International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 7 | ISSN : 2456-3307

Design and Implementation of Reconfigurable Approximation Technique for Arithmetic Unit

I. S. Priya, B. Venkatesh
[1]M. Tech Student, Head Of The Department, ECE Shree Institute of Technical Education, Tirupathi, India
[2]Assistant Professor & Head Of The Department, ECE Shree Institute of Technical Education, Tirupathi, India

ABSTRACT

The research community in the last few years from the field of approximate computing has received significant attention, particularly in the context of different signal processing. Image and video compression algorithms such as JPEG, MPEG and so on, which can be exploited to realize highly power efficient implementations of these algorithms. However, existing approximate architectures typically fix the level of hardware approximations statically and are not adaptive to input data. This project addresses this issue by proposing a reconfigurable approximate for MPEG encoders that optimizes power consumption with the aim of maintaining a particular peak signal-to noise ratio threshold for any video. We propose two heuristics for automatically tuning the approximation degree of the RABs in these two modules during runtime based on the Characteristics of each individual video. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.3.

Keywords: Approximate circuits, approximate computing, low power design, quality configurable.

I. INTRODUCTION

Digital signal processing (DSP) blocks from the Backbone of various multimedia applications used in portable devices. Most of the DSP blocks implement image and video compression algorithms. Approximate computing architectures exploit the fact that a small relaxation in output correctness can result in significantly simpler and lower 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.

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), and so on have a very good correlation with the subjective procedures of measuring the quality of the videos. 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. PSNR of a video is defined as the average PSNR over a constant number of frames of the video. Images and videos differ in a variety of properties, such as color, resolution, brightness, contrast, saturation, blur, format, and so on. Thus, a naive static approximation technique, which provides satisfactory viewing quality for some specific types of videos, will fail to give adequate quality for some others. In that case, the viewing experience is significantly worsened if the approximate mode is not customized for the present type of video being watched. This is not possible for fixed hardware, and therefore a need arises for reconfiguring the architecture based on the characteristics of the video being viewed.

To support this claim, present the PSNR variation of different videos when encoded using an MPEG encoder that used a fixed approximation technique. An approximation mode, we have chosen approximation mode 5 from for implementing the fixed approximation hardware. We replaced all the adders/ subtractors in the ME and the DCT blocks with approximate versions.

II. APPROXIMATE ARITHMETIC UNITS

The introduction of approximate computing techniques has opened up entirely new opportunities in building low-power video compression architectures. Approximate computing methods achieve a large amount of power savings by introducing a small amount of error or inaccuracy into the logic block. Different approaches for approximation include error introduction through voltage over scaling intelligent logic manipulation, and circuit simplification using don’t care-based optimization techniques. The methods introduce imprecision by replacing adders with their approximate counterparts. The approximate adders are obtained by intelligently deleting 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. There also exist instances of approximations introduced in an MPEG encoder. Most of them exploit the inherent error resilience of the motion estimation (ME) algorithm, which results in minor quality degradation. For example, Moshnyagaet al. use a bit width compression technique to reduce power consumption of video frame memory. He and Liou and He et al. Use bit truncation to introduce approximations in the ME block of an MPEG encoder. An adaptive bit masking method is proposed in , where the authors propose to truncate the pixels of the current and previous frames required for ME depending upon the quantization step. 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.

This paper 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 show, helps in maintaining better control over application-level quality metrics while simultaneously reaping the power consumption benefits of hardware approximation. Our proposed technique can automatically adjust the extent of hardware approximation dynamically based on the video characteristics. In addition, such dynamic reconfiguration also provides users with a control knob for varying the output quality of the videos and the power consumption for the battery-powered multimedia devices.

III. MPEG COMPRESSION SCHEME

MPEG is mostly preferred for the video compression scheme in modern video devices and applications. MPEG- 2/MPEG-4 standards are used to squeeze 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 great ratios. In this case, the encoding takes placed based upon the differences between the current frame and previous frame in the video sequence.

