ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

FPGA Based Low Power Motion Estimation with EDDR Architecture for Mobile Video Applications

[I]T.Marish, [II]V.Karthik
[I]Final Year, M.E. – VLSI DESIGN, Sethu Institute of Technology, Kariapatti, India
[II]Asst. Professor, ECE Department, Sethu Institute of Technology, Kariapatti,India

Abstract
Motion estimation is the technique of finding a suitable Motion Vector (MV) that best describes the movement of a set of pixels from its original position within one frame to its new positions in the subsequent frame. This paper presents a flexible and scalable motion estimation processor capable of supporting the processing requirements, which is suited for FPGA implementation. This project predicts and allocates the appropriate data bandwidth for motion estimation under a limited bandwidth supply suitable for dynamically changing bandwidth supply. Reliability and throughput of motion estimation will be improved by using bandwidth scalable motion estimation architecture. This project reliably enhances the fault coverage without performance degradation and reduces the power consumption and also cost of hardware overhead.

Keywords
Motion Estimation (ME), Processing Elements (PEs), TCG, RQ code, EDC, DRC.

I. Introduction

Video coders compress digital video sequences by removing redundancies. The most important temporal redundancy is typically reduced by motion estimation and motion compensation which encodes the differences between intensity values in the current frame and those of their counterparts in the reference frame that has been translated by an estimated motion vector [1]. In most video coding standards, motion estimation is based on blocks. A video frame is divided into nonoverlapping macroblocks [2], typically of size 16*16 pixels. Each macroblock is compared to candidate blocks within a search area in the reference frame. This process is referred to as the block matching algorithm (BMA). Various distortion measures could be used for finding the best match for a macroblock in the motion estimation process.

Mean squared error (MSE), mean absolute error (MAE), and sum of absolute differences (SAD) [3] are commonly used. Recently, a new distortion measure for motion estimation has been proposed- the sum of absolute transformed differences (SATD) [4]. This measure sums the frequency transform coefficients, typically the Hadamard transform of the differences between the pixels in the template macroblock and the corresponding pixels in a candidate block. SATD is considerably slower than the SAD but it more accurately predicts quality from the viewpoints of both objective and subjective metrics. Therefore, it is used in the H.264 reference model software [5], as well as in other new video encoders. Motion estimation, although efficient in reducing temporal redundancy, incurs high computational complexity.

A full search technique for finding the best matching region within the search area in the reference frame is usually impractical for real-time applications due to the large number of comparisons required. Thus, many alternative "fast search" motion estimation algorithms have been proposed in the literature [6]. The main concepts of these fast algorithms can be classified into six categories: reduction in search positions, predictive search, simplification of matching criterion, bitwidth reduction, hierarchical search, and fast full search. The most popular category is the reduction in search positions. Algorithms in this category reduce search complexity by limiting the number of candidate blocks [7]. These algorithms rely on the assumption that the matching error monotonically increases with the distance from the optimal position (having minimum distortion). This assumption is not always valid and the process may converge to a local minimum on the error surface rather than the global minimum as in the full search algorithm. Well-known algorithms in this [8]category are the 2-D logarithmic search, three-step search, four-step search, cross search, diamond search, and center-biased diamond search. Diamond search based algorithms have significantly better performance in speed and quality than prior algorithms. However, due to its simplicity, three-step search is still commonly used.

Predictive motion estimation, for example [9,10] utilize the motion information in the spatial and/or temporal neighbouring macroblocks to form an initial estimate of the current motion vector. Thus, it can effectively reduce the search area as well as the computation. Another approach for fast motion estimation is to speed up the calculation of matching error for each candidate block independently. This is usually achieved by subsampling the pixels in the template and candidate blocks [11, 12]. Finding the optimal match with minimum matching error using this technique is, however, not guaranteed. This approach may be combined with the former two techniques to limit the number of search positions and to predict the current motion vector.

