and Ha proposed a method for low bit-rate wireless communication video networks [9].

Error tolerant researchers have focused on reducing the percentage of incorrect estimated blocks using some data redundancy algorithms such as redundant motion vector generation method [10]. The RT level error resilience researchers try to improve quality of output coded video by making a reliable datapath for motion estimation module. In [11], authors proposed an ET method for tolerating the data bus errors for motion estimation module. In [12], Varatkar and Shanbhag proposed an error resilient structure for tolerating voltage over-scaling produced errors. Their proposed structure is very similar to a version of duplication method. In [13] and [14], two built-in self-detection/correction (BISDC) methods proposed to tolerate faults of Sum of Absolute Differences module (SAD) of motion estimation core.

Dhoot and et al proposed an algorithm level fault tolerant low power motion estimation method [15]. They proposed a low power hierarchical search algorithm for motion estimation core based on probabilistic computations of SAD. Their algorithm has been limited by 0.5 dB reduction in PSNR (quality degradation).

In [16], a RT level error tolerant structure was presented for the motion estimation core. Also a mapping for hierarchical structure of Sum of Absolute Differences module (SAD) to a linear structure was presented in [16] (Fig. 1). After that, the effect of various fault locations on a linear array of Absolute Differences (AD) was mapped. Based on the hierarchical structure of the SAD, it was showed that the six higher levels

of the SAD do not need to be fault tolerated and in remaining levels, all bits of each adder do not need to be fault tolerated.

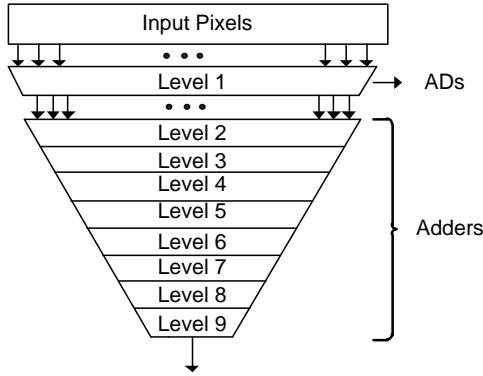


Fig. 1. SAD structure

In [16], only one tolerated fault in each adder was analyzed. If more faults occur in one adder the incorrect data will be passed in the SAD structure.

In this paper, a RT level structure for motion estimation is proposed, which not only tolerated more than one faults but also the power consumption is managed in faulty situations. Working at RT level has one important benefit, and it is its capability to use in addition of any other low power or error resilient method at algorithm or VLSI level, to reach a better design.

The paper is organized as follows. A brief description about mapping strategy and the proposed methods are presented in Section 2. The experimental results are shown and compared in Section 3. Finally, the paper ends in Section 4 with the concluding remarks.

## 2. PROPOSED METHODS

As it is described in [16], SAD module has a tree structure with 9 levels. The tree structure of SAD causes some difficulties for multiple fault tolerant modeling. Hence, a linear model for SAD is developed. In the SAD structure, each adder has a set of ADs as its sub-elements (which their output is as input of this adder). Let  $ADD[i][j]$  be the  $j$ th adder in the  $i$ th level ( $i > 1, j > 0$ ) of the tree structure and  $AD[k]$  is the  $k$ th absolute difference module in the first level of SAD tree. So  $AD[k]$  is a sub-element of  $ADD[i][j]$  if:

$$1 + (j-1) \times 2^{i-1} \leq k \leq j \times 2^{i-1} \quad (1)$$

Based on (1), all occurred faults of the adders will be mapped on the AD array (a linear array). For example, a fault in  $ADD[i][j]$  will be replaced with  $2^{i-1}$  errors in the AD array. With this mapping, an easily study on multiple faults in the SAD structure could be done. When a fault occurs in an adder or AD, it causes erroneous SAD value for appropriate pixels (they will be called faulty ADs) and the number of faulty ADs will be showed with  $N$ . When a system is fault free  $N$  is zero, but when a fault occurs in level  $i$ ,  $N$  becomes  $2^{i-1}$ . Therefore  $N$  has a strong effect on the output of the ME. In this paper, target is reduction the sensitivity of motion vector selection process to  $N$  (number of faulty ADs) and reduction the value of  $N$  for a certain number of faults. When  $N$  is not zero, the output of SAD calculates with:

$$SAD = \sum_{i=1}^{256-N} |c(i) - r(i)| + \sum_{i=1}^N |c'(i) - r'(i)| \quad (2)$$

In this section several methods are presented to reduce the effect of faults on the quality of the output video and decrease the overhead of the error tolerant method. At the first step, the relation of PSNR and the  $N$  will be showed.

