Design of Low Area Approximate Arithmetic Units for Video Encoding

PADIMI. GANGA BHAVANI[1], N.RAJESH BABU[2], DR.S.V.R.K. RAO[3]

[1]*PG Student, Dept of ECE, GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY, Rajahmundry, East Godavari, India.*

[2]*Asst.PROFESSOR, Dept. of ECE, GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY, Rajahmundry, East Godavari, India.*

[3] *HEAD OF THE DEPARTMENT (HOD) AND PROFESSOR, Dept of ECE, GODAVARI INSTITUTE OF ENGINEERING AND TECHNOLOGY, Kakinada, East Godavari, India.*

(E-Mail : gangabhavanipadimi4@gmail.com)

Abstract: The approximate computing is necessary and very much important in the image and video processing applications. The exact computing has attracted the researchers for video and image processing applications such as MPEG and JPEG. As they provide slight visual change in the output after compression, which makes the above said algorithms are more desirable. If we have used fixed hardware configuration, the output quality has not adaptive to input data. This is due to fixed approximation in the hardware. The degradation in the quality of the output can be overcome by varying the Degree of Hardware Approximation. If we take an example of an MPEG encoder, which has static hardware approximation, the output quality is degraded due to the outputs are not in the output quality bound. This can be overcome by automatic change of Degree of Approximation in the hardware depending on the input. Reconfigurable Arithmetic Units are most useful in changing Degree of Approximation In the Motion Estimation and Discrete Cosine Transform modules of the MPEG encoder. The Reconfigurable Arithmetic Units such as Ripple Carry Adder (RCA), Carry Look Ahead Adder (CLA) are designed to get better quality of the output videos, But they require more area, delay and execute the results at low speed. In this project we have implemented Kogge-Stone adder in place of Carry Look Ahead block to get better Speed, reduced Delay and area and Power consumption for compression scheme of An MPEG encoder.

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

I INTRODUCTION

The controlled inaccuracy in the image and video processing algorithms, results in a slighter visual change in the output, which compression algorithms are the most attractive for approximate computing architectures. Approximate computing architectures accomplish the fact that a small relaxation in the correctness of the output can result in specifically simpler and low power implementations. Even though the most of the hardware architectures proposed so far suffer from the limitation that, the output quality degradation for dynamically changing input parameters. In that case, it is very difficult to provide a quality bound on the output, and in some of the cased, the quality of the output may be severely degraded. The main cause for this output quality variation is that the degree of approximation (DA) in the hardware architecture is static and cannot be personalized for different inputs. By using a conservative method, we can use a low DA in the configuration of the hardware, we can prevent the output accuracy degradation, so that the output accuracy is not much effected.

MPEG COMPRESSION

MPEG is the most popular video compression technique nowadays. By using the MPEG-2/MPEG-4 formats, the desired video can be squashed into very small sizes. Both inter frame and intra frame encoding are used by the MPEG. The Intra frame encoding encodes the entire frame of data, whereas the inter frame encoding uses the predictive and interpolative coding techniques for compression. The inter frame encoding exhibits the high temporal redundancy between adjacent frames and only encodes the variation between the frames, which results in better compression efficiency. In addition, motion compensated interpolative coding reduces the data further through by using bidirectional prediction technique. MPEG encoding contains three kinds of frames, they are listed below

1) I-frames (intraframe encoded);
2) P-frames (predictive encoded); and
3) B-frames (bidirectional encoded).

Each I-frame encodes the whole frame of data at a time, without loss. An I-frame precedes each MPEG stream of data. The P-frames are formed by the variation between the current frame and the immediately before I frame or the current frame and immediately before P frame. B-frames are produced among the

nearest two I/P frames on either side of the current frame. All the frames mentioned above are further compressed when applied to DCT, it helps to prevented the inbuilt inter frame spatial redundancy as much as possible.