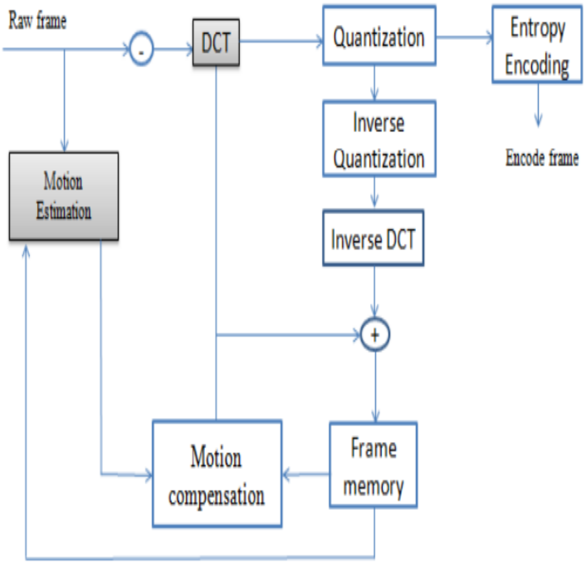


Figure 1. MPEG encoder block diagram

There are three kinds of frames used in MPEG encoding:

1. I-frames means intraframe encoded.
2. P-frames means predictive encoded.
3. B-frames means bidirectional encoded.

An I-frame is encoded as it is without any data loss and usually precedes each MPEG data stream. P-frames are constructed using the difference between the current frame and the immediately preceding I or P frame.B-frames are produced neighbor to the closest two I/P frames on either side of the current frame. The I, B and P frames are compressed when subjected to DCT. It is used to remove the existing frame. A significant portion of the interframe encoding is spent in calculating motion vectors (MVs) from the computed differences. Every non-encoded frame is divided into Macro blocks (MBs), such as 16 × 16 pixels. The Motion vectors (MVs) actually contain the information regarding the relative displacements of the Macro blocks (MBs) in the present frame in comparison with the reference.

Quality of a video

The advantage of encoding operation s used to find from the output quality of the decoded video peak signal-to noise ratio (PSNR), SAD, and so on are used to measuring the quality of video. PSNR metric as a means of video quality estimation. PSNR of a video means the average PSNR over a constant number of frames (50) of the video.

IV. PROPOSED ARCHITECTURE

Main Module's

1. 1-bit DMFA
2. Reconfigurable RCA block.
3. 1-bit dual-mode carry propagate generate blocks. Reconfigurable CLA block.

Module Description

1. 1-BIT DMFA

Reconfigurable Adder/Subtractor Blocks In degree of approximation is dynamically varied which can be done when each of the adder/subtractor blocks with one or more of its approximate copies. Reconfigurable Adder/Subtractor blocks are able to switch between them as per requirement and can include any approximation version of these blocks. 1-bit dual mode full adder is consists A, B, Cin are the inputs and outputs are Sum = A and Cout = A. When each full adder (FA) cell of the adder/subtractor with a dual mode full adder (DMFA) from the proposed scheme.

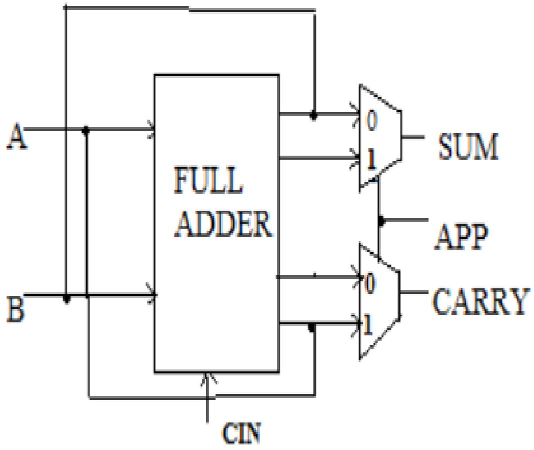


Figure 2. 1 bit DMFA

In which each full adder cell can perform operating either in fully accurate or in some approximation mode depending upon the state of the control signal APP. When operating in the approximate mode the full adder act as power gated. Dual-mode full adder can operated in either the two approximation modes. Approximation was selected for its higher probability of giving the accurate output result than the truncation. In which does not variably outputs 0 irrespective of the input.