### 2.1 PSNR versus N

If a component in the SAD datapath becomes faulty, it will produce incorrect data. The incorrect data propagates to the output of SAD and makes it incorrect. When incorrect data propagates to the SAD output, it is probable that the selected motion vector is not close to the best motion vector. The distance of selected motion vector with the best motion vector depends on the  $N$ , content of video stream and place of faulty ADs. When incorrect data are masked in the SAD, the ME decides on the remaining number of correct output ADs. In other words, the faulty structure of the ME is similar to a random sub-sampling block matching motion estimator module. Deciding on random input sub-sampling results is more reliable and correct than full-sampling with a faulty ME module. Therefore, if incorrect data can be masked, the sensitivity of motion vector selection process to  $N$  will be reduced. To prove this, several simulations (CIF test benches) with a H.264/AVC encoder was run. Table 1 shows the maximum value of  $N$ , where PSNR degradation is not more than 0.1 dB for both masking (M) and propagating (P) incorrect data to the output of SAD for several testcases.

Table 1. Max of  $N$  for 0.1 dB PSNR degradation

Name	M	P	Name	M	P
Bus	121	16	Student	214	54
Foreman	69	12	Crew	115	22
Stefan	90	20	Silent	146	32
City	202	20	Harbour	230	34

Based on Table 1, when an incorrect data propagates to the output of SAD (in the worst case), 12 faulty ADs causes 0.1 dB quality degradation. Whereas masking the incorrect data allows 69 faulty ADs with the same quality degradation. Therefore, with masking the incorrect data, the sensitivity of motion vector selection to the  $N$  is reduced. In the following parts two different methods for masking incorrect data and reducing the number of faulty ADs will be described. The first method is constant-forcing and the other method is bypassing faulty module method.

For fault detection and tolerance in both methods, duplication with comparison and NMR methods are used. These methods have a very small area overhead. By using other efficient methods, the area and time overhead will be reduced.

### 2.2 Constant-Forcing Method

Based on the worst number of faulty ADs in masking and propagating structures, it is not necessary to detect and correct the faults of all components in the SAD tree, because all adders have not more than 12 and 69 sub-elements. In the propagating structure, the five deepest levels must be fault tolerant but in the masking case only the two deepest levels must be fault tolerated.

One choice for removing the effect of  $N$  on the output motion vector is replacing the second part of (2) with a constant value by forcing a constant value on the output of faulty component. The value of the forced constant value is not important because it is ignored in the comparison process because all SAD outputs have the same number of constant values. When an adder becomes faulty, its output is forced to zero (as a

constant value, zero is chosen). In this method, the effect of a fault is similar for all blocks in the search area.

If an adder in level  $i$  becomes faulty, the output of  $2^{i-1}$  ADs will be ignored in calculation of the SAD output. In this case, block matching motion estimation is based on  $256 \cdot 2^{i-1}$  input sub-sampling. Hence  $2^{i-2}$  modules are uselessly working and consuming power. To avoid power consumption of these modules the input of their AD sub-elements must be forced to zero. The zero forcing on the input of the ADs removes all transitions of the useless components. This low power version will be called LPCF method.

Faults on components in levels 1-4 are not necessary to be detected, because the number of their sub-elements is less than 12. Whereas components in levels 5-7 have been fault detected but not fault tolerated because the number of their sub-elements are in the range of 12 to 69 and fault in these modules cannot degrade PSNR more than 0.1 dB. Finally adders in the two deepest levels have to become fault tolerant because the number of their sub-elements is more than 69.

### 2.3 Bypassing Faulty Module Method

In the Constant-Forcing (CF) method, if an adder in level  $i$  becomes faulty, the output of the  $2^{i-1}$  ADs cannot contribute in the calculation of the SAD output. In this part another way to mask incorrect data with less N for a certain number of faults is presented.