A part in the inter frame encoding is used in computing motion vectors (MVs). Every raw frame is squeezed to small

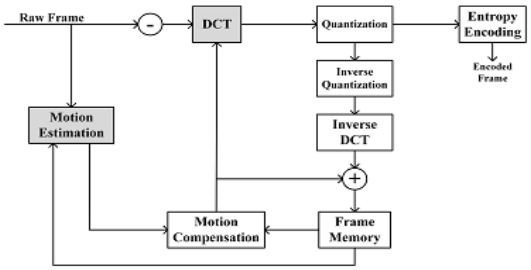


Fig1. MPEG encoder block diagram.

macro blocks (MBs). Each and every MV has it's corresponding MB. These Motion Vectrors consist of information about the change of the places of the MBs in the present frame in comparison with the reference MBs in the reference frame. They are determined by deriving the minimum value of sum of absolute differences (SADs) of the MB in the current frame with respect to all the corresponding MBs in the reference frame. The resultant vectors are also encoded along with the frames. But It is not enough to present an exact explanation of the actual frame. Hence, in addition to the MVs, a residual error is determined, which is then minimized using DCT. The ME and DCT blocks are the expensive modules in the MPEG encoder. The various steps involved in performing MPEG compression are shown in Fig.1.

OUTPUT QUALITY OF A COMPRESSED VIDEO

The excellence of the encoding operation can be measured the quality of the output decoded video. The parameter such as Peak Signal-to-Noise Ratio (PSNR), SAD have the better relation with the quality of the videos. Therefore, PSNR is considered as the most effective performance parameter for measuring the video quality estimation. Peak Signal to Noise Ration (PSNR) and it is a full-reference video quality assessment technique, it uses the pixel to pixel change with the original video.

APPROXIMATE ADDERS CIRCUITS

In multiplication adder circuits are used to generate the result. The two adders that are most widely used are Ripple Carry Adder (RCA) and Carry Look Ahead Adder (CLA). In the RCA, each carry in the previous full adder is rippled to the next stage. Due to that the next state full adder must wait to get the carry from the previous full adder, which makes the RCA is slower when compared to other adders. In case of the CLA, it generates the carries before calculating the sum, by using the propagate and generate signals. It reduces the delay and increases the speed. the delay of CLA is logarithmic in n (or O(log(n))), thus significantly shorter than that of RCA in n bit CLA. But it requires larger area, power consumption when compared to the RCA, which makes the circuit complexity is more. The area complexity of CLA is O(nlog(n)) in an n bit CLA. There are Many approximation methods that are proposed so far, for minimum critical path and complexity of the hardware in accurate adder. The methodology proposed so far is based on a speculative operation. In speculative adder, sum bit is predicted from the previous least significant bit, which is less than the number n, in case of n bit speculative adder. A speculative designing of adder, gets faster speed compared conventional designing of the adder. Segmented designing of adders is proposed, which are implemented by operating number of small adders in parallel fashion. Therefore, the carry propagation is trimmed into small segments. The Segmentation is used for carry propagation, but the selection of the carry inputs are chose differently, this is called as Carry Select Adder. Another method for reducing the critical path delay and power dissipation of a conventional adder is by approximating the full adder; the approximate adder is usually applied to the LSBs of an accurate adder.

II RELATED WORK

It is difficult to construct energy-efficient video compression schemes for MPEG encoding operation. There are different techniques used to reduce the power consumption, such as modifications in the algorithms, voltage over scaling, inaccurate computations of parameters. By introducing small error in the logic block, a large amount of power saving can be achieved by the Approximate methods. Approximate computing techniques have provided new opportunities to build video compression with low power consumption. Using different methods of approximation, we can introduce inaccuracy by replacing the adder circuits with their approximate counterparts and the approaches include intelligent logic manipulation, circuit simplification by using don't care based optimization and voltage over scaling. Approximate adders are constructed by removing some of the transistors in the mirror adder circuit. The important thing that we have to note that these approximate circuits are hardwired and can't be changed without re-synthesizing the whole circuit.

III EXISTING APPROXIMATE ARITHMETIC UNITS

RECONFIGURABLE ADDER/SUBTRACTORS

The reconfigurable adder/subtractor blocks have the ability to change their degree of approximation depending upon the input data. The reconfigurable adder/subtractor blocks have introduced in the motion estimation (ME), discrete cosine

transform (DCT) modules in MPEG encoder. This can be done by replacing the basic circuits with the approximate circuits. In some of the adders , Full adder is the fundamental block, it should be changed to the approximate full adder by replacing the full adder block with the Dual Mode Full Adder. In the Dual Mode Full Adder, it has two modes, one is Accurate mode and the other one is Approximate mode. Depending upon the requirement it is operated by the selection of control signal APP. If the control signal is high, it is operated in approximate mode otherwise, it is operated in accurate mode. The below figure shows the Dual Mode Full Adder circuit