2. 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 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.

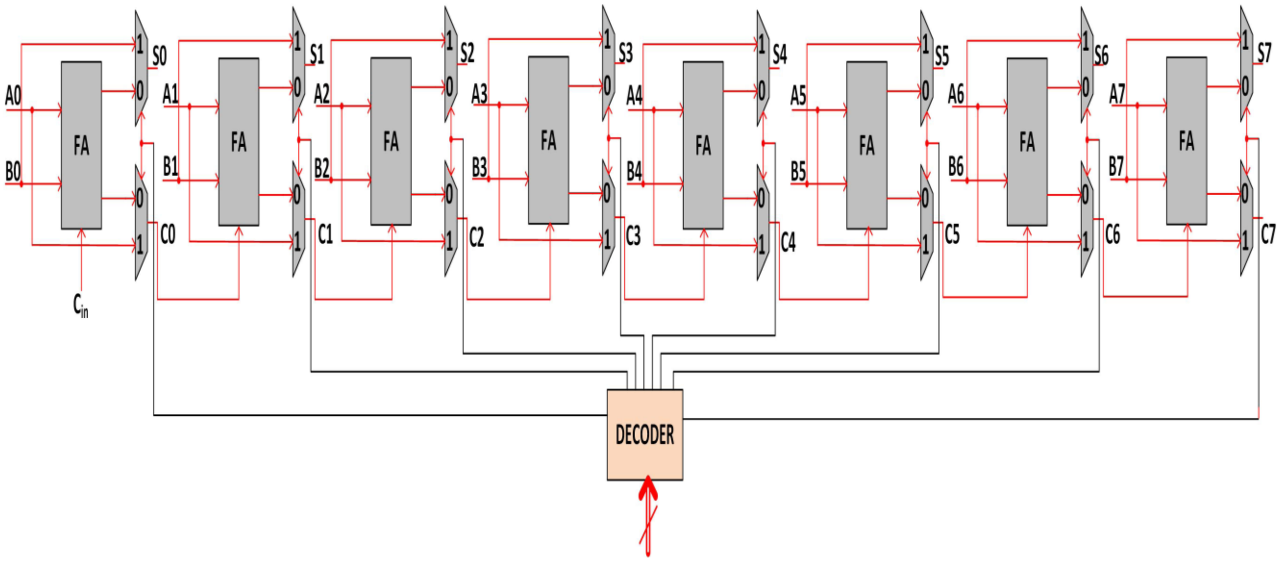


Figure 3. 8-bit reconfigurable RCA block

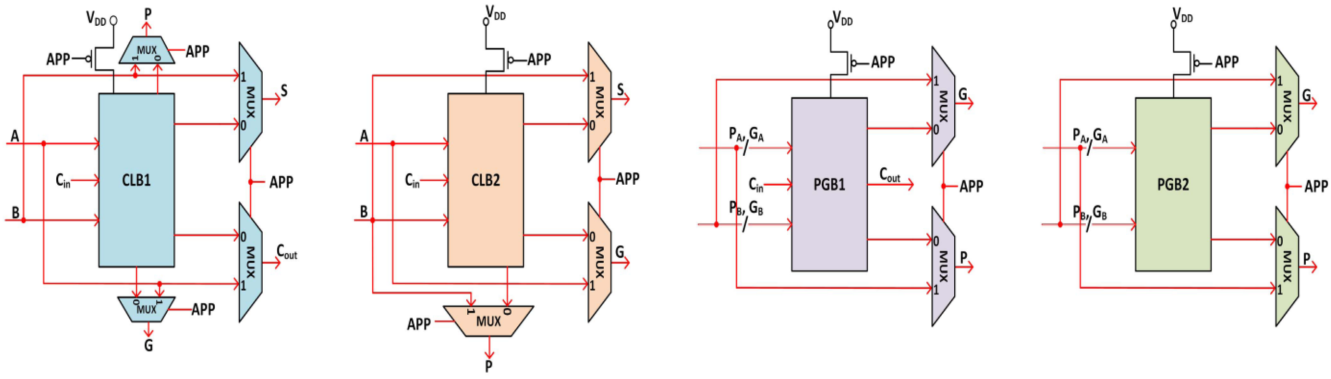


Figure 4. 1-bit dual-mode carry propagate generate blocks

1. 8-Bit Reconfigurable Carry Look ahead 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 look ahead blocks, CLB1 and CLB2. The difference among them being that CLB1 produces an additional Cout signal compared with CLB2. Their corresponding to the dual-mode versions, DMCLB1 and DMCLB2, have both sum S and propagate P approximated by input operand B and both Cout and generate G approximated by input operand A, as shown in figure 4. The basic blocks present in the higher levels of carry look ahead adder CLA hierarchy are represented as the configurable as propagate P and generate G blocks, PGB1 and PGB2.

In this case PGB1 generate an extra Cout output as compared with PGB2. As shown in figure 3, the configurable dual-mode versions, DMPGB 1 and DMPGB 2, use inputs Pa and Pb as approximations for outputs propagate and generate. These approximations ensuring that the ratio of the probability of match output to the additional circuit overhead for each of the blocks is large. Table 1 show and realize the additional circuit overhead for each of the Dual-mode full adder blocks. When operating in either accurate or else approximate mode. Reconfigurable of Carry look ahead adder (CLA), Dual-mode carry look ahead blocks such as DMCLB1 and DMCLB2 blocks are approximated in according with the Dual-mode (DA).

However the Dual-mode propagate generator blocks such as DMPGB1 and DMPGB2 blocks approximated when each and every Dual-mode carry propagate generator blocks such as DMCLB1, DMCLB2, DMPGB1 and DMPGB2 block, which belongs to the transitive fan-in cones of the concerned block is approximated. Otherwise, the block is performed in the accurate mode. For example, any Dual-mode propagate generator blocks (DMPGB) block at the second level of the carry look ahead adder (CLA) can be performed in approximate mode, and both of its constituent DMCLB1 and DMCLB2 blocks are performed in the approximate mode.

In each DMPGB block can be approximated only when both of its constituent DMPGB1 and DMPGB2 blocks are approximated. This architecture can be realize extrapolated to other similar type Carry look ahead adders (CLAs), and so on