A different approach for fast motion estimation uses simple matching criteria to reject search positions while ensuring the global minimal matching error can still be attained. Only candidate blocks that have not been disqualified are further processed using more precise distortion calculations. Using an appropriate test, many search positions may be excluded from being further considered in the motion vector search, thus reducing search complexity significantly. Reduction of computation and memory access by using pixel truncation presented in [13], these designs are also presented for reducing or considering the computational complexity for limited resource environments.

II. Proposed Method

The proposed bandwidth scalable ME integrated with the full search ME engine in the proposed ME architecture design as shown in fig. 1. This architecture consists of two major blocks: Bandwidth scalable ME controller and ME engine. First, the bandwidth scalable ME controller decides the SR of the current MB. The data request of SR data and current MB are loaded from external memory by using memory controller and are stored in the current pixel buffer and reference pixel buffer.

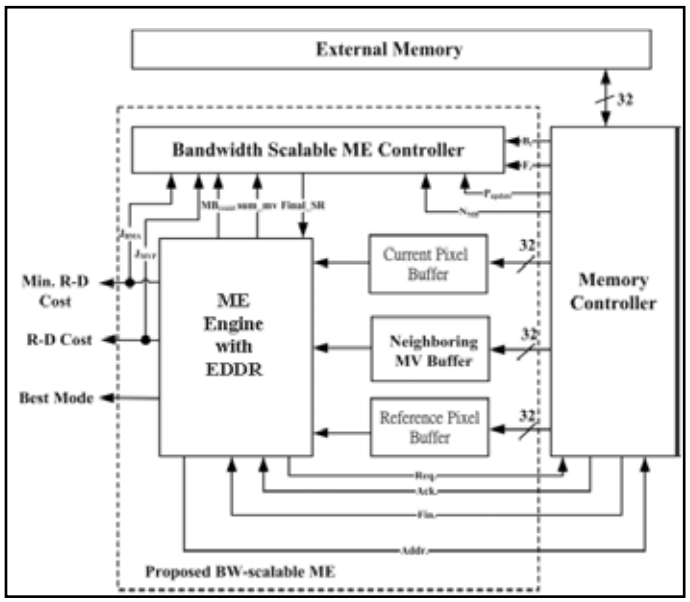


Fig. 1: Block Diagram of Bandwidth scalable Motion Estimation

In ME engine, SAD generation modules, mode decision module pre-retrieval control unit are used to calculate the best mode. Pre-retrieval control unit issues the data request of SR data and current macroblocks. The data in the current pixel buffer and reference pixel buffer are used to calculate SADs through the SAD generation module 1 and 2, both can be done at each cycle. The control unit simultaneously produces control signals to the two SAD generation modules, motion vector predictor generator and the pre-retrieval control unit.

Additionally, it needs to count the number of MBs (MB_{count}) up to the current coded kth MB. Handshaking policy takes place between the pre-retrieval control unit and the memory controller is in the order of request-acknowledge- address-finish significantly. The pre-retrieval control unit must send a request signal to the memory controller for the sake of data processing. If the memory controller is idle, it will be activated and respond with an acknowledge signal to the pre-retrieval control unit. When the pre-retrieval control unit receives the acknowledge signal, it generates a memory address for the memory controller to load the required reference pixels for the following step. After the task is finished, the memory controller sends a finish signal and waits for the next request. The mode decision module selects the final best mode and sent as output along with its motion vector difference. For SR decision of the next MB, the corresponding R-D cost data like J_{BMA} and J_{MVP}, the best MB mode, and the sum_mv are sent back to the bandwidth scalable ME controller. In this method, EDDR architecture is merged into ME engine for detecting and correcting the errors. The error detection and data recovery circuit as shown in fig.2.

