ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 4, April 2017

Realization of Reconfigurable Approximate Arithmetic Units for Motion Estimation with Error Detection and Data Recovery Architecture

B.Kiruthiga[1] and M.Rekha[2]
ME VLSI DESIGN / Srinivasan Engineering College/Perambalur.
Assistant Professor /Srinivasan Engineering College/Perambalur.

***Abstract*—The field of approximate computing has established significant attention in image and video compressions algorithm such as MPEG and JPEG. However, existing approximate architectures typically static approximate hardware configuration is used for an MPEG encoder (i.e., a fixed level of approximation), the output quality varies greatly for different input videos.This project addresses this issue by suggesting a reconfigurable approximate arithmetic units for MPEG encoders that improves power consumption. To design the dual mode full adder(DMFA) and then implement it in reconfigurable adder/subtractor blocks (RABs), which have the capability to moderate their degree of approximation and consequently incorporate these blocks in the motion estimation and discrete cosine transform components of the MPEG encoder. In order to test Motion Estimation in a video coding system with Error Detection and data recovery Architecture(EDDR) is designed based on the Residue-and-Quotient (RQ)code. Residue-and-quotient code is used in the error detection architecture that has 16 Processing Elements (PEs) and 16 Test Code Generation (TCG) blocks for computing the test codes for each pixel value in the macroblock this architecture is used which has a single PE and a TCG for computing the test codes for the pixel valuesAn error in processing elements (PEs), key components of a ME, can be detected and recovered effectively by using the proposed EDDR design.It can be easily sustained to other DSP applications and it attain the power saving upto 38% over a conventional nonapproximated MPEG encoder. The motion estimation with EDDR are design by Verilog HDL and simulated by Modelsim 6.4 c. The power and Delay will be synthesized by Xilinx tool**

***Index Terms*— Approximate circuits, approximate computing, error detection and correction ,low power design.**

I.INTRODUCTION

Introducing a limited amount of computing imprecision in image and video processing algorithms often results in a negligible amount of perceptible visual change in the output, which makes these algorithms as ideal candidates for the use of approximate computing architectures. Approximate computing architectures exploit the fact that a small relaxation in output correctness can result in significantly simpler and lower power implementations. However, most approximate hardware architectures proposed so far suffer from the limitation that, for widely varying input parameters, it becomes very hard to provide a quality bound on the output, and in some cases, the output quality may be severely degraded. The main reason for this output quality fluctuation is that the degree of approximation (DA) in the hardware architecture is fixed statically and cannot be customized for different inputs.

This project adopts a different approach to addressing this problem by dynamically reconfiguring the approximate hardware architecture depending on the inputs.We demonstrate that, for a fixed level of hardware approximation in an MPEG encoder, the output quality varies widely across different videos, often going below acceptable limits.This shows that setting the level of hardware approximation statically is insufficient.

Video encoding, also known as video transcoding.It is basically a process of converting a given video input into a digital format that is compatible with most types of Web players and mobile devices.This project addresses this issue by proposing a reconfigurable approximate architecture for MPEG encoders that optimizes power consumption. The proposed adder design are dual mode full adder(DMFA) to implement the reconfigurable adder/subtractor blocks (RABs), which have the ability to modulate their degree of approximation, and subsequently integrate these blocks in the motion estimation and discrete cosine transform modules of the MPEG encoder.It can be easily extended to other DSP applications and this structure are assessed by comparing their power ,delay and area those of other adder design by Verilog HDL and simulated by Modelsim 6.4c and synthesized by Xilinx tool.

We investigate, for the first time, the use of dynamically reconfiguration approximate hardware archietectures that vary the DA during run-time across multiple computational cycles, depending on the inputs. Toward this end, we propose the design of reconfigurable adder/subtractor blocks (RABs) for four commonly used adder

architectures, viz., ripple carry adder (RCA), carry lookahead adder (CLA), carry bypass adder (CBA), and carry select adder (CSA), and subsequently integrate them into the MPEG encoder to enable quality configurable execution.We propose a design methodology to adapt the DA dynamically based on the video characteristics with the goal of ensuring that output quality is within a specified bound.