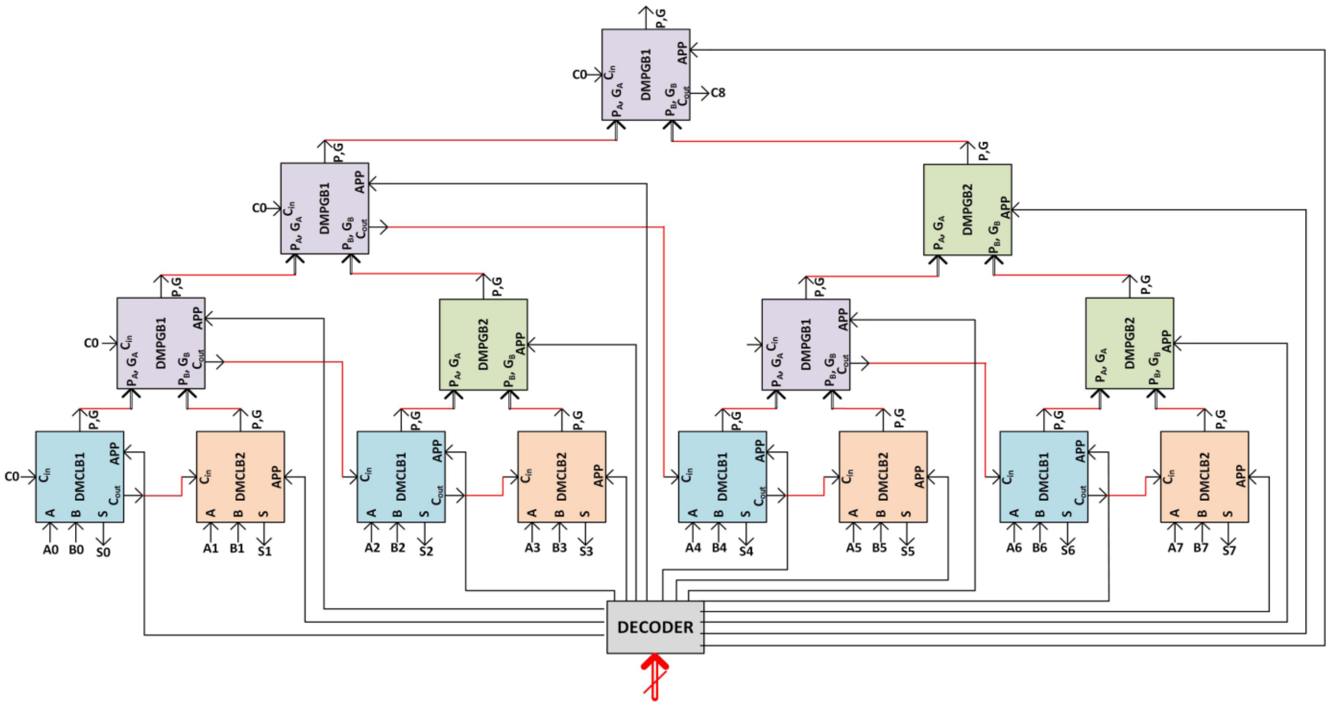


Figure 5. 8-bit reconfigurable CLA block

Table 1. dual-mode block outputs for accurate and approximate modes

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)	$P = P_A P_B$ $G = G_B + G_A P_B$	$P = P_A$ $G = G_B$

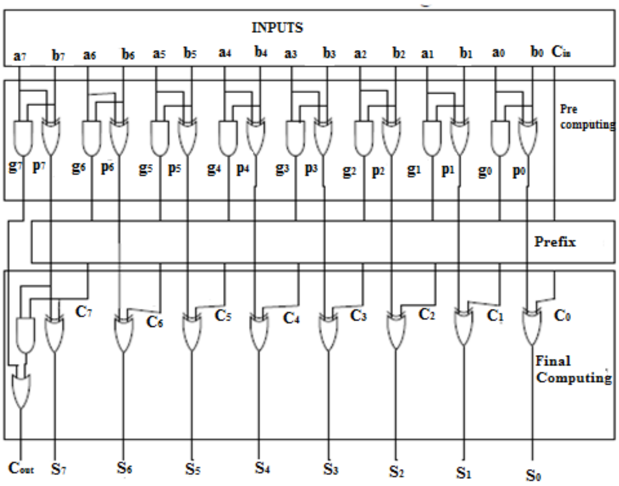

INPUTS
a7 b7 a6 b6 a5 b5 a4 b4 a3 b3 a2 b2 a1 b1 a0 b0 Cin
g7 p7 g6 p6 g5 p5 g4 p4 g3 p3 g2 p2 g1 p1 g0 p0
Pre computing
Prefix
C7 C6 C5 C4 C3 C2 C1 C0
Final Computing
Cout S7 S6 S5 S4 S3 S2 S1 S0

Figure 6. Parallel-Prefix structure with carry save notation

V. ENHANCEMENT WORK

A. Parallel Prefix Adder Structure

Parallel-prefix structures are found to be common in high performance adders because the delay is logarithmically proportional to the adder width. PPA's basically consists of 3 stages

- Pre computation
- Prefix stage
- Final computation

Implementation stages of PPA

PPA's basically consists of 3 stages:

Pre-Computation Stage

In pre computation stage, propagates and generates are computed for the given inputs using the given equations (1) and (2).