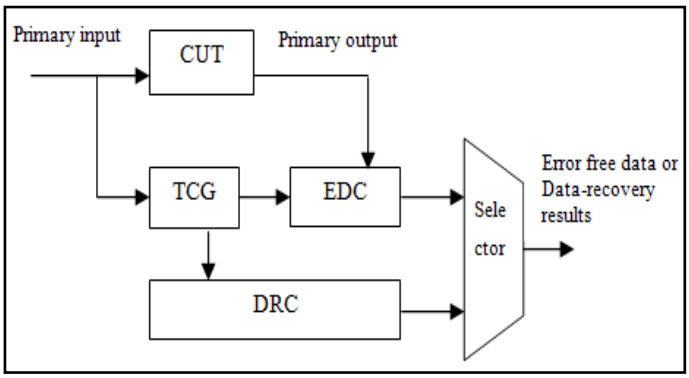


Fig. 2: Block diagram of EDDR architecture

The EDDR scheme comprises two major circuit designs, i.e. error detection circuit (EDC) and data recovery circuit (DRC), to detect errors and recover the corresponding data in a specific circuit under test (CUT). The test code generator (TCG) utilizes the concepts of RQ code to generate the corresponding test codes for error detection and data recovery. In other words, the test codes from TCG and the primary output from CUT are delivered to EDC, to determine whether the CUT has errors. DRC is in charge of recovering data from TCG. Additionally, a selector is enabled to export error-free data or data-recovery results. This work adopts the systolic ME as a CUT to demonstrate the feasibility of the proposed EDDR architecture. ME consists of many PEs incorporated in a 1-D or 2-D array for video encoding applications.

A. Fault model

The PEs are essential building blocks and are connected regularly to construct ME. Generally, PEs are surrounded by sets of ADDs and accumulators that determine how data flows through them. PEs testing assignment can be easily achieved by using the fault model, cell fault model (CFM). Using CFM has received considerable interest due to accelerated growth in the use of high-level synthesis, as well as the parallel increase in complexity and density of integration circuits (ICs). Arithmetic modules are designed in an extremely dense configuration due to their regularity.

B. TCG design

According to Fig. 2, TCG is an important component of the proposed EDDR architecture. Notably, TCG design is based on the ability of the RQCG circuit to generate corresponding test codes in order to detect errors and recover data. The specific *PE* in Fig. 2 estimates the absolute difference between the Cur_pixel of the search area and the Ref_pixel of the current macroblock. TCG is the combination of PE and RQCG. It will produce RT and QT value and given to EDC circuit to detect errors. In the existing system residue and quotient value are found in a separate circuit. It produces residue and quotient value only in 22 clock cycles. Division can be done by continuous subtraction. We can use a combination of subtraction unit and calculate residue and quotient value in reduced clock cycles, so the testing time will be greatly reduced.

C. EDDR circuit

The outputs between TCG and RQCG are compared in EDC (Error Detection Circuit) in order to determine whether errors have occurred. If the values of R_{PE} = RT and Q_{PE} = QT, then the errors in a specific PE can be detected. The EDC output is then

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

used to generate a 0/1 signal to indicate that the tested PE is error free/erroneous. DRC (Data Recovery circuit) is in charge of recovering data from TCG.

D. Memory Controller

ME controller receives these R-D cost data along with number of MBs, P_{update} which denotes the period to update the available system bandwidth depends on the system condition, Br denotes the data rate (bytes/s) allocated by the memory controller, Fr denotes the coded frame number per second. P_{update} is set from one frame to multiple frames. Notably P_{update} can be changed at any time to adapt for changing the bandwidth abruptly. Dynamically SR can be determined by bandwidth scalable ME controller. And also the memory address to the pre-retrieval control unit to load the required reference pixels for ME search can be generated. This architecture accepts R-D cost data of J_{BMA} and J_{MVP} from ME engine and system information from the memory controller to generate the appropriate SR for ME coding.

There are six major tasks presented in the proposed architecture of the bandwidth scalable ME controller. They are BW initialization, BW efficiency calculation, BW prediction, BW allocation, SR prediction, and Final SR prediction shown in fig. 3. These tasks take five cycles to complete.