This work develops a novel EDDR architecture based on the RQ code to detect errors and recovery data in PEs of a ME Regular arrangement of PEs with size 4x4 constitutes a ME. Advancements in VLSI technologies facilitate the integration of large number of PEs into a single chip. Large number of PEs arranged as an array helps in accelerating the computation speed.

The remainder of this paper is organized as follows. Section II gives an account of related work in the domain of approximate computing. Section III gives an account of related work of existing system. Section IV serves as the motivation for our work and the proposed reconfigurable approximate architecture for Motion estimation with error detection and correction architecture. Section V reports the results obtained through Synthesis and Simulation implementation for our design.Section VI concludes this paper.

II RELATED WORK

There has been a lot of effort in constructing energy-efficient video compression schemes. Many of them are related to the specific case of an MPEG encoder. Different methods of power-reduction include algorithmic modifications [1], [2], voltage over-scaling [3], and imprecise computation of metrics [4]. The introduction of approximate computing methods has opened up entirely new opportunities in building low-power video compression designs. Approximate computing methods attain a large amount of power savings by introducing a small amount of error or inaccuracy into the logic block. Different methodologies for approximation include error introduction through voltage overscaling [5], [6], intelligent logic manipulation [7], and circuit simplification using don't care-based optimization methods [8]. The methods in [9] and [10]introduce imprecision by replacing adders with their approximate counterparts.

The approximate adders are attained by logically removing some of the transistors in a mirror adder. An important point to note is that these approximate circuits are hardwired and cannot be modified without resynthesizing the entire circuit..(He and Liou [12] and He *et al.* [13] use bit truncation to introduce approximations in the ME block of an MPEG encoder.

However, such a coarse-grained input truncation is applicable only to the specific case of ME and gives unsatisfactory results for other blocks, such as discrete cosine transform (DCT), which requires a finer regulation over error.As in [9] and [10], this paper also aims in approximating the adders of the ME and DCT blocks of an MPEG encoder. However, this paper introduces the concept of dynamically reconfigurable approximation, which, as we will display, benefits in retaining better control over application-level quality metrics while simultaneously reaping the power consumption.

III EXISTING APPROXIMATION ADDERS

In existing design, Logic complexity reduction as an alternative approach to take advantage of the relaxation of numerical accuracy. We demonstrate this concept by suggesting various imprecise or approximate Full Adder (FA) cells with reduced complexity at the transistor level, and utilize them to design approximate multi-bit adders.

In addition to the inherent reduction in switched capacitance, our methods result in significantly shorter critical paths.However, its introduces the concept of dynamically reconfigurable approximation, application-levelquality metrics while simultaneously reaping the power consumption benefits of hardware approximation.

In the system introduce imprecision or approximate FA by replacing conventional FA with their approximate counterparts. The approximate adders are obtained by intelligently deleting some of the transistors in a full adder, an important point to note is that these approximate circuits are hardwired and cannot be modified without re synthesizing the entire circuit.It reduced complexity at the transistor level, and utilize them to design approximate multi-bit adders

3.1.Approximation 1:

To get an approximation adder with less transistor, low power application we started to reduce transistors from the digital circuit one by one. The approximation 1 is having 16 transistors and it introduces one error in Cout and two errors in sum.

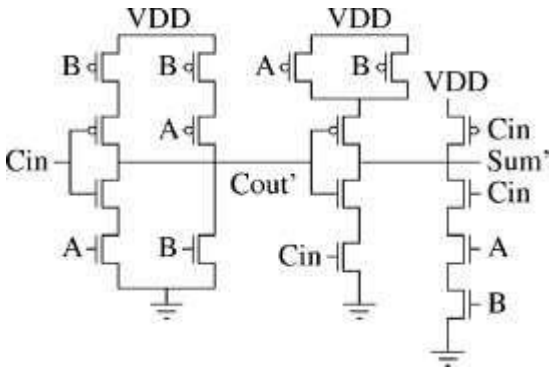

Fig-1: Approximation 1

3.2 Approximation 2:

In approximation 2 the Cout=A and sum=B for six out of eight cases. The approximation 4 have 11 transistors and it have two errors in Cout and three error in sum..