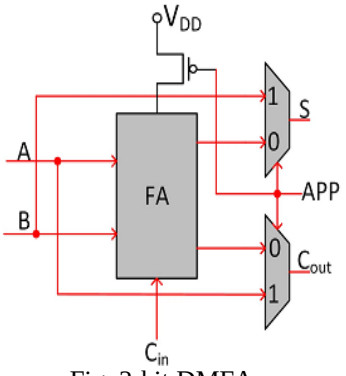


Fig. 2-bit DMFA

The Above figure shows Dual Mode version of the full adder, it contains 1 bit inputs A, B, Cin and the outputs are Sum (S) and Carry Out (Cout). In the Approximate mode Sum (S) is approximated to the input B and Carry out (Cout) is approximated to input A.It is important to note that the FA cell is power-gated when operating in the approximate mode. The power consumption of a 16-bit RCA were performed in Synopsys Design and Power Compiler. The results shown that there is negotiable difference in the power consumption in the two modes. Fig. 2 shows the logic block diagram of the DMFA cell, which replaces the FA cells of an 8-bit RCA, as shown in Fig.3.

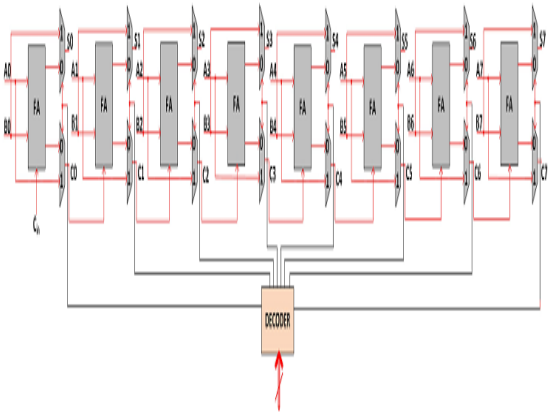


Fig.3. 8-bit reconfigurable RCA

The above figure shows the reconfigurable RCA, in which each full adder is replace by Dual mode full adder, then it can be operated in either accurate mode and in approximate mode, depending upon our requirement. Additionally it has approximate controller and multiplexers for the selection of the right select signals. In other adder adders like CLA and tree like architectures, Propagate and Generate blocks are the basic blocks. We use different types of propagate and generate blocks and needs additional modifications to function as Reconfigurable blocks. The below shows the Control Logic Blocks and Propagate and Generate Blocks.

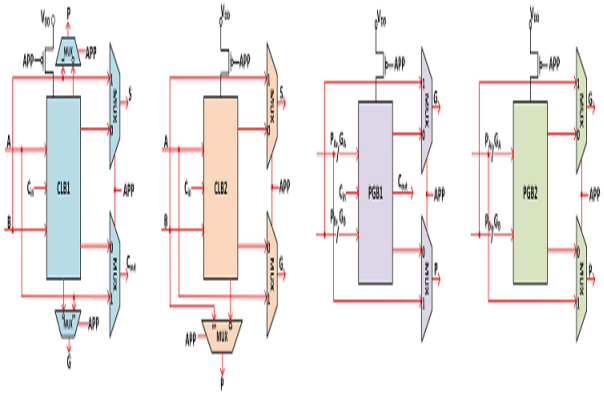


Fig.4. 1-bit dual-mode carry propagate generate blocks

In these Control Logic Blocks (CLB), the inputs are A, B, Cin and the outputs are Sum (S) and Carry out (Cout). In the dual mode approximation versions, the Sum (S) and Propagate (P) signals are approximated to input B and the Carry out (Cout) and Generate (G) signals are approximated to the input (A). In case of Propagate Generate blocks (PGB), the Propagate (P) and Generate (G) signals are approximated to PA and GB in their dual mode versions.

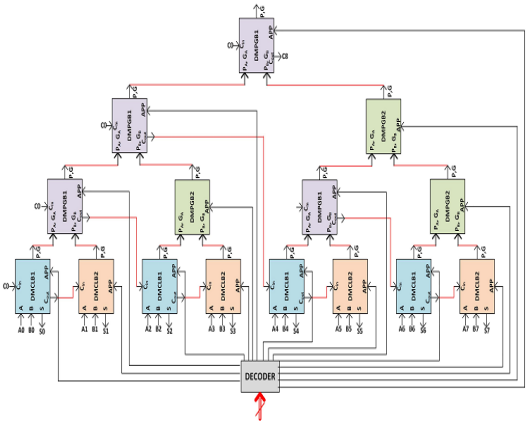


Fig.5. 8-bit reconfigurable CLA block

. Table I shows the outputs of each of the dual-mode blocks in accurate or approximate modes.