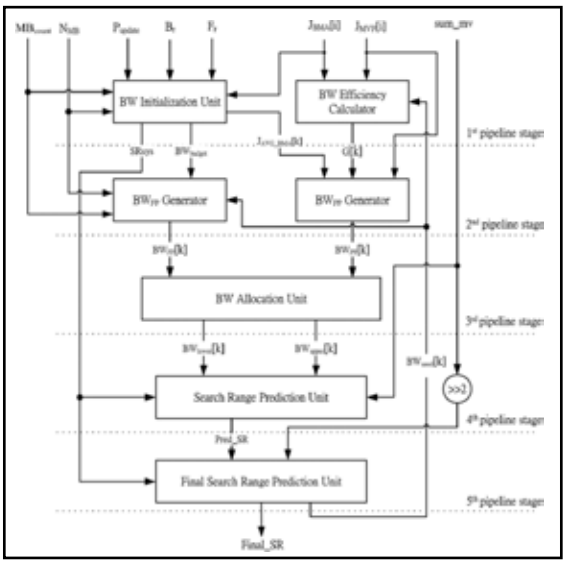


Fig. 3: Block diagram of the proposed bandwidth-scalable ME controller

The main idea of the proposed algorithm is to allocate the available bandwidth in R-D optimized sense within the given bandwidth budget. During this allocation, the R-D performance should be kept as smooth as possible for consecutive coding, while the R-D performance should also be maximized within the available bandwidth. However, the true consumed bandwidth is unknown before the real coding process occurs. A reasonable way to realize this task is based on using the statistics of past bandwidth usage and R-D performance to predict the possible bandwidth usage of the next coding unit. With the above constraints and prediction, an appropriate bandwidth can be allocated to meet the requirements. The operation mode with the greatest coding efficiency should be selected, because it also indicates a better R-D performance gain with equal bandwidth consumption.

Thus, the whole algorithm can be divided into four steps as follows:

Step 1: For bandwidth allocation, Initialize the memory bandwidth budget (BW_{budget}) within P_{update} and is calculated by

$$BW_{budget} = \left(2 \times SR_{sys} + 16\right)^2 \times N_{MB} \quad (1)$$

where N_{MB} is the total number of MBs within P_{update}. Thus, the global system SR (SR_{sys}) for the ME hardware is

$$SR_{sys} = \text{Floor}\left\{\left(\sqrt{\frac{BW_{budget}}{N_{MB}}} - 16\right)\right\} \quad (2)$$

P_{update} can be changed abruptly with respect to the available bandwidth; new value can be set as well to force lower bandwidth usage when necessary.

Step 2: Calculate the bandwidth efficiency for current coded kth MB for justifying the coding efficiency. The bandwidth efficiency G[k] which represents the R-D gain per unit bandwidth can be expressed as

$$G[k] = \frac{\sum_{i=0}^{k-1}(J_{MVP}[i]-J_{BMA}[j])}{BW_{used}[k]} G[k] = \frac{\sum_{i=0}^{k-1}(J_{MVP}[i]-J_{BMA}[j])}{BW_{used}[k]} \quad (3)$$

where BW_{used} denotes the accumulated used data bandwidth up to the (k-1)th MB.

Step 3: Find the past BW prediction (BW_{PP}) for the current coded MBs and future BW prediction (BW_{FP}) for the remaining MBs. The required bandwidth of the current coded MB can be predicted with the help of average bandwidth efficiency having quality smoothness constraint. R-D gain should be changed slowly between neighbouring coded MBs by involvement of quality smoothness constraint. Therefore, the past and future BW prediction can be represented as

$$BW_{pp}[k] = \frac{J_{MVP}[k] - avg\left(\sum_{i=1}^{k-1} J_{BMA}[i]\right)}{G[k]} \quad (4)$$

$$BW_{pp}[k] = \frac{BW_{budget} - BW_{used}[k]}{N_{MB}-(k-1)} \quad (5)$$

An available bandwidth interval can be determined by using the future and past predicted bandwidth. In this case, two possible conditions occur. They are if the future BW prediction is greater than the past BW prediction, this implies that less bandwidth had been allocated to the previous MBs and thus more bandwidth can be allocated to the next MB.