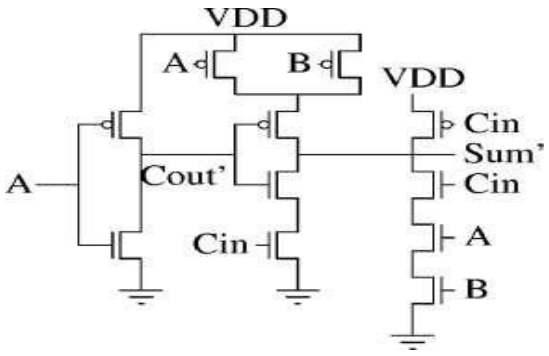

Fig-2: Approximation 2

IV. PROPOSED ARCHITECTURE

The proposed scheme replaces each FA cell of the adders/subtractors with a dual-mode FA (DMFA) cell in which each FA cell can operate either in fully accurate or in some approximation mode depending on the state of the control signal APP. A logic high value of the APP signal denotes that the DMFA is operating in the approximate mode A logic low value of the APP signal denotes that the DMFA is operating in the accurate mode. We term these adders/subtractors as RABs.

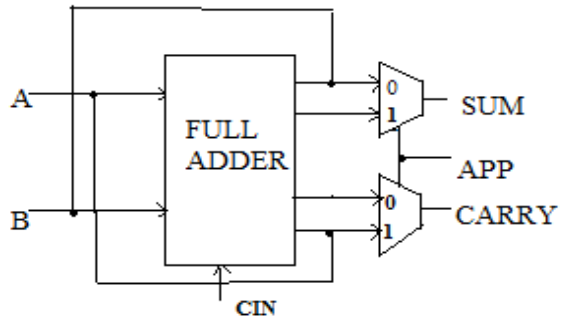

Fig. 3. 1-bit DMFA.

Table I Power Consumption of Different DMFA Modes

Original FA (μW)	DMFA Accurate Mode (μW)	DMFA Approximate Mode (μW)
1.53	1.74	0.01

4.1 8-Bit Reconfigurable Ripple Carry Adder Block:

The logic block of the DMFA cell, which replaces the constituent FA cells of an 8-bit RCA. In addition, it also consists of the approximation controller for generating the appropriate select signals for the multiplexers. A multimode FA cell would provide even a better alternative to the DMFA from the point of controlling the approximation magnitude However, it also increases the complexityof the decoder block used for asserting the right select signals to the multiplexers as well as the logic overhead for the multiplexers themselves..

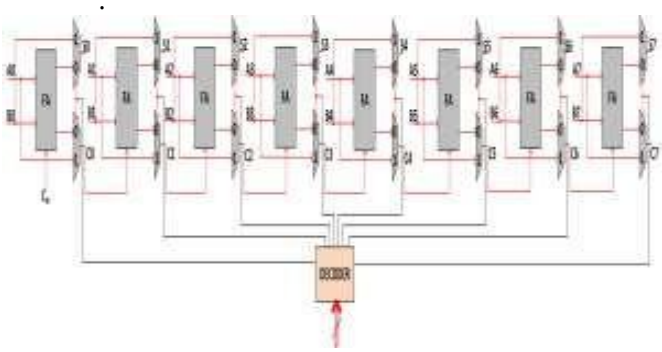

Fig-4: 8-bit reconfigurable RCA block.

This undermines the primary objective as most of the power savings that we get from approximating the bits are lost. Instead, the two-mode decoder and the 2:1 multiplexers have negligible overhead and also provide sufficient command over the approximation degree.