Let an adder in level  $i$  becomes faulty; this adder has  $2^{i-2}$  useless correct modules in its domain. For reviving their effect on the output of SAD, another correct adder must be replaced instead of the faulty adder. But half of the correct modules can be revived, if the output of the faulty module connects to one of its inputs. In this case the output of the  $2^{i-2}$  ADs is only ignored in the calculation of the SAD output.

The maximum N in this method is the same as CF method and it is equal to 69 ADs. If an adder in level 8 becomes faulty, only 64 ADs are ignored in the SAD output calculation process. Therefore for tolerating one fault in the SAD structure, it is necessary to tolerate faults of the final deepest level adder in the SAD. But all components in levels 5-8 must be equipped with the fault detection method for detecting faults. In this case, the number of faulty ADs will be half that of the CF method for a similar level of error tolerance. This low power version of BFM method will be referred by LPBFM method.

## 3. RESULTS AND DISCUSSION

To evaluate the proposed error tolerant methods in terms of area, power consumption a VHDL model of SAD module were developed. For evaluating reliability of our methods a C++ platform were developed.

In tables in this section STMR refers to soft TMR method. In this method, all levels of SAD are not equipped with fault detection and correction methods and only levels 5-9 are fault tolerant (based on results in [16]).

Results of proposed methods were compared with two references [15] and [16]. Area overhead, power consumption and reliability are factors those are compared with results of [16] and possible extractable parameters of [15]. In [15], authors proposed an algorithm level method to tolerate the produced errors with the voltage over-scaling low power method. It should be noted that since they have used prediction based on input sub-sampling, the results are typically less accurate. In addition, the failing condition in their work has been considered to be 0.5 dB, but in this paper

it was considered 0.1 dB as limitation of degradation in quality of video coding.

### 3.1 Area Overhead

By increasing using of portable video coding systems, area overhead is became an important factor for evaluating an error tolerant and power reduction method. On the other hand, area cost has a direct effect on the manufacturing cost, power consumption, and failure rate of the system.

Based on results of [15], considering 0.1 dB PSNR degradation, the area overhead will be close to 30%. Whereas the area overhead of CF and BFM methods is around 8% and after applying bit-wised error tolerance that it is proposed in [16], it is closed to 2%. Area overhead of all methods is shown in Fig. 2. In Fig. 2, 'CF' is constant-forcing method, 'BCF' is bit-wised error tolerant constant-forcing method (applying of [16] on CF method), 'BFM' and 'BBFM' (applying of [16] on BFM method) are bypassing faulty module method and its upgraded version. 'LPCF' and 'LPBFM' are low power versions of 'BCF' and 'BBFM' structures.

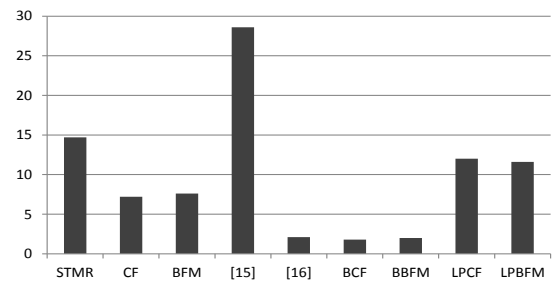


Fig. 2. Area overhead comparison of various structures

The area overhead of Fig. 2 is for tolerating one fault. Some applications may require more reliability (especially in portable devices). At a higher level, the area overhead depends on the number of tolerated faults. For tolerating one, two or three faults in SAD, each level must be fault tolerant for several number of faults. The needed number of tolerated faults in each level is shown in Table 2. In this table, '\*' shows that in this level fault detection method is sufficient. Reported values indicate the number of tolerated faults in each level.

