AN IMPROVED DESIGN OF APPROXIMATE ARITHMETIC UNITS USING REVERSIBLE LOGIC GATES

A. Phanimadhuri [1] **J. Priyanka[2]**

[1]Department of ECE, M.Tech Asst.prof in lendi institute of engineering and technology

[2]Department of ECE, M.Tech Asst.prof in lendi institute of engineering and technology

Abstract:

In recent years, Reversible Logic is becoming more and more prominent technology having its applications in Low Power CMOS, Quantum Computing, Nanotechnology, and Optical Computing. Reversibility plays an important role when energy efficient computations are considered. In this paper, Reversible Full Adder/Subtractor with Design I, and Design II are proposed. The performance analysis is verified using number reversible gates, Garbage input/outputs and Quantum Cost. The reversible full adders are placed in the DMFA. In this paper a modified 8-bit reconfigurable CLA block is presented. The proposed system gives the best performance compared to the previous systems.

INTRODUCTION

Introduction to VideoEncoding:

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. One possible remedy is to adopt a conservative approach and use a very low DA in the hardware so that the output accuracy is not drastically affected. However, such a conservative approach will, as expected, drastically impact the power savings as well. This paper adopts a different approach to addressing this problem by dynamically reconfiguring the approximate hardware architecture depending on the inputs. Specifically, this paper makes the followingcontributions.

Introduction To reversible logicgates:

Traditional logic computation is irreversible since the outputs do not have enough information to reconstruct the inputs. In logic calculation, if there are 'p' inputs and 'q' outputs such that p > q, then at least (p-q) bits of information are lost. As an example, a logic gate with two inputs and one output destroys at least one bit of information during computation. Landauer's principle states that each bit of information that is disregarded results in dissipation of heat, regardless of underlying technology. The amount of dissipated heat is at least kTln2 joules for every bit of lost information, where k is the Boltzmann's constant and T is the absolute temperature. At room temperature, this amount becomes 2.9*10-21 Joules. According to the problem of heat dissipation arises from, (1) technological deviation from ideality of switches and materials and (2) Landauer'sprinciple.

Reversible Function:

The multiple output Boolean function F(x1; x2;… xn) of n boolean variables is called reversible, if: a. The number of outputs is equal to the number of inputs, b. Any output pattern has a uniquepre-image.

Reversible Logic Gate:

Reversible gates are circuits in which number of outputs is equal to the number of inputs and there is a one-to-one correspondence between the vector of inputs and outputs. It not only helps us to determine the outputs from the inputs, but also helps us to uniquely recover the inputs from theoutputs.

Ancilla Inputs/ Constant Inputs:

This refers to the number of inputs that are to be maintained constant at either 0 or 1 in order to synthesize the given logicalfunction.

Garbage Outputs:

Additional inputs or outputs can be added so as to make the number of inputs and outputs equal whenever necessary. This also refers to the number of outputs which are not used in the synthesis of a given function. In certain cases these become mandatory to achievereversibility.

BASIC REVERSIBLE GATES:

There are many number of reversible logic gates that exist at present. Some of the important reversible logic gates are

Not Gate:

The simplest Reversible gate is NOT gate and is a 1*1 gate.

Fig. 1.2.1 Block diagram of NOT gate

Feynman Gate:

The Feynman gate which is a 2*2 gate and is also called as Controlled NOT, is widely used for fan-out purposes. The inputs are (A, B) and outputs are P=A, Q= A XOR B.

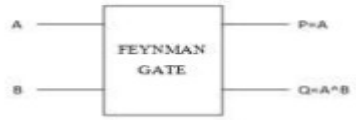


Fig. 1.2.2 Block diagram of Feynman gate

Fredkin Gate:

Fredkin gate is a 3*3 gate with inputs (A, B, C) and outputs P=A, Q=A'B+AC, R=AB+A'C.