4.2 8-Bit Reconfigurable Carry Lookahead Adder Block:

We implemented a 8-bit CLA consisting of four different types of basic blocks depending upon the presence of sum (S), Cout, carry propagation (P), and carry generation (G) at different levels.We address the basic blocks present at the first (or lowermost) level of a CLA, which have inputs coming in directly, as carry lookahead blocks, CLB1 and CLB2. The difference among them being that CLB1 produces an additional Cout signal compared with CLB2.

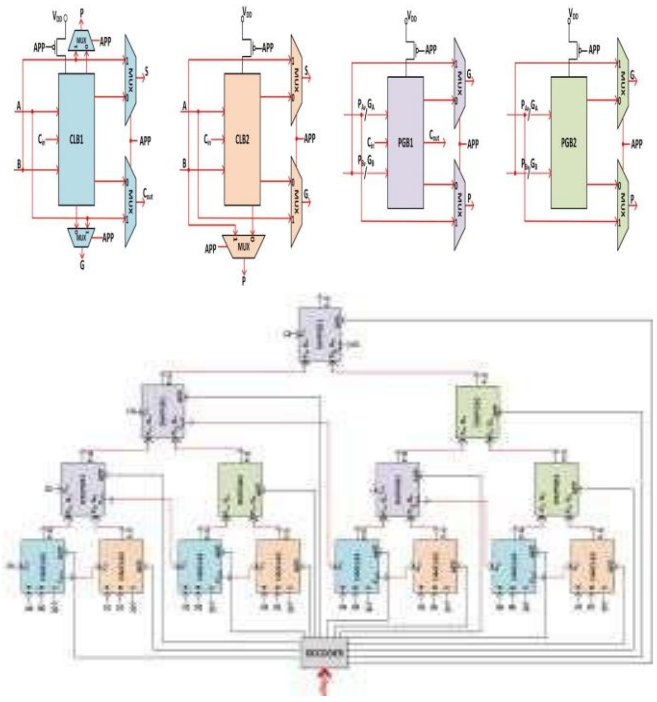

Fig-5. 8-bit reconfigurable CLA block

4.3 Motion Estimation

Motion estimation is the process of determining that motion vectors describe the transformation from one 2D image to another; usually from adjacent frames in a video sequence. It is an ill-posed problem as the motion is in three dimensions but the images are a projection of the 3D scene onto a 2D plane. The motion vectors may relate to the whole image (global motion estimation) or specific parts, such as rectangular blocks, arbitrary shaped patches or even per pixel.

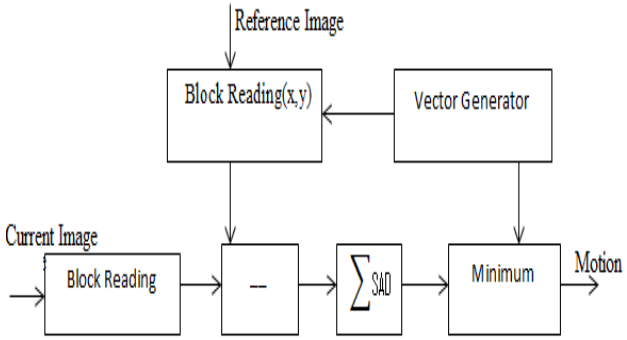

Fig.6. Block Diagram of Motion Estimation

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

Cur_pixel

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

Ref_pixel

Thus,by utilizing PEs, SAD shown in as follows, in a macroblock with size of N*N can be evaluated:

$$SAD=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|Xij-Yij|$$

Where Xij and Yij represent current pixel and reference pixel.

The SAD uses a three-step method. The initial stage of the AD is to calculate the difference between the current pixel data and reference pixel data. These outcomes are then given as the input to second step where they are accumulated and saved in the registers. The purpose of register and Accumulators is to confirm that as soon as the computations has been completed these are saved and served back to calculate the AD for all the sub-blocks.Then all the outputs of the accumulated values are compared using comparators to find the minimum SAD.

