

FPGA DESIGN OF LOW POWER HIGH PERFORMANCE ARCHITECTURE FOR MOTION ESTIMATION

T. Suganya Thevi*¹, C. Chitra²

¹ Assistant Professor, Department of ECE, PSNA College of Engineering and Technology, Dindigul, India

² Professor, Department of ECE, PSNA College of Engineering and Technology, Dindigul, India

Abstract:-

The existing methodology provides low power consumption and will not tolerate errors in the computation of SAD (sum of absolute difference) process. The EDDR (Error detection and data recovery) architecture is proposed based on residue –and- quotient (RQ) code and integrated with motion estimation engine in order to reduce error rate and power consumption. The error in PEs which is the key component in motion estimation can be detected and recovered effectively by the proposed EDDR design. Motion estimation explores the temporal redundancy, which is inherent in video sequences, and it represents a basis for lossy video compression. Other than video compression, motion estimation can also be used as the basis for powerful video analysis and video processing.

compression is computation intensive and dissipates significant amount of power. The bandwidth scalable ME algorithm considers the video contents and bandwidth constraints to reduce memory access and to maximize coding performance. A greater bandwidth is allocated to high-motion MBs while bandwidth is significantly reduced for low-motion MBs.

A video consists of a time ordered sequence of frames, i.e., images. An obvious solution to video compression would be predictive coding based on previous frames. Compression proceeds by subtracting images: subtract in time order and code the residual error. It can be done even better by searching for just the right parts of the image to subtract from the previous frame. Consecutive frames in a video are similar temporal redundancy exists. Temporal redundancy is exploited so that not every frame of the video needs to be coded independently as a new image. The difference between the current frame and other frame(s) in the sequence will be coded, the small values and low entropy, good for compression. In Video Coding for Compression, the basic idea is to exploit the redundant data. Advances in semiconductors, digital signal processing, and communication technologies have made multimedia applications more flexible and reliable.

Introduction

1.1 Overview

Mobile devices performing video coding and streaming over wireless communication network are limited in the energy or power supply. Video compression allows raw video data to be compressed before it is sent through wireless channel. But the video

A good example is the H.264 video standard, which is widely regarded as the next generation video compression standard. Video compression is necessary in a wide range of applications to reduce the total data amount required for transmitting

or storing video data. Among the coding systems, a ME is of priority concern in exploiting the temporal redundancy between successive frames, yet also the most time consuming aspect of coding. ME is widely regarded as the most computationally intensive of a video coding system. The architecture design focuses more on hardware costs and throughput. H.264/AVC is the newest video coding standard demands higher encoding complexity adversely affects the speed and power. To analyze, control, and optimize the rate-distortion (R-D) behavior of the wireless video communication system under the energy constraint, power-rate-distortion (P-R-D) analysis framework is used. Replacement of SAD with a comparison reduces the computational complexity. It can achieve faster execution time and reduction in power consumption.

To achieve superior coding efficiency Rate-distortion (R-D) optimal selection of both the motion vectors (MVs) and the coding mode of each MB is essential for H.264 encoder. It uses pixel truncation to reduce the computation and memory access for variable block size motion estimation. A fast full-search block matching algorithm to save power. A power-aware design has multiple power modes of operation and can adapt its operating configurations based on environmental conditions, such as energy constraint (battery status) or user preference.

- A motion estimation circuit which consumed 81.37mW of power has been used.
- The existing ME circuit reduces power consumption, but not able to correct the errors in motion estimation process.

An efficient hardware architecture for motion estimation design to resource-limited mobile video applications using an integrated bandwidth rate distortion optimization framework. This framework predicts and allocates the appropriate data bandwidth for motion estimation under a limited bandwidth supply to fit a dynamically changing bandwidth supply. We propose EDDR Architecture (Error detection and data recovery) in place of existing Motion estimation engine to reduce the power consumption. In existing design, ME engine performs motion estimation only, not able to detect and correct any type of errors in motion estimation process. Hence, we replace ME engine by EDDR architecture, which able to generate motion vector with error correcting capabilities. Relative to previous low-complexity VLSI designs, this work achieves hardware utilization reduction in gate counts and requires only one-line memory buffer.