In contrast, the other case implies that too much bandwidth had been allocated to the previous MBs, and hence, less bandwidth can be allocated to the next MB. These conditions are applied for defining the lower and upper bound of the bandwidth usage in the BW generator as shown in fig. 3.

Step 4: Determine the predictive SR (Pred_SR) according to three bandwidth modes:

- Previous MB falls outside the BW interval bounded by BW_{upper}- Mode BW_H
- Previous MB falls within the BW interval- Mode BW_N
- Previous MB falls outside the BW interval bounded by BW_{Lower}- Mode BW_L.

The R-D performance is optimized by increasing the SR with the sufficient usage of bandwidth supply. Further, the motion vector information of the neighboring coded blocks is used to refine the Search range. The adjacent motion vectors (MVs) from neighboring blocks related to current block such as the motion vector of the neighboring upper MB ($MV_{U)}$, upper right MB ($MV_{UR)}$, left MB ($MV_{L)}$, and upper left MB of the current coded MB ($MV_{UL).}$ When MV_{UR} is absent, on that situation MV_{UL} is used. Then, we compare these first three MVs (MV_{U}, $MV_{L,}$ MV_{UR})

and the summation of the absolute values of these three MVs (sum_mv). Finally the Search Range is determined as

$$\text{Final\_SR} = \min\left(\text{SR}_{sys}, \max\left(\text{Pred\_SR}, \frac{1}{4} \times \text{sum\_mv}\right)\right) \quad (5)$$

Where the SR_{sys} is used as the upper bound of the allocated SR to avoid bandwidth usage overflow. It can be easily adapted to sudden bandwidth change by setting new system SR. To reduce the critical path delay, these major tasks are pipelined into five stages. For low hardware cost and sake of simplicity bandwidth allocation unit, SR prediction unit and bandwidth efficiency calculator are implemented with subtractors, shifters, adders, comparators and one divider only.

III. Simulation Results

The proposed architecture is designed using verilog HDL, simulated using modelsim software and synthesized using Xilinx project navigator.