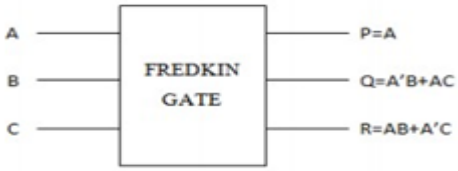


Fig. 1.2.3 Block diagram of Fredkin gate

Peres Gate:

Peres gate is a 3*3 gate having inputs (A, B, C) and outputs P = A; Q = A XOR B; R = AB XOR C.

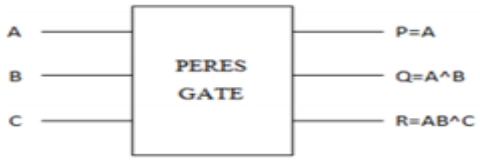


Fig. 1.2.4 Block diagram of Peres gate

Sayem Gate:

Sayem Gate is a 4*4 reversible gates built using reversible combination of Fredkin gate and Feynman gate.

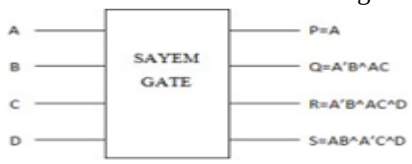


Fig. 1.2.5 Block diagram of Sayem gate

BME Gate:

BME Gate is a 4*4 reversible logic gate. The input vector is I(A,B,C,D) and the output vector is O(P,Q,R,S).

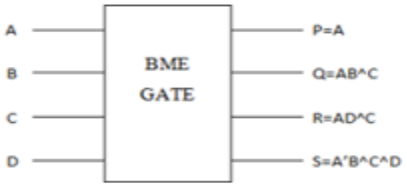


Fig. 1.2.6 Block diagram of BME gate

TSG Gate:

The TSG gate is a 4*4 reversible gate. The input vector is I (A, B, C, D) and the output vector is 0 (P, Q, R, S). The output is defined by P = A, Q = A'C' ^ B', R = (A'C' ^ B') ^ D and S =(A'C' ^ B').D ^ (AB ^ C). The proposed TSG gate is capable of implementing all Boolean functions and can also work singly as a reversible Full adder.

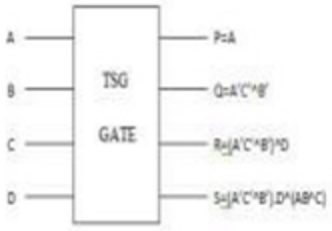


Fig. 1.2.7 Block diagram of TSG Gate

Introduction toVLSI:

Large-scale combination (VLSI) is the procedure of combining so as to make coordinated circuits a huge number of transistor-based circuits into a solitary chip. VLSI started in the 1970s when complex semiconductor and correspondence advancements were being produced. The chip is a VLSI gadget. The term is no more as normal as it once seemed to be, as chips have expanded in multifaceted nature into the countless transistors

LITERATURE REVIEW

The field of approximate computing has received significant attention from the research community in the past few years, especially in the context of various signal processing applications. Image and video compression algorithms such as JPEG, MPEG, etc., are particularly attractive candidates for approximate computing since they are tolerant of computing imprecision due to human imperceptibility, which can be exploited to realize highly power-efficient implementations of these algorithms. However, existing approximate architectures typically fix the level of hardware approximation statically and are not adaptive to input data. For example, if a fixed 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 paper addresses this issue by proposing a reconfigurable approximate architecture for MPEG encoders that optimizes power consumption while maintaining a particular PSNR threshold for any video. Experimental results show that our approach of dynamically adjusting the degree of hardware approximation based on the input video respects the given quality bound (PSNR degradation of 5-20%) across different videos while achieving a power savings of 13-18% over a conventional non- approximated MPEG encoder architecture. Although the proposed reconfigurable approximate architecture is presented for the specific case of an MPEG encoder, it can be easily extended to other DSPapplications.

PROJECT DESCRIPTION

This section describes the different steps ensued in constructing our proposed reconfigurable architecture and how it was embedded within the MPEG encoder.