TABLE-I
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)	$P = P_A P_B$ $G = G_B + G_A P_B$	$P = P_A$ $G = G_B$

The approximations were chose, depending upon the ratio of the probability of correct output to the additional circuit overhead for each and every blocks is high. In a reconfigurable Carry Look Ahead adder, DMCLB1 and DMCLB2 blocks are approximated according to the DA of the CLB1 and CLB2. The Dual mode blocks, DMPGB1 and DMPGB2 blocks are approximated only when the corresponding DMCLB1, DMCLB2, DMPGB1,and DMPGB2 blocks are approximated Otherwise, the block is operated in the accurate mode.

IV PROPOSED ARITHMETIC UNIT

kogge-stone adder is a parallel prefix formation of Carry Look-ahead Adder. It generates the carries in parallel. It is the fastest adder with based on designing time. It is the common choice for high performance adders in industry. The Kogge-Stone Adder was first developed by Peter M. Kogge and Harold S. Stone in 1973. In the proposed architecture CLA Adder is replaced by Kogge-Stone adder**.** In KSA, carries are computed fast by computing them in parallel at the cost of increased area. It has three processing stages for calculating the sum bits. Working of KSA:

(I) Pre-processing: In this step the computation of generate and propagate signals corresponding too each pair of bits in A and B. These signals are given by the logic equations below:

Pi = Ai xor Bi

Gi = Ai and Bi

(II) Carry look ahead network: This block differentiates KSA from other adders and is the main force behind its high performance. This step involves computation of carries corresponding to each bit. It uses group propagate and generate as intermediate signals which are given by the logic equations below:

BLACK CELL: The black cell takes two pairs of generate and propagate signals (Gi, Pi) and (Gj, Pj) as input and computes a pair of generate and propagate signals (G, P) as output

G = Gj OR (Gi AND Pj)

P = Pi AND Pj

GREY CELL**:** The gray cell takes two pairs of generate and propagate signals (Gi, Pi) and (Gj, Pj) as inputs and Computes a generate signal G as output

G = Gj OR (Gi AND Pj)

(III) Post processing: This is the final step and is common to all adders of this family (carry look ahead). It involves computation of sum bits. Sum bits are computed by the logic give

Si = pi xor Ci-1.

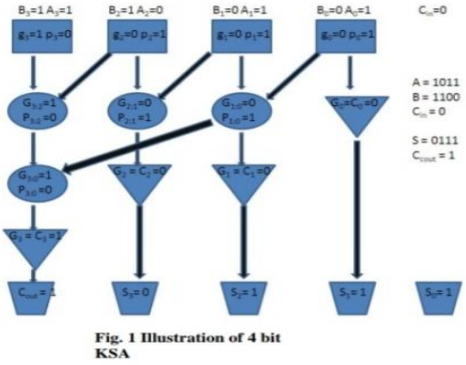


Fig. 1 Illustration of 4 bit KSA

Fig 6. 4-Bit Kogge-Stone adder

The 4-bit Kogge-Stone adder is mentioned above, in this every vertical column creates propagate and generate bits. The carries produced in the previous stage are in XOR'd operation with the first propagate bit after giving the input to the stage. The sum bit is produced by XORing the first propagate with the last stage carry that is S present = P present Xor C before.