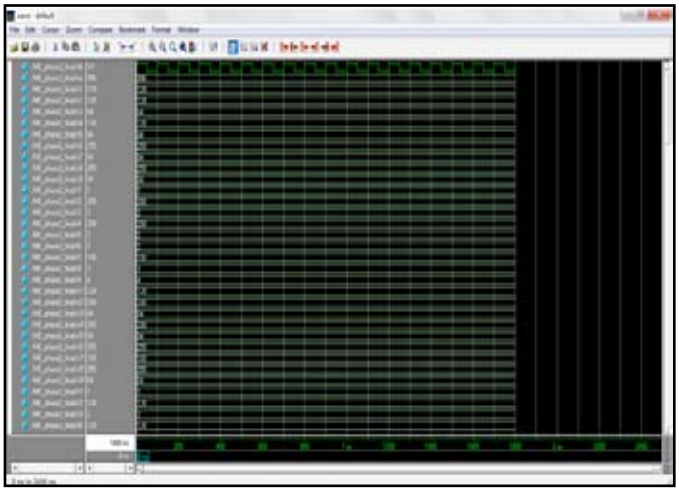


Fig. 4: Waveform Simulation

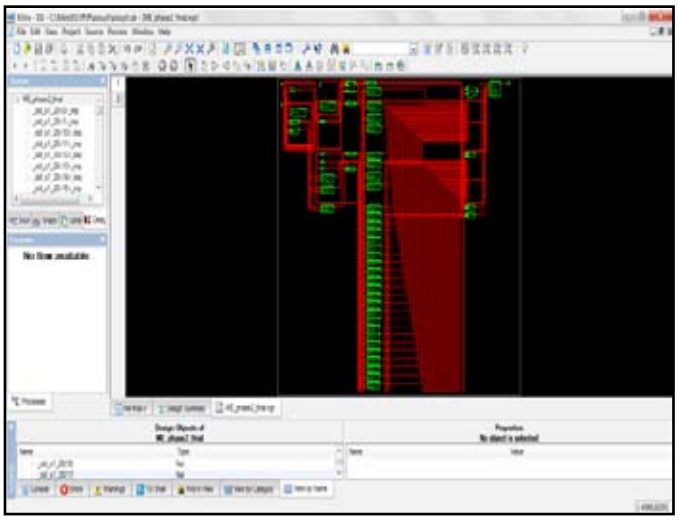


Fig. 5: RTL Schematic View of Proposed Architecture

The simulated waveform of the final search range prediction for the lower memory bandwidth requirement of the motion estimation is shown in fig. 4. RTL schematic view of the proposed architecture is shown in fig. 5. The total power consumption of the proposed architecture is shown in fig. 6.

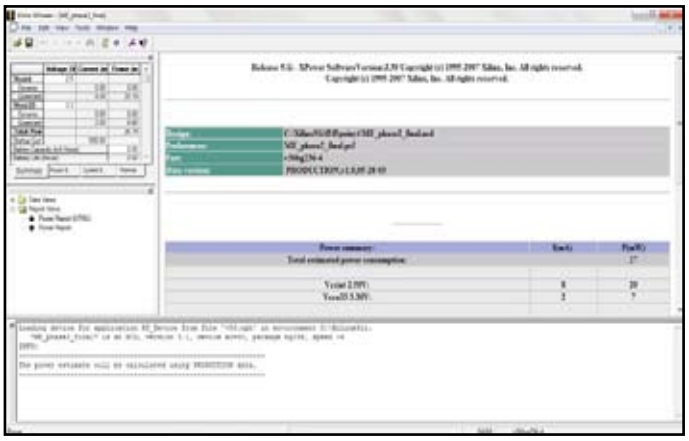


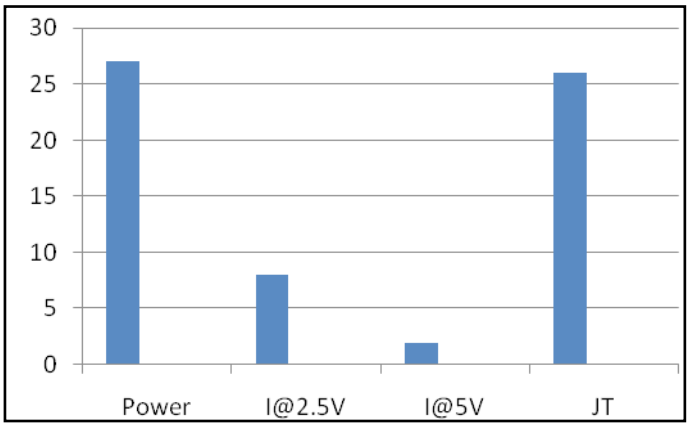
Fig. 6: Power Summary

The Table I and graph I shows the power consumption, quiescent current at 1.2v and quiescent current at 2.5v for the proposed architecture.

Table 1 : Performance Analysis

Evaluation Parameters	Results
Power consumption	27mW
Quiescent current at 1.2v	8mA
Quiescent current at 2.5v	2mA
Junction Temperature	26C

Graph 1 :Performance Analysis



IV. Conclusions

This project has presented efficient hardware architecture for motion estimation design to resource-limited mobile video applications. From the result, the final design provides superior power efficiency and lower bandwidth requirement. The proposed methods obtained the minimal area overhead and also acquired low hardware cost. The proposed ME scalable Motion Estimation with EDDR architecture is also implemented by using Verilog and synthesized by the ModelSim and Xilinx. The presented approach can be combined with the error detection and data recovery architecture to further reduce the error and to improve the power reduction compared with the bandwidth scalable ME.