MPEG CompressionScheme:

MPEG has for long been the most preferred video compression scheme in modern video applications and devices. Using the MPEG-2/MPEG-4 standards, videos can be squeezed to very small sizes. MPEG uses both interframe and intraframe encoding for video compression. Intraframe encoding involves encoding the entire frame of data, while interframe encoding utilizes predictive and interpolative coding techniques as means of achieving compression. The interframe version exploits the high temporal redundancy between adjacent frames and only encodes the differences in information between the frames, thus resulting in greater compression ratios. In addition, motion compensated interpolative coding scales down the data further through the use of bidirectional prediction. In this case, the encoding takes place based upon the differences between the current frame and the previous and next frames in the videosequence.

MPEG encoding involves three kinds of frames: 1) I-frames (intraframe encoded); 2) P-frames (predictive encoded); and 3) B-frames (bidirectional encoded). As evident from their names, an I-frame is encoded completely as it is without any data loss. An I-frame usually precedes each MPEG data stream. P-frames are constructed using the differences between the current frame and the immediately preceding I or P frame. B-frames are produced relative to the closest two I/P frames on either side of the current frame. The I, P, and B frames are further compressed when subjected to DCT, which helps to eliminate the existing interframe spatial redundancy as much aspossible.

Quality of aVideo:

The merit of the encoding operation can be determined from the output quality of the decoded video. Objective metrics, such as Peak Signal-to-Noise Ratio (PSNR), SAD, and so on, have a very good correlation with the subjective procedures of measuring the quality of the videos [16], [17]. Hence, we have utilized the popular and simple PSNR metric as a means of video quality estimation. PSNR is a full- reference video quality assessment technique, which utilizes a pixel-to-pixel difference with respect to the original video. In this paper, PSNR of a video is defined as the average PSNR over a constant number of frames (50) of thevideo.

Reconfigurable Adder/SubtractorBlocks:

Dynamic variation of the DA can be done when each of the adder/subtractor blocks is equipped with one or more of its approximate copies and it is able to switch between them as per requirement. This reconfigurable architecture can include any approximate version of the adders/subtractors. As a reference, Gupta et al. [9], [10] proposed six different kinds of approximate circuits for adders. However, it also needs to be ensured that the additional area overheads required for constructing the reconfigurable approximate circuits are minimal with sufficiently large power savings. As examples, we have chosen the two most naive methods presented in [2], namely, truncation and approximation 5, for approximating the adder/subtractor blocks. The latter one can also be conceptualized as an enhanced version of truncation as it just relays the two 1-bit inputs, one as Sum and the other as Carry Out (Choice 2). In case A, B, and Cin are the 1-bit inputs to the full adder (FA), then the outputs are Sum = B and Cout = A. The resultant truth-table [10] shows that the outputs are correct for more than half of all input combinations, thus proving to be a better approximation mode thantruncation.