Pi=Ai^Bi(1)

Gi=Ai.Bi (2)

Prefix Stage

In the prefix stage, group generate/propagate signals are computed at each bit using the given equations. The black cell generates the ordered pair and the grey cell generates only the left signal. The fundamental carry operator is denoted by the symbol "o". The generate and propagate signals are combined using the fundamental operator as shown in equation (3).

$(g_L, p_L) \circ (g_R , p_R) = (g_L + p_L.g_R , p_L. g_r)$ (3)

The black cell and grey cell logic definition are shown in Figure 6.

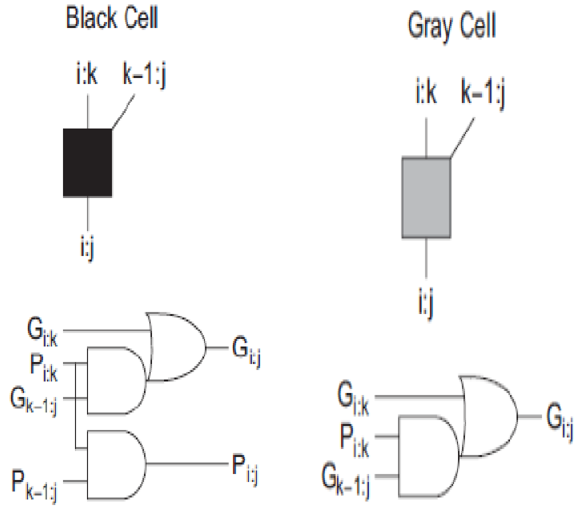


Figure 7. Black and Grey Cell logic definition

From the black and grey cell logic definition, the generate and propagate signals are given by equations (4) and (5).

$G_{i:k} = G_{i:j} + P_{i:j} .G_{j-1:k}$ (4)

$P_{i:k}= P_{i:j} . P_{j-1:k}$(5)

The "o" operation will help make the rules of building prefix structures (6)

$G_{i:k} : P_{i:k} =(G_{i:j} , P_{i:j})\circ(G_{j-1:k} , P_{j-1:k})$ (6)

Final Computation

In the final computation, the sum and carryout are the final output. The generate/propagate signals can be grouped in different fashion to get the same correct carries. Based on different ways of grouping the generate/propagate signals, different prefix architectures can be created.

$S_i= P_i.G_{i-1:-1}$ (7)

$C_{out}= G_{n:-1}$ (8)

B. 16-bit Spanning Tree Adder

As the FPGA uses a fast carry-chain for the RCA, it is interesting to compare the performance of this adder with the other adders. Also of interest for the spanning-tree CLA better in area and delay.

