A DYNAMIC DESIGNING AND IMPLEMENTATION OF ACCURATE MULTIPLIER FOR DIGITAL SIGNAL PROCESSING APPLICATIONS

[1]Y RAVINDER, [2]DHARMAJI MUNJAM, [3]FARHEEN MARYAM, [4]CHITYALA PRAVALIKA
[1, 2,3]Assistant Professor, [4]Student, Dept. of Electronics and Communication Engineering, Brilliant Institute of Engineering and Technology, Hyderabad, Telangana, India

ABSTRACT
The design of accurate multipliers which were applied in image processing has many applications and they lead to less delay and power. In any general purpose processor the use of conventional full precision multipliers results in increase in the power, area and computational time. So, multipliers being the basic key element ofany computation unit take its own importance in decreasing the power as well as increase in the speed. In this project, design and implementation of accurate multiplier for digital signal processing applications is done. The two input multiplier and multiplicand are divided in to two groups of bits each according to their computational significance in product generation. Scaling of these bits and ordering can be done by using a shifter. Then the multiplication can be performed and generates partial products that are aligned. After that an accurate adder is used to add the products and carry propagation is also performed in order to obtain an accurate final product.
KEYWORDS: Accurate multiplier, Shifter, partial products, accurate adder, carry propagation.

INTRODUCTION

Numerous inexorably well knownapplications, for example, picture preparingand acknowledgment, are inalienablytolerant of little mistakes. These applicationsare computationally requesting and increaseis their major number juggling capacity,which makes a chance to exchange offcomputational exactness for decreased power utilization. Basically, fault tolerant system produces the unpleasant figuring in the applications of fault tolerant [1]. Because of this the trade of f delay and power will provide accurate results. In someapplications, the fault tolerant applicationrequires distinctive precision. This will mainly used in the program stages of fault tolerance system. Here the programs stages will run depend upon the precision control. If the accuracy is fixed then the precision is not required. In the same waythe accuracy is not fixed then the precisionis required. Hence for diverse applications the precision requirement is necessary. There should be reconfigurable of multipliers in various program stages applications. So in this paper we designed a multiplier which will control the precision ineffective way [2].
Approximate computing is an alluring worldview for advanced preparing at nano metric scales. Estimated processing is especially intriguing for PC math structures. The examination and structure of two new evaluated blowers for use in a multiplier [3].These plans depend upon various highlights of weight, to such an extent, that imprecision in calculation (as assessed by the foul up rate and the alleged standardized fumble clear) can meet concerning circuit-based figures of estimation of a structure. The fragmentary outcomes of the multiplier are changed to present moving likelihood terms.Reason whimsy of measure is moved for the aggregate of balanced divided things dependent on their likelihood. The proposed system is used in two assortments of multiplier.
The need to support advanced sign handling (DSP) and course of action applications on imperativeness obliged devices has reliably created. Such applications routinely generally perform matrix increments using fixed-point number juggling while in the meantime appearing for some computational missteps [4]. Accordingly, improving the imperativeness capability of duplication isessential. At long last, the exhibited

computational mistake does not make anyprominent effect on the nature of DSP and the precision of characterization applications.

LITERATURE SURVEY

We study some of the research performed onthe design of estimated multipliers in thissegment. The input operands were cut to m bit in the dynamic segment (DSM) method depending on the location of the leading element, while the values were truncated using a fixed width multiplication. The resultant production is often smaller than the actual one by this method of truncation, producing a negative mean relative error. It is unacceptable since the medium loss near to zero would be a larger signal to noise (SNR) ratio when coping with optical signal management with roughly integer units that have the Gaussian error distribution.

The partial products were produced and accumulated in approximate radix-4 boothmultipliers in [5]. In addition, an estimated radix-9 multiplier was introduced to generate the smallest significant bits of the triple multiplicand, using estimatedadditives. In [6], the most relevant bits of themultiplier were coded by identical encoding of radix-4 and the lowest bits were encoded with a higher approx. encoding of the radix that rounded the least meaningful bits to the nearest power of two. In [7] many estimated additional elements were known as building blocks and a design space was investigatedto find the right solution for this approximate multiplier. Another solution is to adjust the counting scheme to the logarithmic scheme such that the speed of multiplication is increased rather than compounded. This method produces the logarithm of the input operands, determines the total of them and conducts an antilogarithm procedure on the total toobtain the final result. This method iscomplex due to the logarithm and antilogarithm steps produced.

In the reduction was suggested and worked with four estimated 4:2 compressors. The multiplier rates. Appropriate 4:2 compressorand an error recovery element have been suggested to improve the precision of themultiplication. In [8], the largest estimated multiplier arrangement has provided a range of estimated 5:3 compressors in an about15:4 compressor. It should be stated that the most significant elements of the test were obtained by utilizing precise compressors to improve the accuracy. Many compressors estimate were suggested. A concept algorithm for effective estimated multipliers made of such compressors has also been suggested.