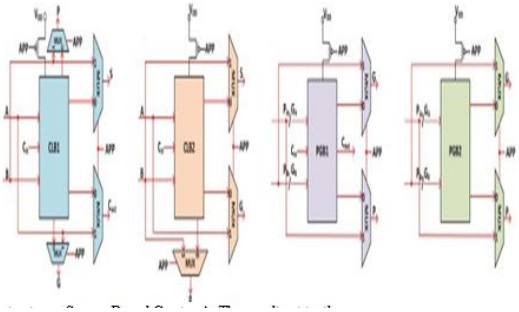


Fig 3.3.1 1-bit DMFA

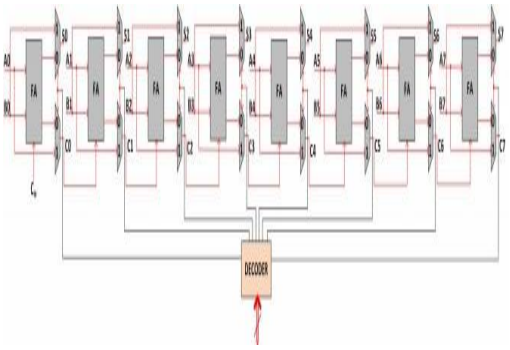


Fig 3.3.3 8-bit Reconfigurable RCA Block

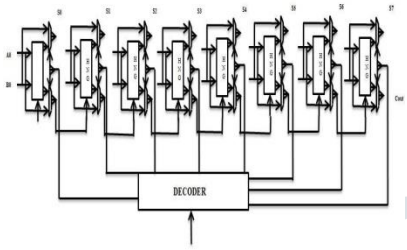


Fig 3.3.4 8-bit Reconfigurable RCA Block with HNG gate

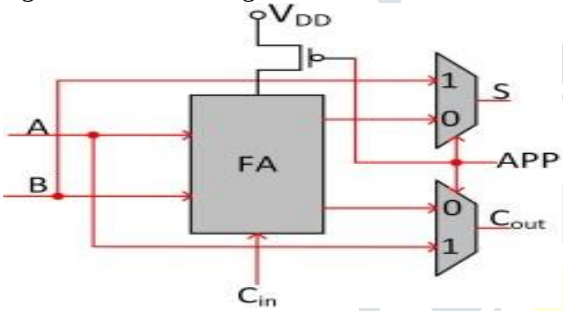
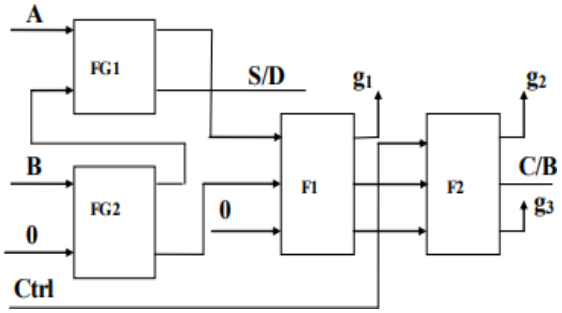


Fig 3.3.5 Dual Mode Carry Propagate Generate Block

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 complexity of the decoder block used for asserting the right select signals to the multiplexers as well as the logic overhead for the multiplexers themselves. 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.

The concept of RAB can also be extended to other adder architectures as well. Adder architectures, such as CBA and CSA, which also contain FA as the fundamental building block, can be made accuracy configurable by direct substitution of the



FAs with DMFAs. Other varieties, like CLA and tree adders, use different types of carry propagate and generate blocks as their basic building units, and hence require some additional modifications to function as RABs. As an example,we implemented a 16-bit CLA consisting of four different types of basic blocks (Fig. 8) 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. Their corresponding dual-mode versions, DMCLB1 and DMCLB2, have both S and P approximated by input operand B and both Cout and G approximated by input operand A, as shown in Fig. 7. The basic blocks present at the higher levels of CLA hierarchy are denoted as propagate and generate blocks, PGB1 and PGB2. In this case, PGB1 produces an extra Cout output as compared with PGB2. As shown in Fig. 7, the configurable dual-mode versions, DMPGB1 and DMPGB2, use inputs PA and GB as approximations for outputs P and G, respectively, when operating in the approximate mode. These approximations were selected empirically ensuring that the ratio of the probability of correct output to the additional circuit overhead for each of the blocks is large. Table I summarizes the outputs of each of the dual-mode blocks when operating in either accurate or approximatemode.