Hence, EDDR Architecture (Error detection and data recovery) is integrated with the existing Motion estimation engine to further reduce the power consumption. In existing design, ME engine performs motion estimation only, not able to detect and correct any type of errors in motion estimation process. Finally, ME engine with EDDR architecture, is able to generate

motion vector with error correcting capabilities.

1.1 EDDR Architecture Design

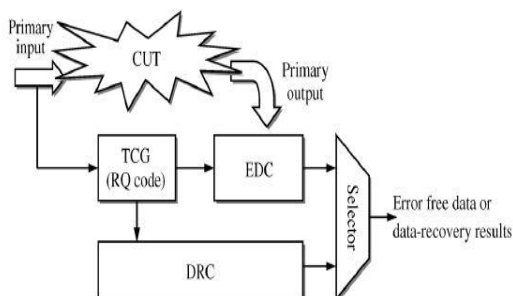
EDD scheme, which comprises of two major circuit designs, i.e. error detection circuit (EDC) and data recovery circuit (DRC), to detect errors and recover the corresponding data. It also consists of PE(Processing Elements) and test code generator(TCG). The test codes from TCG and the primary output from CUT are delivered to EDC to determine whether the CUT has errors. DRC is in charge of recovering data from TCG.

	0	1	2	3
0	128	128	64	255
1	128	64	255	64
2	64	255	64	128
3	255	64	128	128

Fig:1(a) Current pixel block

	0	1	2	3
0	1	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	5

Fig:1(b) Reference pixel block



Test Code Generator (TCG)

TCG is an important component of the proposed EDDR architecture. TCG is used to generate corresponding test codes in order to detect errors and recover data. Each TCG Consists of RQ (residue-and-quotient)Code generator(RQCG) to generate residue-and-quotient.

Processing Elements

The Processing Element generate the SAD(Sum of Absolute Difference) value. It estimates the absolute difference between the Cur_pixel of the search area and the Ref_pixel of the current macro block.

$$SAD = \sum_{i=0}^3 \sum_{j=0}^3 |X_{ij} - Y_{ij}|$$

3.1 BANDWIDTH SCALABLE MOTION ESTIMATION ARCHITECTURE

The ME architecture design combines the bandwidth-scalable algorithm with a full search ME engine. The idea of proposed algorithm is to assign the available bandwidth in an R-D-optimized sense within the given bandwidth budget. The whole algorithm is as follows.

The memory bandwidth budget, within Pupdate is initialized for bandwidth allocation in later coding processes and is calculated by

$$BWbudget = (Br/Fr) *Gp$$

where *Br* denotes the data rate (bytes/s) allocated by the memory controller, *Fr* denotes the coded frame

number per second, and Gp denotes the frame numbers within Pupdate. Suppose the bandwidth budget is evenly distributed over all coded MBs within Pupdate

$$BW_{budget} = (2 * SR_{sys} + 16) NMB$$

where NMB is the total number of MBs within Pupdate. Thus, the global system SR (SR_{sys}) for the ME hardware is

$$SR_{sys} = \text{Floor}\{1/2((BW_{budget}/NMB)^{0.5}-16)\}$$

as the upper bound of the next SR adjustment. If P update is changed when available bandwidth is changed suddenly, new value can be set as well to force lower bandwidth usage when necessary.

Justify the coding efficiency under a given bandwidth, the actual bandwidth efficiency for coding the k th MBs is expressed as follows. The bandwidth efficiency for a current coded k th MB is calculated which describes how much R-D gain can be achieved for one unit

$$G[K] = (JMVP[i]-JBMA[i])/BW_{used}$$

Using past statistics, we can calculate the possible bandwidth usage (called past BW prediction BWPP) for the current coded MB. However, the remaining available bandwidth should be equally distributed for the remaining MBs (called future BW prediction BWFP). Thus,

$$BWFP[k] = (BW_{budget}-BW_{used}[k])/(NMB-(k-1))$$

With the above available bandwidth bounds and R-D data, a SR decision for the MB coding is

formulated. Finally, the SR is determined as

$$Final_SR = \min(SR_{sys}, \max(Pred_SR, 1/4 * sum_mv))$$

The SR is determined according to three bandwidth modes and bandwidth condition. The mode BW L denotes the low available bandwidth case as the average bandwidth usage of previous MBs falls outside the available bandwidth interval. The SR should be reduced to avoid an overflow of bandwidth usage. The mode BWN denotes the low available bandwidth case as the average bandwidth usage of previous MBs falls within the available bandwidth interval. The R-D performance can be optimized by increasing the SR.