In the complex DRUM structure [9], the least important bit of the truncated input wasset to "1" for the truncation procedure tobring the MRE to zero. In LETAM structure,the input operands were truncated and halfof partly produced components were ignoredduring the multiplication. Delays andelectricity usage have also been increasedwith the removal of the incomplete goods relative to those in the DSM and DRUM systems. The input operand in RoBA multipliers was rounded to the next power oftwo in which certain multiplication, addition and subtraction processes provided the production. Throughout this company, In relation to the same multiplier, which led to improved energy and rpm, the amount of elements to incorporate in order to produce the final effect was decreased. In a certain manner, the least important bits of the component items have been eliminated [10] to increase the multiplier 's pace and size. A easy way to construct partial products is to multiply any multiplier portion, which can simply be achieved through logical AND action. Another method is to encrypt andsubtract an encoded multiplier by multiplier in higher radixes. The encoding of the multiplier becomes more complicated as the radix decreases. And it should be reduced Approximate encoders may be used toproduce partial products utilizing thiscomplexity.

system. The main intent of register multiplier is to store the bit representation and give polynomial output a(t). Here parallel load operation is performed in the most significant bit position. In the sameway left shift operations are performed in MSB bit. The multiplicand bit is used b(t) value to store the value in register. The parallel load operation is also applied in the multiplicand. The obtained value is stored inthe register. The right shift operation is performed in the multiplicand register block.Crypto core processor is used to

transfer the data in multiplicand register.

PROPOSED SYSTEM

The below figure (1) shows the architecture of accurate multiplier for digital signalprocessing applications. The entire system isdivided into following modules. The following modules are inputs X and Y, partial products ($A_{H,}$ A_L and $B_{H,}$ B_L), shifter, accurate adder, Carry propagation and register. First the input numbers are divided into two parts each based on theirsignificance level while product computation as lower significant and higher significant bits. Now shifter will shift thedata words by sequential bits. Then the alignment of partial products generator will be done. Now these bits perform the addition operation using accurate adder and give the final product by subsequentlyperforming the carry propagation. After that partial product takes this registers and generates final output.

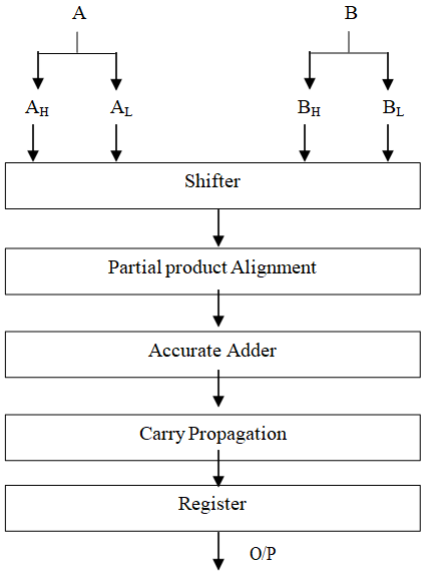

Fig. 1: Proposed System

Here firstly, the operands are loaded in the multiplier in which say A and B are input nits they are divided in to the highersignificant bits and lower significant bits as $A_{H,}$ A_L and $B_{H,}$ B_L. The arithmetic operationslike addition and multiplication operationsare performed. The obtained result of this will be saved in the shifter. Here irreducible polynomial function is not used in the

The barrel shifter consists of root and load Band this are taken as input to this block. The multiplier register is generally attached tothe finite field arithmetic circuit. In the sameway, multiplicand register consists of shift, data_in and load_B bits which are taken as input to the barrel shifter. It will shift thedata and as well as load the data in effective way. Result register consists of output and saves the entire arithmetic result. Compared to existed system, the proposed system giveseffective results. The result multiplier and multiplicand is saved in the result barrel shifter block. The both a (t) and b (t) values are assigned in the barrel shifter blocks. The obtained values in the barrel shifter block will shift the bits to adder block. This block will perform the addition operation. Afterperforming particular operation, the bits are shifted to the result register. This result register will save the output as product. At last the barrel shifter will perform the parallel operation in effective way.

Partial-Product

Partial-Product Multiplication is analternative method for solving multi-digit multiplication problems. This is a strategy that is based on the distributive (grouping) property of multiplication. The first partial product is created by the LSB of the multiplier, the second partial product is created by the second bit in the multiplier, etc. The final partial products are added witha accurate adder circuit.

Shifter

A barrel shifter is a digital circuit that can shift a data word by a specified number of bits without the use of any sequential logic, only pure combinational logic.

Accurate Adder

The accurate block in the proposed architecture uses full adders, half adders, and approximate adders to provide improvedresult in the higher order bits. The accurate block is extended from MSB to LSB for N bits, to provide optimized power delayproduct (PDP) at the cost of least error in the overall output. The accurate adder is decidedby architecture/system-level applications. A self configuration technique has been proposed for the scenarios where architecture/system- level choice is either unclear or difficult. A carry is propagated through several consecutive bits because of the actual path delay is large. When the actual carry propagation chain is short, there is no need to use approximation configuration, which is intended to cut carry chain shorter. The accurate adder is decided by architecture/system-level applications. A self configuration technique has beenproposed for the scenarios where architecture/system-level choice is either unclear or difficult. A carry is propagated through several consecutive bits because of the actual path delay is large. When the actual carry propagation chain is short, there is no need to use approximation configuration, which is intended to cut carry chain shorter.