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

V. Acknowledgement

References

[1] *S. Lee, "Fast motion estimation based on search range adjustment and matching point decimation," IET Image Process., vol. 4, no. 1, pp. 1–10, 2010.*

[2] *A. Bahari, T. Arslan, and A. T. Erdogan, "Low-power H.264 video compression architectures for mobile communication," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 9, pp. 1251–1261, Sep. 2009.*

[3] *Y. H. Chen, T. C. Chen, C. Y. Tsai, S. F. Tsai, and L. G. Chen, "Algorithm and architecture design of power-oriented H.264/AVC baseline profile encoder for portable devices," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 8, pp. 1118–1128, Aug. 2009.*

[4] *J.C. Tuan, T. S. Chang, and C. W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," IEEE Trans. Circuits Syst. Video Technol., vol. 12, no. 1, pp. 61–72, Jan. 2002.*

[5] *T. C. Chen, Y.H. Chen, S. F. Tsai, S. Y. Chien, and L. G. Chen, "Fast algorithm and architecture design of low-power integer motion estimation for H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., vol.17, no. 5, pp. 568–577, May 2007.*

[6] *H. Shim and C. M. Kyung, "Selective search area reuse algorithm for low external memory access motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 7, pp. 1044–1050, Jul. 2009.*

[7] *Y. K. Lin, C. C. Lin, T. Y. Kuo, and T. S. Chang, "A hardware-efficient H.264/AVC motion-estimation design for high-definition video," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 6, pp. 1526–1535, Jul. 2008.*

[8] *H. F. Ates and Y. Altunbasak, "Rate-distortion and complexity optimized motion estimation for H.264 video coding," IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 2, pp. 159–171, Feb. 2008.*

[9] *M.Miyama, J. Miyakoshi, Y. Kuroda, K. Imamura, H. Hashimoto, and M. Yoshimoto, "A sub-mWMPEG-4 motion estimation processor core for mobile video application," IEEE J. Solid-State Circuits, vol. 39, no. 9, pp. 1562–1570, Sep. 2004.*

[10] *Z. He, Y. Liang, L. Chen, I. Ahmad, and D.Wu, "Power-rate-distortion analysis for wireless video communication under energy constraints," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 5, pp. 645–658, May 2005.*

[11] *M. Yang, H. Cui, and K. Tang, "Efficient tree structured motion estimation using successive elimination," Proc. IEEE Vis., Image, Signal Process., vol. 151, no. 5, pp. 369–377, 2004.*

[12] *S. H.Wang, S. H. Tai, and T. H. Chiang, "A low-power and bandwidth efficient motion estimation IP core design using binary search," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 5, pp. 760–765, May 2009.*

[13] *C. Y. Kao and Y. L. Lin, "A memory-efficient and highly parallel architecture for variable block size integer motion estimation in H.264/AVC," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 6, pp. 866–874, Jun. 2010.*

Author's Profile

T.MARISH received her Bachelor of Engineering (B.E.,) degree in Electronics and Communication Engineering from the Sree Sowdambika College of Engineering, Virudhunagar, India, in 2011. Currently doing M.E - VLSI Design in Sethu Institute of Technology, Virudhunagar, India. Her research interest includes Low Power VLSI and Testing of VLSI circuits.

V.KARTHIK received his B.E., degree from National Engineering College in 2009 and completed his M.Tech from Sathyabama University in 2011. Presently working as an Assistant Professor in the Department of ECE at Sethu Institute of Technology, Tamilnadu, India. His current research areas includes: Low Power VLSI and Analog VLSI.