3.2. BLOCK DIAGRAM OF BANDWIDTH SCALABLE ME

The motion estimation engine compares the current pixel buffer and reference pixel buffer to compute the SAD (Sum of Absolute Differences). It comprises of ME engine and bandwidth scalable ME controller. The SR of the current MB is determined by the bandwidth scalable ME controller. First, the SR of the current MB is decided by the bandwidth scalable ME controller. The data request of Search Range data and current MB are given by the pre retrieval control unit and the data are loaded from external memory by the memory controller and stored in the reference pixel buffer and current pixel buffer. The data are used to calculate SADs through the two SAD generation modules. The final best mode is selected by the mode decision module and output is sent along with its motion vector difference. The R-D cost data like

JBMA and JMVP , the best MB mode, and the sum_mv are sent back to the bandwidth scalable ME controller for SR decision of the next MB. The control unit produces control signals to the two SAD generation module, the motion vector predictor generator, and the pre-retrieval control unit.

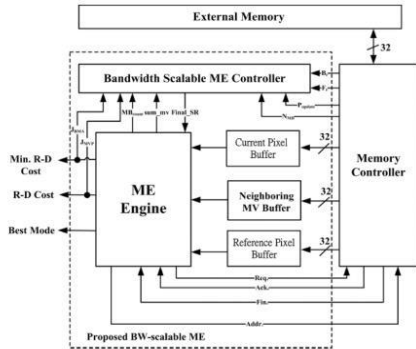


Fig: 4 Block Diagram of Bandwidth Scalable ME

3.3 MOTION ESTIMATION ENGINE

In the ME engine the control unit produces control signals to the two SAD generation module, the motion vector predictor generator, and the pre-retrieval control unit. It needs to count the number of MBs up to the current coded kth MB. The handshaking policy between the pre-retrieval control unit and the memory controller is in the order of request-acknowledge- address-finish. After the task is finished, the memory controller sends a finish signal and waits for the next request.

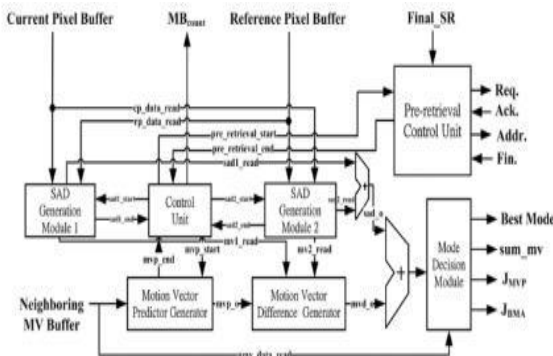


Fig: 5 Block Diagram of Motion Estimation Engine

The variable-block-size motion estimation (VBSME) is implemented by the SAD tree with four pipeline stages. The VBSME calculates all ME modes of a search position in a search area within a clock cycle.

3.4 BANDWIDTH SCALABLE MECONTROLLER

The bandwidth-scalable ME controller is used to determine the SR as defined and generate the memory address to the pre-retrieval control unit to load the required reference pixels for the next ME search. The system architecture has six major tasks Initialization, Bandwidth efficiency calculation, Bandwidth prediction, Bandwidth allocation, Search range prediction, Final Search range prediction.