4.4 RQ Code Generation

Coding approaches such as parity code, Berger code, and residue code have been considered for design applications to detect circuit errors. Residue code is generally separable arithmetic codes by estimating a residue for data and appending it to data. Error detection logic for operations is typically derived by a separate residue code, making the detection logic is simple and easily implemented. For instance, assume that N denotes an integer, N_1 and N_2 represent data words, and m refers to the modulus. A separate residue code of interest is one in which N is coded as a pair $(N|N|_m)$. Notably, $|N|_m$ is the residue of N modulo However, only a bit error can be identified based on the residue code. Additionally, an error cannot be recovered effectively by using the residue codes. There-fore, this work presents a quotient code, which is derived from the residue code, to assist the residue code in detecting multiple errors and recovering errors.

The mathematical model of RQ code is simply described as follows. Assume that binary data X is expressed as

$$X=\sum_{j=0}^{n-1} bj2^{j}$$

The RQ code of X modulo m expressed as

$$R=|X|$$
$$Q=|X/m|$$

If the modulus $m=2^{k}-1$,the value of k= [n/2]

4.5 Test Code Generator Design

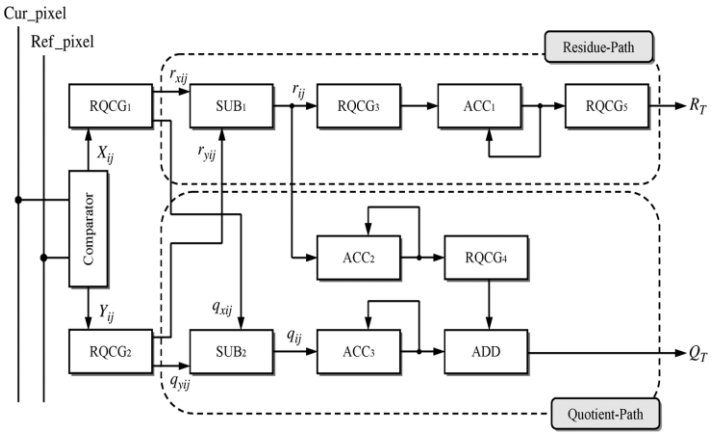

Fig.7. Block Diagram of Test Code Generator

TCG is an important module of the motion estimation with EDDR architecture. Notably, TCG design is based on the ability of the RQCG circuit to produce corresponding test codes in order to identify errors and recover data. The SAD evaluate the absolute difference between the Cur_pixel of the search area and the Ref_pixel of the current macroblock.

Notably, if $X_{ij} \geq Y_{ij}$, then X_{ij} and Y_{ij} are the luminance pixel value of Cur_pixel and Ref_pixel, respectively. Conversely, X_{ij} represents the luminance pixel value of Ref_pixel, and Y_{ij} denotes the luminance pixel value of Cur_pixel when $X_{ij}<Y_{ij}$. At the 2nd clock, the values of X_{00} and Y_{00} are produced and the corresponding RQ code r_{x00}, q_{x00}, r_{00}, q_{00} can be captured by the RQCG and RQCG circuits if the 3rd clock is triggered. The 4th clock displays the operating results. The modulus value of r_{00} is then obtained at

the 5th clock. Next, the summation of quotient values and residue values of modulo m are proceeded with from clocks 5–21 through the circuits of ACCs.

4.6 EDDR Architecture Design

Motion estimation with Error Detection and Data Recovery scheme, which 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 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.

A ME consists of many PEs incorporated 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). Next, the 8-b ADD (a pixel has 8-b data) is used to estimate the addition of the current pixel (Cur_pixel) and referencepixel (Ref_pixel). Additionally, a 12-b ADD and an ACC are required to accumulate the results from the 8-b ADD in order to determine the sum of absolute difference (SAD) value for video encoding applications.