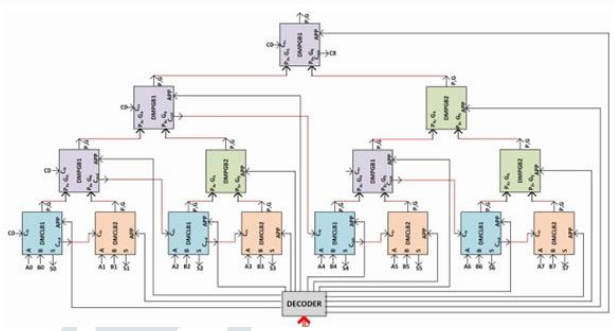


Fig 3.4.1 8-bit Reconfigurable CLA block

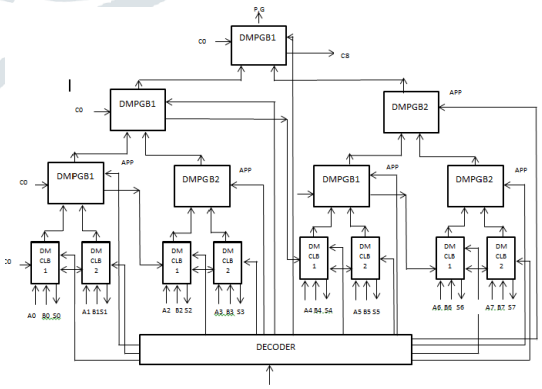


Fig 3.4.2 8-bit Reconfigurable CLA block using reversible logic gates

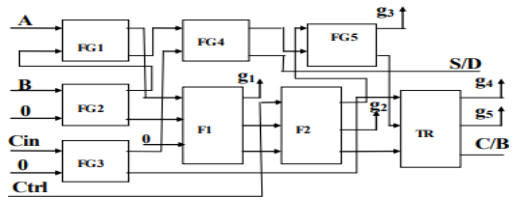


Fig 3.5.1: Reversible Half Adder/Subtractor

For a reconfigurable CLA, DMCLB1 and DMCLB2 blocks are approximated in accordance with the DA. However, the DMPGB1 and DMPGB2 blocks are approximated only when each and every DMCLB1, DMCLB2, DMPGB1, and DMPGB2 block, which belongs to the transitive fan-in cones of the concerned block, is approximated. Otherwise, the block is operated in the accurate mode.

Basic Block *(adder type)*	Outputs for APP = 0 (accurate mode)	Outputs for APP = 1 (approximate mode)
DMFA *(RCA, CBA, CSA)*	$S = A \oplus B \oplus C_{in}$ $C_{out} = AB + BC_{in} + AC_{in}$	$S = B$ $C_{out} = A$
DMCLB1 *(CLA)*	$P = A \oplus B$ $G = AB$ $S = P \oplus C_{in}$ $C_{out} = G + PC_{in}$	$P = B$ $G = A$ $S = B$ $C_{out} = A$
DMCLB2 *(CLA)*	$P = A \oplus B$ $G = AB$ $S = P \oplus C_{in}$	$P = B$ $G = A$ $S = B$
DMPGB1 *(CLA)*	$P = P_A P_B$ $G = G_B + G_A P_B$ $C_{out} = G + PC_{in}$	$P = P_A$ $G = G_B$ $C_{out} = G + PC_{in}$
DMPGB2 *(CLA)*	$P = P_A P_B$ $G = G_B + G_A P_B$	$P = P_A$ $G = G_B$

Table 3.4.1 Dual Mode Block Outputs for Accurate and Approximate Modes

For example, any DMPGB block at the second level of CLA can be made to operate in approximate mode, if and only if, both of its constituent DMCLB1 and DMCLB2 blocks are operating in the approximate mode. Similar protocol is ensued for the blocks residing at higher levels of the tree, where each DMPGB block can be approximated only when both of its constituent DMPGB1 and DMPGB2 blocks are approximated. This architecture can be easily extrapolated to other similar type CLAs, such as Kogge–Stone, Brent–Kung, Manchester-carry chain, and so on.

Half Adder/Subtractor

Reversible half Adder/Subtractor–Design I is implemented with four Reversible gates of which two F and two FG gates is shown in the Figure 4. The numbers of Garbage outputs are three represented as g1 to g3, Garbage inputs are two represented by logical zero and Quantum Cost is twelve as it requires two FG gates each costing one and two F gates each costing five each.