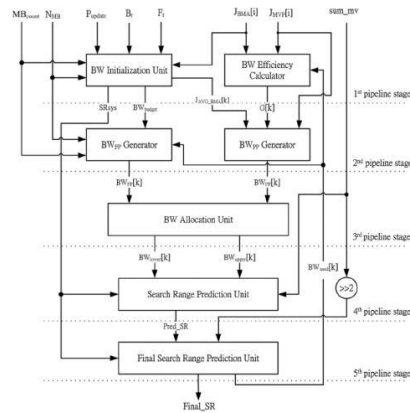


Fig: 6 Block Diagram of Bandwidth Scalable ME Controller

3.5 ARCHITECTURE OF THE BANDWIDTH ALLOCATION UNIT

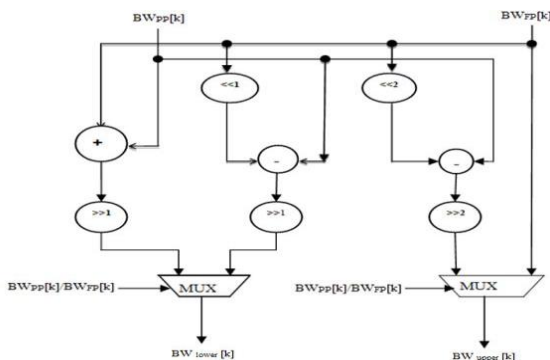


Fig: 7 Architecture of Bandwidth Allocation Unit

The bandwidth allocation unit allocates the bandwidth for the current frame and reference frame.

3.6 ARCHITECTURE OF THE SEARCH RANGE PREDICTION UNIT

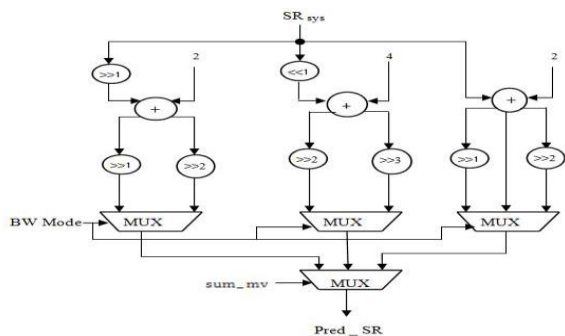


Fig: 8 Architecture of the Search Range Prediction Unit

The search range is determined according to the bandwidth mode. Then

compare the three MVs and the summation of the absolute values of these three MVs (sum_mv). The predictive SR ($Pred_SR$) depends on the sum_mv and the bandwidth interval consideration. There are three bandwidth modes BW_H (Bandwidth high mode), BW_N (Bandwidth normal mode), BW_L (Bandwidth low mode). Finally, the SR is determined.

The SR_{sys} indicates the system SR, the factors SR_{shift_low} , SR_{shift_middle} and SR_{shift_high} are set as spatial resolution-independent parameters. The system SR is used as the upper bound of the SR to prevent bandwidth usage overflow.

3.7 ARCHITECTURE OF BANDWIDTH EFFICIENCY CALCULATOR

To compute the bandwidth efficiency from the bandwidth budget and bandwidth used. The bandwidth allocation unit, search range prediction unit and bandwidth efficiently implemented with only shifters, subtractors, adders, comparators and one divider for low hardware cost.

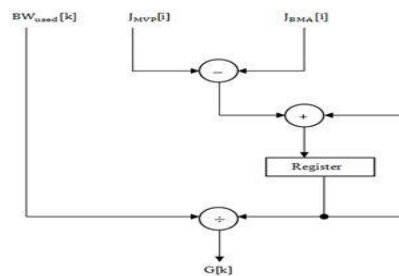


Fig: 9 Architecture of Bandwidth Efficiency Calculator

3.8 PROPOSED ARCHITECTURE

The bandwidth scalable motion estimation architecture provides low power consumption, but it may not tolerate errors in the computation of SAD. To overcome such limitation EDDR architecture is integrated with ME. Fig. 7 shows the theoretical view of the proposed EDDR scheme, which consists of 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 CUT. The test code generator (TCG) uses the idea of RQ code to generate the equivalent test codes for error detection and data recovery. The test codes from TCG and the primary output from CUT are given to EDC to determine whether the CUT has errors. DRC is in charge of re-covering data from TCG.

Additionally, a selector is enabled to export error-free data or data-recovery results. An array-based computing structure, such as ME, discrete cosine transform (DCT), iterative logic array (ILA), and finite impulse filter (FIR), is possible for the proposed EDDR scheme to detect errors and recover the corresponding data. The systolic ME is used as CUT to demonstrate the possibility of the EDDR architecture. A ME consists of many PEs added in a 1-D or 2-D array for video encoding applications. A PE generally consists of two ADDs (i.e. an 8-b ADD and a 12-b ADD) and an accumulator (ACC). Then, the 8-b ADD (a pixel has 8-b data) is used to estimate the addition of the current pixel (Cur_pixel) and reference pixel (Ref_pixel).