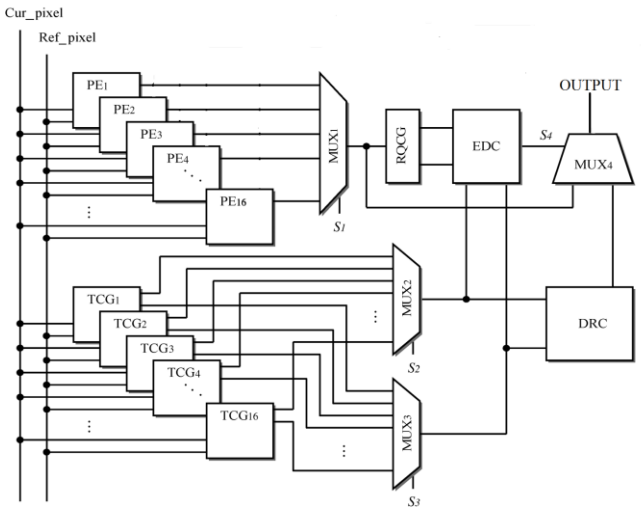

Fig.8. Block Diagram of Motion Estimation with EDDR

4.7 Error Detection circuit

Error detection circuit is used to perform the operation of error detection in the specific PEi,this block is used to match the output from the TCG(RT&QT)with output from RQCG1(RPEi&QPEi).In order to identify the occurrence of an error,if the value RPEi ≠ RT & QPEi ≠ QT.Then the error in the PE can be detected.The EDC output is generated 0/1 signal to indicate that the tested PE is error free/error. Based on the definition of the fault model, the SAD value is influenced if either SA1 and/or SA0 error have occurred in a specific Pei. The SAD value is transformed to SAD=SAD+e.If an error e occurred .The error signal e is expressed as e=qe.m+re

4.8 Data Recovery circuit

The original data is recovered in the Data Recovery Circuit (DRC) during error correction process, by separating the RQ code from the TCG. The data recovery is possible by implementing the mathematical model as,

$$SAD=m*QT+RT$$
$$=(2^{j}-1)*^{QT+RT}$$
$$=2^{j}*QT-QT+RT$$

The proposed motion estimation with EDDR design executes the error detection and data recovery operations simultaneously. Additionally, error-free data from the tested PE_i or the data recovery that results from DRC is selected by a multiplexer (MUX) to pass to the next specific PE_{i+1} for subsequent testing.

V. RESULT AND OUTPUT

Reconfigurable approximate arithmetic units for motion estimation with error detection and data recovery architecture is designed based on the Residue-and-Quotient (RQ)code . Residue-and-quotient code is used in the error detection architecture that has 16 Processing Elements (PEs) and 16 Test Code Generation (TCG) blocks for computing the test codes for each pixel value in the macroblock this architecture is used which has a single PE and a TCG for computing the test codes for the pixel values.By this the overall power and delay can be calculated.

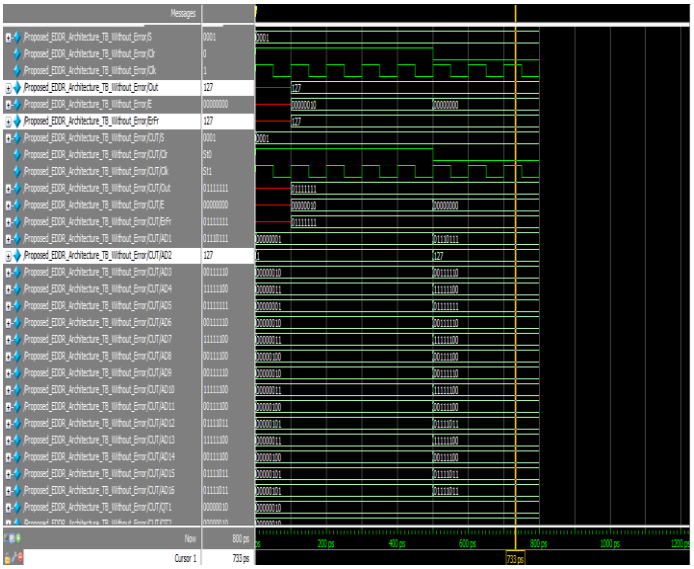

Fig 9.Output Waveform of Motion Estimation With EDDR

METHOD	POWER(w)
Motion Estimation with EDDR in Normal Full Adder	9.504
Motion Estimation with EDDR in DMFA	5.589