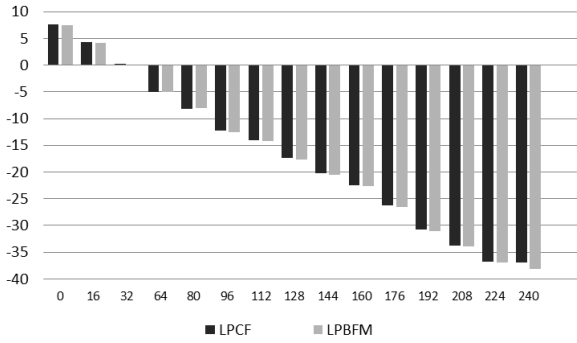
Table 2. High level area cost for various tolerated faults

Level	STMR			CF			BFM		
	1	2	3	1	2	3	1	2	3
3	0	0	1	0	0	0	0	0	0
4	0	1	2	0	0	0	0	0	0
5	1	2	2	*	*	*	*	*	*
6	1	2	3	*	*	1	*	*	*
7	1	2	3	*	1	2	*	*	1
8	1	2	3	1	2	3	*	1	2
9	1	2	3	1	2	3	1	2	3

### 3.2 Power Consumption

The video coding applications are wide and many of them have a low power requirement. To evaluate power consumption of the mentioned structures, pseudo random inputs are applied to the VHDL models. The power consumption overhead with respect to that of the normal structure is 14.04% for the STMR method, 8.5% for the CF method, 3% for the BCF method, 7.89% for BFM method, and for BBFM method it is about 2.78%. Power consumption

of the LPCF and LPBFM methods depends on the number of faulty ADs. As shown in Fig. 3, power consumption decreases by increasing number of ADs.



**Fig. 3. Power consumption comparison of various methods**

Power reduction for the LPCF method is close to LPBFM because of their similar power reduction strategy and area cost. Based on minimum, average and maximum number of faulty ADs in Table 1, power consumption can be reduced to 3.12%, 17.42% and 36.63%, respectively.

These fault tolerant low power methods can also be used to reduce power consumption of motion estimation core in fault-free state as a normal low power method.

Reduction power consumption that proposed in [15] is based on voltage scaling method of motion estimation core. Proposed method in [15] improved 13%-15% over voltage scaling method.

### 3.3 Reliability Modeling

To evaluate reliability of the proposed methods, several C++ programs have been developed. In the developed programs, based on a predefined fault density, appropriate number of faults is injected in the SAD, and N is checked. The failure condition is defined based on the minimum number of acceptable faulty ADs in Table 1 (N = 69).

To inject faults in SAD, two parameters must be indicated: failure rate of each component in each level and average number of injected faults in each clock cycle. The probability of fault occurrence is considered to be monotonic throughout the chip. Hence, the percentage of occurring faults in each component adds up to the ratio of the number of its gates to the total number of gates. The total number of gates is estimated for each structure with Equations (3)-(6), where  $A_i$  refers to the area cost of each component in level  $i$ . The area cost of adder, voters, comparators and multiplexers are considered in  $A'_i$ .

$$A_{Normal} = \sum_{i=1}^9 2^{9-i} \times A_i \quad (3)$$

$$A_{STMR} = \sum_{i=1}^4 2^{9-i} \times A_i + \sum_{i=5}^9 2^{9-i} \times A'_i \quad (4)$$

$$A_{CF} = \sum_{i=1}^7 2^{9-i} \times A_i + \sum_{i=8}^9 2^{9-i} \times A'_i \quad (5)$$

$$A_{BFM} = \sum_{i=1}^8 2^{9-i} \times A_i + A'_9 \quad (6)$$

A single fault tolerated system was designed in C++ and 20,000 permanent faults were injected in that, in various number of clock cycles. Variation in the occurred fault density simplifies the reliability comparison between different structures. Table 3 shows the percentage of system failures for several fault density (fault density: number of occurred faults in a clock cycle in average) and structures.

**Table 3. Percentage of system failures**

Fault Density	Normal	STMR	[16]	CF	BFM
20	98.2	92.3	76.8	18.6	2
10	81.57	30.85	54.2	3.95	0.1
3	29.26	0.48	10.7	0.22	0
2	12.38	0.06	6.8	0.11	0
1	5.72	0	2.9	0	0
0.2	1.1	0	0	0	0

## 4. CONCLUSION

Low power and error resilient methods are two important subjects in VLSI system designs such as video coding systems. Motion estimation is an important part of many video coding standards. Sum of Absolute Difference (SAD) is the biggest part of datapath of the motion estimation module. In this paper, two low power error tolerant methods for hardware implementation for SAD are proposed. The proposed error tolerant methods can tolerate any permanent fault in the SAD structure. Low power version of the proposed methods is also presented. Experimental results show that the proposed methods have a smaller area overhead, low power consumption and higher reliability. Proposed method in this paper can be used only to reduce power consumption of SAD module in fault-free state. But, the proposed methods can be applied to other tree-based computational modules too.