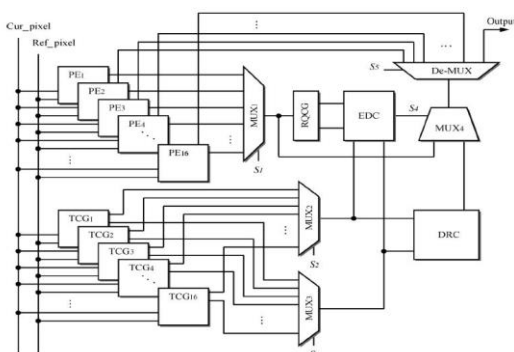


Fig: 10 Proposed ME Architecture

3.9 Residue-and-Quotient Generation

$$\begin{aligned}
 R_T &= \left| \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - Y_{ij}) \right|_m \\
 &= |(X_{00} - Y_{00})|_m + |(X_{01} - Y_{01})|_m + \dots \\
 &\quad + |(X_{(N-1)(N-1)} - Y_{(N-1)(N-1)})|_m \\
 &= |(q_{x00} \cdot m + r_{x00}) - (q_{y00} \cdot m + r_{y00})|_m \\
 &\quad + \dots + |(q_{x(N-1)(N-1)} \cdot m + r_{x(N-1)(N-1)}) \\
 &\quad \quad - (q_{y(N-1)(N-1)} \cdot m + r_{y(N-1)(N-1)})|_m \\
 Q_T &= \left\lfloor \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (X_{ij} - Y_{ij})}{m} \right\rfloor \\
 &= \left\lfloor \frac{(X_{00} - Y_{00}) + (X_{01} - Y_{01}) + \dots + (X_{(N-1)(N-1)} - Y_{(N-1)(N-1)})}{m} \right\rfloor
 \end{aligned}$$

FAULT MODEL - The PEs are essential building blocks and are connected regularly to construct a ME. Generally, PEs are surrounded by sets of ADDs and accumulators that decide how data flows through them. PEs can be considered the class of circuits called ILAs, whose testing assignment can be easily achieved by using the fault model, cell fault model (CFM).

TCG - TCG is an important component of the EDDR architecture. The TCG design is based on the ability of the RQCG circuit to generate corresponding test codes in order to detect errors and recover data. TCG is the combination of PE and RQCG. It will produce RT and QT value and given to EDC circuit to detect errors.

EDDR CIRCUIT - The outputs between TCG and RQCG is compared in EDC (Error Detection Circuit) in order to find whether errors have occurred.

5. Performance Comparison

Methodology	Power Consumption
Existing	74mW
Proposed (initial)	52mW
Proposed (final)	34mW

CONCLUSION

The existing motion estimation and bandwidth controller achieved 81.37mw of power consumption which yields to the bandwidth allocation of each frame in the video. It may not support or provide any error detection capability during the motion estimation process. Such a drawback can be reduced using EDDR architecture combined with bandwidth controller which significantly reduces the hardware utilities. Even though, the existing methodology provides low power consumption, it may not tolerate errors in the computation of SAD (Sum of Absolute Difference) process. To overcome such a drawback EDDR architecture is proposed and integrated with motion estimation engine in order to reduce error rate and power consumption.

REFERENCES

- [1] T. Wiegand, G. J. Sullivan, G. Bjontegaad, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–575, Jul. 2003.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] M. G. Koziri, A. N. Dadaliaris, G. I. Stamoulis, and I. X. Katsavounidis, "A novel low-power motion estimation design for H.264," in *Proc. IEEE Int. Conf. Appl. Specific Syst. Arch. Processors*, 2007, pp. 247-252.
- [7] M. Miyama, J. Miyakoshi, Y. Kuroda, K. Imamura, H. Hashimoto, and M. Yoshimoto, "A sub-mW MPEG-4 motion estimation processor core for mobile video application," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1562–1570, Sep. 2004.
- [8] P. Li and H. Tang, "A low-power VLSI implementation for variable block size motion estimation in H.264/AVC," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2010, pp. 2972–2975.
- [9] T. C. Chen, Y. H. Chen, C. Y. Tsai, S. F. Tsai, S. Y. Chien, and L. G. Chen, "2.8 to 67.2 mw low-power and power-aware H.264 encoder for mobile applications," in *Proc. IEEE Symp. VLSI Circuits*, 2007, pp. 222-223.