Table II Comparison of Power in Existing and Proposed Architecture

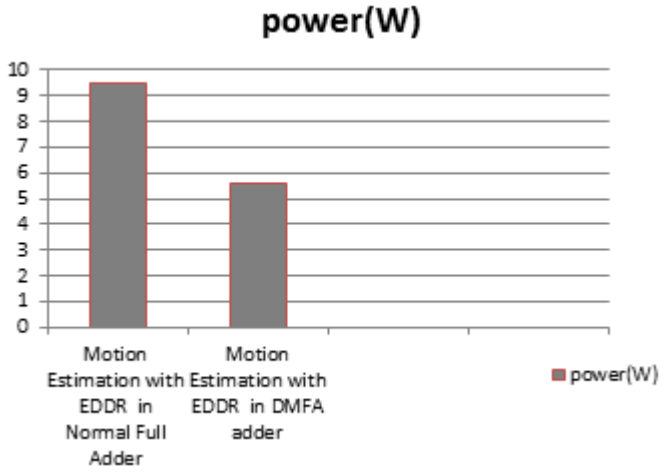

Fig.10 power consumption of Existing and Proposed Architecture

VI CONCLUSION

Reconfigurable approximate arithmetic units for motion estimation with error detection and data recovery architecture is designed based on the Residue-and-Quotient (RQ)code. This architecture is used which has a single PE and a TCG for computing the test codes for the pixel valuesThe proposed architecture is based on the concept of dynamically reconfiguring the level of approximation, while maintaining output quality across different input videos , it optimize power consumption and less delay It includes the incorporation of other approximation techniques and extending the approximations to other arithmetic and functional blocks. We will implement 16 Bit and 32 Bit adders.To Implement a full adder circuit using gate diffusion input technology

REFERENCES.

[1] M. Elgamel, A. M. Shams, and M. A. Bayoumi, "A comparative analysis for low power motion estimation VLSI architectures," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2000, pp. 149–158.

[2] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, no. 6,pp. 858-876,Jun.1995

[3] I. S. Chong and A. Ortega, "Dynamic voltage scaling algorithms for power constrained motion estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 2. Apr. 2007, pp. II-101–II-104.

[4] I. S. Chong and A. Ortega, "Power efficient motion estimation using multiple imprecise metric computations," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2007, pp. 2046–2049

[5] D. Mohapatra, G. Karakonstantis, and K. Roy, "Significance driven computation: A voltage-scalable, variation-aware, quality-tuning motion estimator," in *Proc. 14th ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, 2009, pp. 195–200.

[6] M. Elgamel, A. M. Shams, and M. A. Bayoumi, "A comparative analysis for low power motion estimation VLSI architectures," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2000, pp. 149–158.

[7] M. Elgamel, A. M. Shams, and M. A. Bayoumi, "A comparative analysis for low power motion estimation VLSI architectures," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2000, pp. 149–158.

[8] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, no. 6,pp. 858-876,Jun.1995

[9] I. S. Chong and A. Ortega, "Dynamic voltage scaling algorithms for power constrained motion estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 2. Apr. 2007, pp. II-101–II-104.

[10] M. Elgamel, A. M. Shams, and M. A. Bayoumi, "A comparative analysis for low power motion estimation VLSI architectures," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2000, pp. 149–158.

[11]F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, no. 6,pp. 858-876,Jun.1995

[12]M. Elgamel, A. M. Shams, and M. A. Bayoumi, "A comparative analysis for low power motion estimation VLSI architectures," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2000, pp. 149–158.

[13]M. Elgamel, A. M. Shams, and M. A. Bayoumi, "A comparative analysis for low power motion estimation VLSI architectures," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2000, pp. 149–158.

[14]F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV: A review and a new contribution," *Proc. IEEE*, vol. 83, no. 6,pp. 858-876,Jun.1995

[15]I. S. Chong and A. Ortega, "Dynamic voltage scaling algorithms for power constrained motion estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 2. Apr. 2007, pp. II-101–II-1