Fig 3.5.2:Full adder using HNG gate Full Adder/Subtractor

The Design I Reversible Full Adder/Subtractor with five FG, two F and a TR gate is asshownintheFigure5.ThethreeinputsareA,BandCinandtheoutput sare

Sum/Difference (S/D) and Carry/Borrow (C/B). The Control (Ctrl) input differentiates the Addition and Subtraction functionalities. For Ctrl value zero i.e., Logical low the circuit performs addition and Subtraction for Ctrl value one i.e., Logical high. The numbers of Garbage inputs are 3 represented by logical zero. The Garbage outputs are 5 represented by g1 to g5. The Sum/Difference function is realized from FG4 gate, and the Carry/Borrow function is realized from the output of TR gate. The Quantum Cost for five FG gates are five as each gate costs one, for two F gates is ten as each gate costs five, one TR gate costs six and total design Quantum Cost is21.

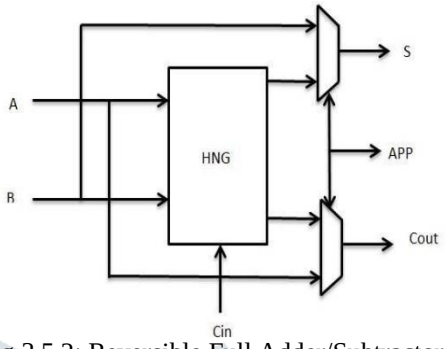


Fig 3.5.2: Reversible Full Adder/Subtractor

XILINX TOOL

Migrating Projects from Previous ISE SoftwareReleases:

When you open a project file from a previous release, the ISE® software prompts you to migrate your project. If you click Backup and Migrate or Migrate Only, the software automatically converts your project file to the current release. If you click Cancel, the software does not convert your project and, instead, opens Project Navigator with no project loaded.

LANGUAGE VERILOG HDL

In the semiconductor and electronic outline industry, Verilog is an equipment portrayal language (HDL) used to show electronic frameworks. Verilog HDL, not to be mistaken for VHDL (a contending dialect), is most generally utilized as a part of the outline, confirmation, and usage of digital rationale chips at the register-exchange level of reflection. It is likewise utilized as a part of the confirmation of analog and blended sign circuits.

System Tasks:

System tasks are available to handle simple I/O, and various design measurement functions. All system tasks are prefixed with $ to distinguish them from user tasks and functions. This section presents a short list of the most often used tasks. It is by no means a comprehensive list.

$display - Print to screen a line followed by an automaticnewline.

$write - Write to screen a line without thenewline.

$swrite - Print to variable a line without thenewline.

$sscanf - Read from variable a format-specified string.(*Verilog-2001)

$fopen - Open a handle to a file (read orwrite)

$fdisplay - Write to file a line followed by an automaticnewline.

$fwrite - Write to file a line without thenewline.

- $fscanf - Read from file a format-specified string. (*Verilog-2001)
- $fclose - Close and release an open filehandle.
- $readmemh - Read hex file content into a memoryarray.
- $readmemb - Read binary file content into a memoryarray.
- $monitor - Print out all the listed variables when any changevalue.
- $time - Value of current simulationtime.
- $dumpfile - Declare the VCD (Value Change Dump) format output filename.
- $dumpvars - Turn on and dump thevariables.
- $dumpports - Turn on and dump the variables in Extended-VCDformat.
- $random - Return a randomvalue.