## 5. REFERENCES

- [1] P. Kuhn, Algorithms, complexity analysis and VLSI architectures for MPEG-4 motion estimation. Kluwer Academic Publishers, 1999.
- [2] J. Minocha, and N. R. Shanbhag, S. 1999. A low power data-adaptive motion estimation algorithm. In Proceedings of IEEE Workshop on Multimedia Signal Processing.
- [3] B. Zeng, R. Li, and M. L. Liou, 1997. Optimization of fast block motion estimation algorithms. IEEE Transaction on Circuits and Systems for Video Technology. 7 (Dec. 1997), 833-844.
- [4] V. L. Do, and K. Y. Yun, 1998. A Low-Power VLSI Architecture for Full-Search Block-Matching Motion Estimation. IEEE Transaction on Circuit and System for Video Technology. 8 (Aug. 1998), 393-398.
- [5] S. Saponara, and L. Fanucci, 2004. Data-adaptive motion estimation algorithm and VLSI architecture design for low-power video systems. IEE Journal of Computers and Digital Techniques. 151 (Feb. 2004), 51-59.
- [6] S.H. Wang, S.H. Tai, and T Chiang, 2009. A low-power and bandwidth-efficient motion estimation IP core design

- using binary search. *IEEE Transaction on Circuits and Systems for Video Technology*. 19 (May 2009), 760-765.
- [7] Z.L. He, C.Y. Tsui, K. K. Chan, and M. L. Liou, 2000. Low-power VLSI design for motion estimation using adaptive pixel truncation. *IEEE Transaction on Circuits and Systems for Video Technology*. 10 (Aug. 2000), 669–678.
- [8] H. Kaul, M.A. Anders, S.K. Mathew, S.K. Hsu, A. Agarwal, R.K. Krishnamurthy, and S. Borkar, 2009. A 320 mV 56  $\mu$ W 411 GOPS/Watt Ultra-Low Voltage Motion Estimation Accelerator in 65 nm CMOS. *IEEE Journal of Solid-State Circuits*. 44 (J. 2009), 107-114.
- [9] R. Steven Richmond II, and D.S. Ha, 2001. A low-power motion estimation block for low bit-rate wireless video. In *Proceedings of the International Symposium on Low Power Electronics and Design*.
- [10] M.B. Dissanayake, C.T.E. Hewage, S.T. Worrall, W.A.C. Fernando, and A.M. Kondo, 2008. Redundant motion vectors for improved error resilience in H.264/AVC coded video. In *Proceedings of International Conference on Multimedia and Expo*.
- [11] H. Chung, and A. Ortega, 2005. Analysis and testing for error tolerant motion estimation. In *Proceeding of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*.
- [12] G. V. Varatkar, and N. R. Shanbhag, 2008. Error-Resilient Motion Estimation Architecture. *IEEE Transactions on Very Large Scale Integration Systems*. 16 (Oct. 2008), 1399-1412.
- [13] C.L. Hsu, C.H. Cheng, and L. Liu, 2010. Built-in Self-Detection/Correction Architecture for Motion Estimation Computing Arrays. *IEEE Transactions on Very Large Scale Integration Systems*. 18 (Feb. 2010), 319-324.
- [14] K.K. Ghouse and S. A. Mastani, 2012. Built in Self-Test for SAD Module in Motion Array Detection. *International Journal of Electronics Signals and Systems*. 2 (2012), 126-130.
- [15] C. Dhoot, V.J. Mooney, S.R. Chowdhury, and L.P. Chau, 2011. Fault tolerant design for low power hierarchical search motion estimation algorithms. In *Proceeding of International Conference on VLSI and System-on-Chip*.
- [16] M.H. Sargolzaie, M. Semsarzadeh, M.R. Hashemi, Z. Navabi, 2010. Low cost error tolerant motion estimation for H.264/AVC standard. In *Proceeding of IEEE East-West Design & Test Symposium*.