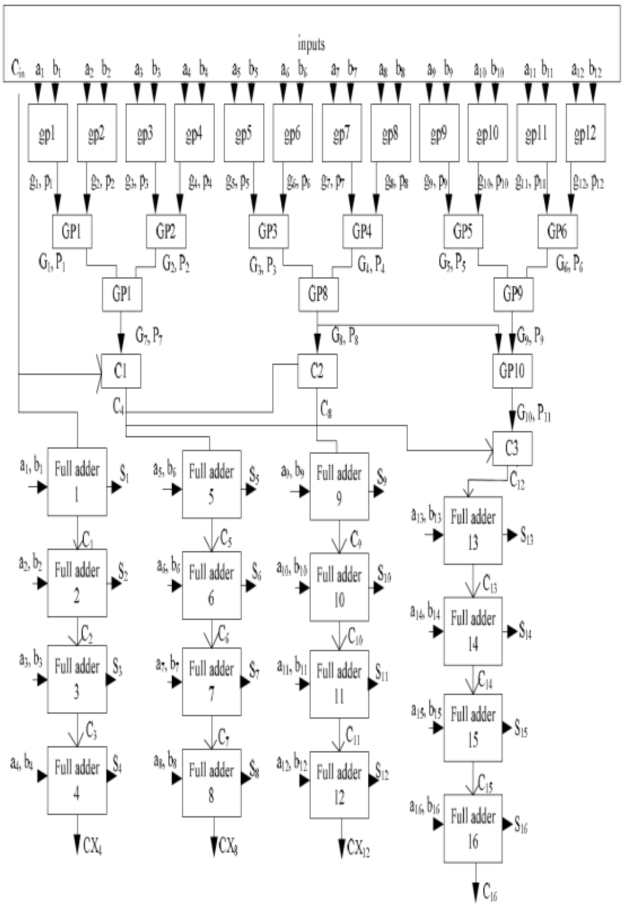


Figure 8. 16-bit Spanning Tree Adder

Spanning tree applications:

In processors (DSP) and microprocessor data path units, adder is an important element. As such, extensive research continues to be focused on improving the power-delay performance of the adder. In VLSI implementations, parallel adders are known to have the best performance.

VI. RESULTS

Block diagram

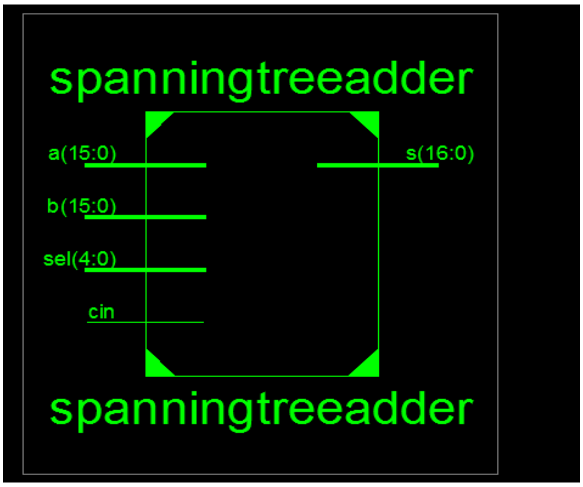


Figure 9

RTL schematic

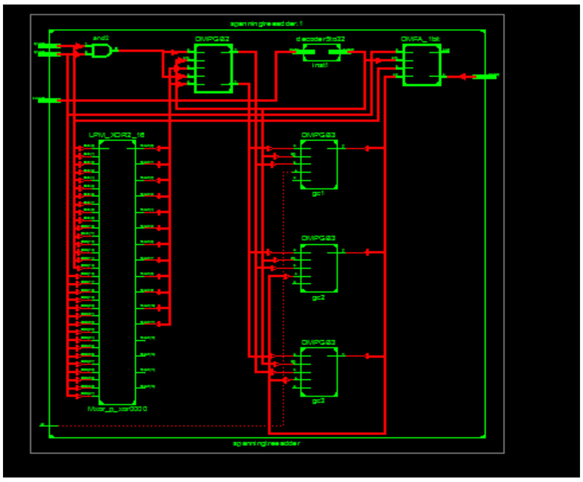
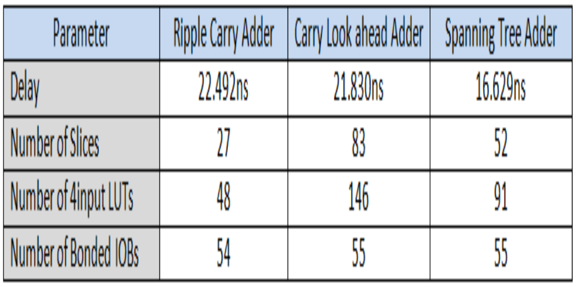