RESULTS

Reconfigurable RCA Results:

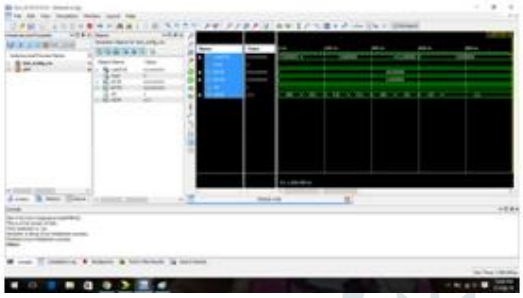


Fig 6.1.1: Reconfigurable RCA simulationresults

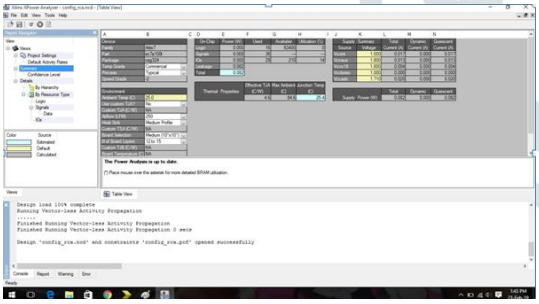


Fig 6.1.2: Reconfigurable RCA power analysis

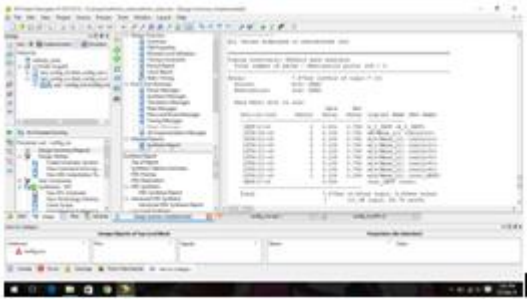


Fig 6.1.3: Reconfigurable RCA timing report

Reconfigurable CLAresults:

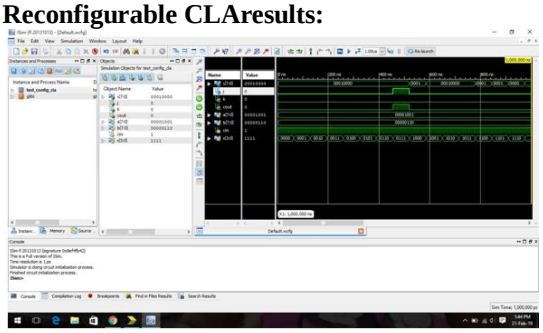


Fig 6.2.1: Reconfigurable CLA simulation result

Fig 6.2.2: Reconfigurable CLA power report

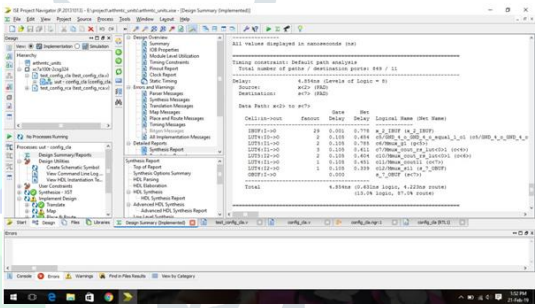


Fig 6.2.3 : Reconfigurable CLA timing report

Reversible Reconfigurable RCAResults:

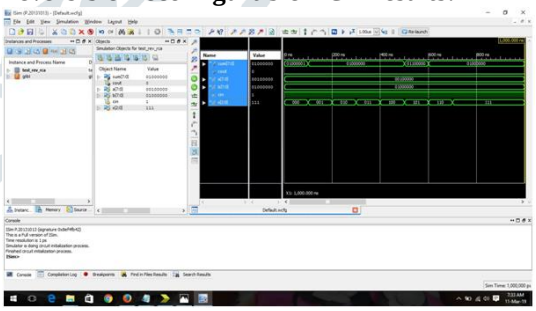


Fig 6.3.1: Reversible Reconfigurable RCA simulation result

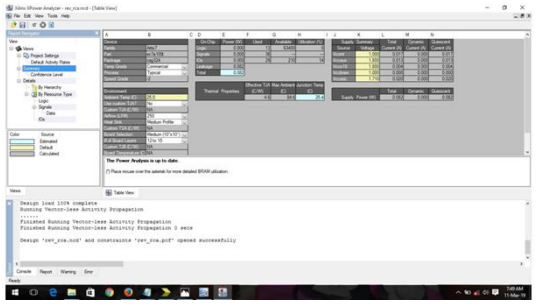


Fig 6.3.2: Reversible Reconfigurable RCA power report

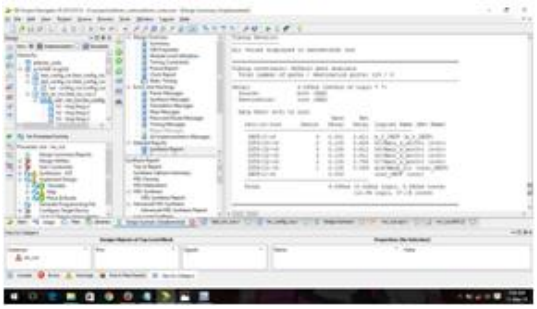


Fig 6.3.3 : Reversible RCA timing report

Reversible Reconfigurable CLAResult:

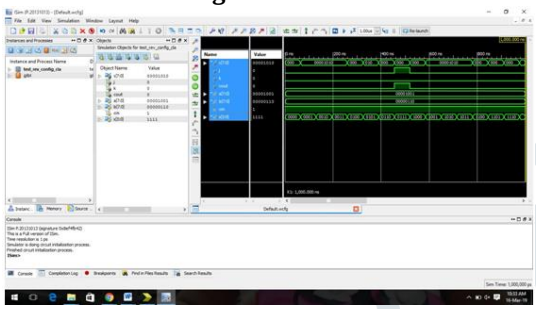


Fig 6.4.1: Reversible Reconfigurable CLA simulation results

Fig 6.4.2: Reversible Reconfigurable CLA powerreport

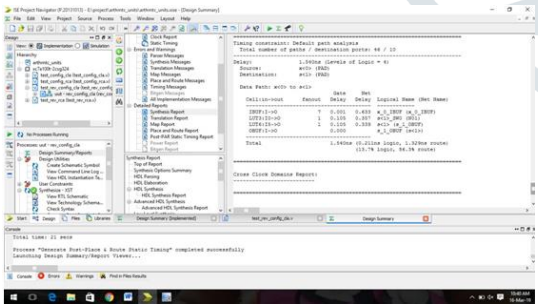


Fig 6.4.3: Reversible reconfigurable CLA timingreport.

Comparison of CLA and RCA reports:

	Timing Report	Power Report
Reconf_CLA	.854ns	.082W
Reversible_CLA	.542ns	.082W
Reconf_RCA	.475ns	.082W
Reversible_RCA	.088ns	.082W

Table 6.1: Results comparison

CONCLUSION

This paper proposed a reconfigurable approximate architecture for the MPEG encoders implemented using Reversible logic that optimize power consumption while maintaining output quality across different input videos. The proposed architecture is based on the concept of dynamically reconfiguring the level of approximation in the hardware based on the input characteristics. It requires the user to specify only the overall minimum quality for videos instead of having to decide the level of hardware approximation. Our experimental results show that the proposed architecture results in power savings equivalent to a baseline approach that uses fixed approximate hardware while respecting quality constraints across different videos.

Future work includes the incorporation of other approximation techniques and extending the approximations to other arithmetic and functional blocks

REFERRENCES

[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] J. George, B. Marr, B. E. S. Akgul, and K. V. Palem, “Probabilistic arithmetic and energy efficient embedded signal processing,” in Proc. Int. Conf. Compil., Archit., Synth. Embedded Syst. (CASES), 2006, pp.158–168.

[7] D. Shin and S. K. Gupta, “A re-design technique for datapath modules in error tolerant applications,” in Proc. 17th Asian Test Symp. (ATS), 2008, pp.431–437.

[8] S. Venkataramani, A. Sabne, V. Kozhikkottu, K. Roy, and A. Raghunathan, “SALSA: Systematic logic synthesis of approximate circuits,” in Proc. 49th Annu. Design Autom. Conf. (DAC), Jun. 2012, pp.796–801.