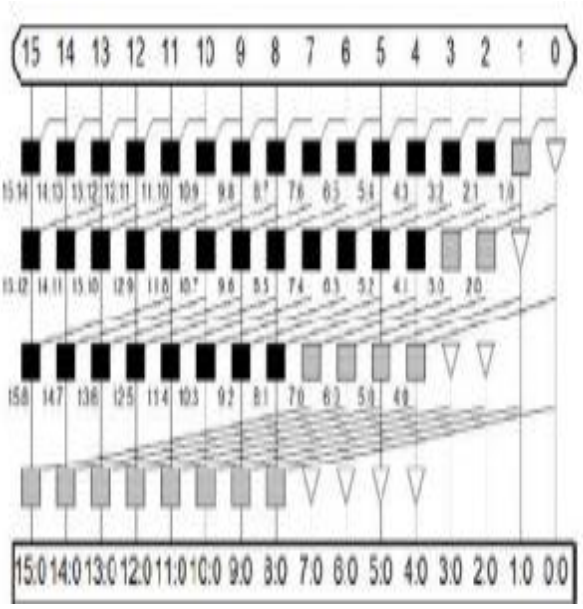
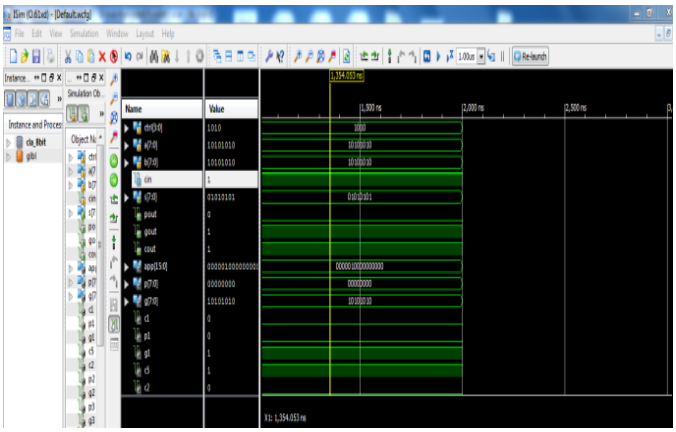


Fig.8: 16-bit KS Adder

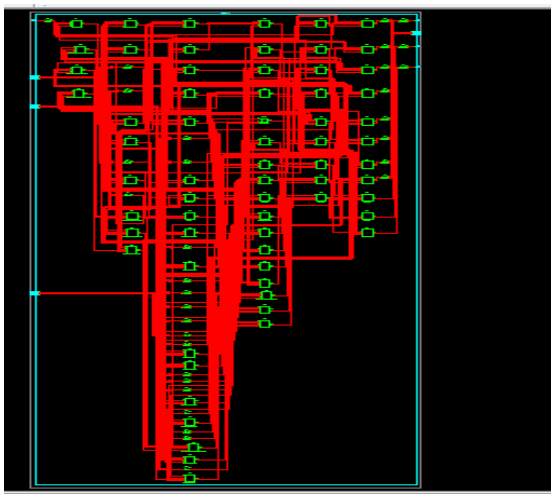
The 32-bit Kogge-stone construction structure and pin diagram is shown above, which is designed using xilinx software .

SIMULATION RESULTS

Proposed Simulation Result:



RTL Schematic:



Design Summary:

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	37	4656	0%
Number of 4 input LUTs	65	9312	0%
Number of bonded IOBs	32	232	13%

Timing Summary:

```
Delay:               13.835ns (Levels of Logic = 10)
  Source:            ctrl<0> (PAD)
  Destination:       s<5> (PAD)

  Data Path: ctrl<0> to s<5>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             19   1.106   1.074  ctrl_0_IBUF (ctrl_0_IBUF)
     LUT4:I0->O             5   0.612   0.541  de/Mdecod_out01 (app<0>)
     LUT4:I3->O             1   0.612   0.426  pg3/old_g_4_and00001_SW0 (N64)
     LUT4:I1->O             2   0.612   0.532  pg3/old_g_4_and00001 (N18)
     LUT4:I0->O             1   0.612   0.387  pg7/old_g_4_and0000162_SW0 (N74)
     LUT4:I2->O             3   0.612   0.454  pg7/old_g_4_and0000162 (N17)
     LUT4:I3->O             2   0.612   0.532  pg3/cout2 (c6)
     LUT4:I0->O             1   0.612   0.360  m5/cout1 (c3)
     LUT4:I3->O             1   0.612   0.357  m6/s1 (s_5_OBUF)
     OBUF:I->O                  3.169          s_5_OBUF (s<5>)
    ----------------------------------------
    Total                      13.835ns (9.171ns logic, 4.664ns route)
                                       (66.3% logic, 33.7% route)
```

V CONCLUSION

This paper proposes the designing of the approximate architectures with low area, because the architectures proposed so far are the approximate arithmetic units such as reconfigurable adder/subtractor blocks. The reconfigurable adder/subtractor blocks such as Reconfigurable RCA, Reconfigurable CLA have taken optimum power consumption, more area and less speed, compared with the extended architecture. In the extended architecture, Kogge-Stone adder increases the speed, reduces the delay when compared to the other architectures proposed so far. It can be used in high performance industries as well.

I. 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 digitalTV: 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," inProc. 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," inProc. 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," inProc. Int. Conf. Compil., Archit., Synth. Embedded Syst. (CASES), 2006, pp. 158–168.

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