Figure 10

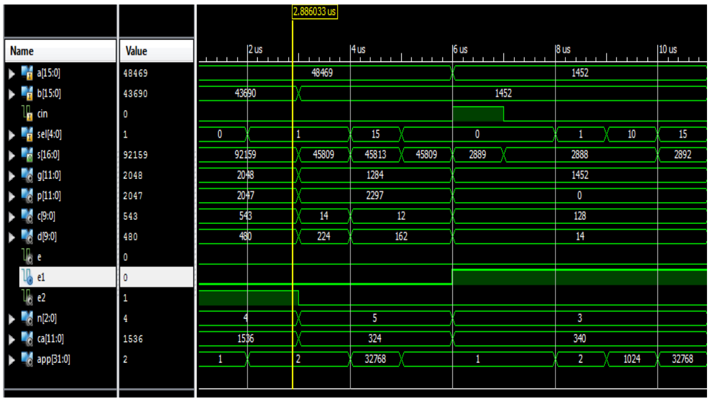
Comparison table

Table 2. Delay and Area report



Parameter	Ripple Carry Adder	Carry Look ahead Adder	Spanning Tree Adder
Delay	22.492ns	21.830ns	16.629ns
Number of Slices	27	83	52
Number of 4input LUTs	48	146	91
Number of Bonded IOBs	54	55	55

Simulation results



Name	Value
a[15:0]	48469
b[15:0]	43690
cin	0
sel[4:0]	1
s[16:0]	92159
g[11:0]	2048
p[11:0]	2047
c[9:0]	543
d[9:0]	480
e	0
e1	0
e2	1
n[2:0]	4
ca[11:0]	1536
app[31:0]	2

Figure 11

VII. CONCLUSION

We proposed a reconfigurable approximate architecture for the MPEG encoders 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 delay and area. Future work includes the incorporation of other approximation techniques and extending the approximations to other arithmetic and functional blocks. The simulation results are shown Xilinx 14.3 software.

VIII. 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]. 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.

[9]. V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for low-power approximate computing," in Proc. 17th IEEE/ACM Int. Symp. Low-Power Electron.Design (ISLPED), Aug. 2011, pp. 409–414.

[10]. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Lowpowerdigital signal processing using approximate adders," IEEETrans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[11]. V. G. Moshnyaga, K. Inoue, and M. Fukagawa, "Reducing energy consumption of video memory by bit-width compression," in Proc. Int.Symp. Low Power Electron. Design (ISLPED), 2002, pp. 142–147.

[12]. Z. He and M. L. Liou, "Reducing hardware complexity of motion estimation algorithms using truncated pixels," in Proc. IEEE Int. Symp.Circuits Syst. (ISCAS), vol. 4. Jun. 1997, pp. 2809–2812.

[13]. Z.-L. He, C.-Y. Tsui, K.-K. Chan, and M. L. Liou, "Low-power VLSI design for motion estimation using adaptive pixel truncation," IEEETrans. Circuits Syst. Video Technol., vol. 10, no. 5, pp. 669–678, Aug. 2000.

[14]. A. Raha, H. Jayakumar, and V. Raghunathan, "A power efficient video encoder using reconfigurable approximate arithmetic units," in Proc.27th Int. Conf. VLSI Design, 13th Int. Conf. Embedded Syst., Jan. 2014, pp. 324–329.

[15]. P. M. Kuhn, Algorithms, Complexity Analysis and VLSI Architectures for MPEG-4 Motion Estimation, 1st ed. Norwell, MA, USA: Kluwer,1999.

[16]. K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, "Study of subjective and objective quality assessment of video," IEEETrans. Image Process., vol. 19, no. 6, pp. 1427–1441, Jun. 2010.

[17]. S. Winkler, "Video quality measurement standards—Current status and trends," in Proc. 7th Int. Conf. Inf., Commun., Signal Process. (ICICS), Dec. 2009, pp. 1–5.