Carry Propagation

Here first inputs are given to the propagator and generator unit. This unit will generate and propagated the input signals to adder logic. This logic will control the overall operation of the system. The adder will perform the addition operation in parallel form. Because of this the speed of operation is reduced.

RESULTS

The Xilinx design environment was used to implement and examine the developed algorithm. The FPGA architecture ofproposed accurate multiplier design is shown in Fig. 2 and Fig. 3. The below Fig. 2and Fig. 3 show the RTL schematic and technology schematic of proposed accurate multiplier. RTL (Register-transfer logic) schematic is the combination of inputs and outputs. Technology schematic is thecombination of Look up tables, Truth Tables, K-Map and equations. This schematic is generated after the optimization and technology targeting phase of the synthesis process.

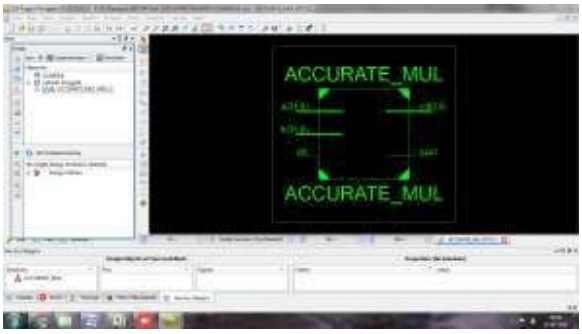

Fig. 2: RTL SCHEMATIC OF PROPOSEDMETHOD

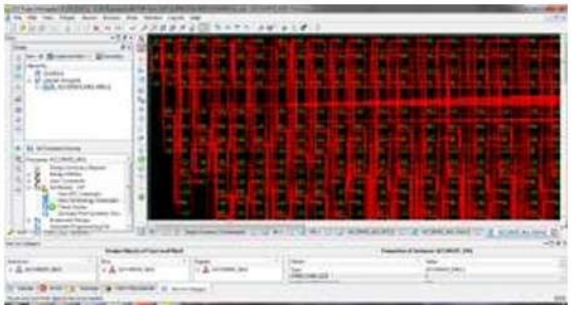

Fig. 3: TECHNOLOGY SCHEMATIC OFPROPOSED METHOD

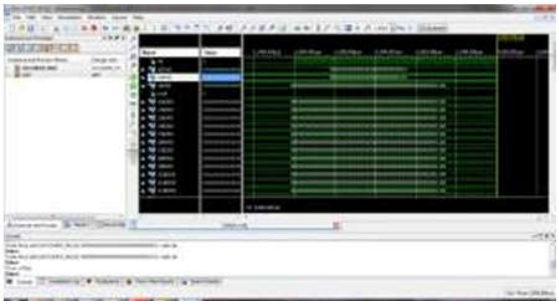

Fig. 4: OUTPUT WAVEFORMS OF PROPOSEDMETHOD

CONCLUSION

Design and implementation of accurate multiplier for digital signal processing applications is presented in this document. Our proposed approach divides themultiplicand and multiplier into two halves based on the significance as lower order and higher order. Each half is multiplied with theother and our architecture is made accuracy- configurable. The proposed multiplier architecture is uses four-stage structures to get final output such as shifter, accurate

adder, Carry propagation and register. Shifter is used to shift the data words bysequential bits. Then resultant partial products are order in the alignment of partial products generator. The aligned partial products are then added using accurate adder and give the final product by subsequently performing the carry propagation. After that partial product takes the registers and generates final output. The proposed multipliers has a high speed of operation as well as area effective implementation because of employing divide and conquerapproach in the design.

REFERENCES

[1] Mary Swarna, Latha Gade, S Rooban, "Design of Fault Tolerant Multiplier UsingSelf checking adder and GDI Technique", 2021 6th International Conference for Convergence in Technology (I2CT), 2021

[2] M. S. Ansari, B. F. Cockburn and J. Han,"A Hardware-Efficient LogarithmicMultiplier with Improved Accuracy," 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 928-931, 2019

[3] Tongxin Yang, Tomoaki Ukezon, Toshinori Sato, "A Low-Power High-Speed Accuracy-Controllable ApproximateMultiplier Design" 978-1-5090-0602- 1/18/$31.00 ©2018 IEEE.

[4] Yajuan He, Xilin Yi, Bin Ma, Ziji Zhang, Bo Zhang, "A ProbabilisticPrediction Base Fixed-Width Booth Multiplier", IEEE Asia PacificConference on Circuits and Systems (APCCAS), 2018

[5] S. Venkatachalam, H. J. Lee, and S.-B. Ko, "Power efficient approximate booth multiplier," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Florence, Italy, May2018, pp. 1–4

[6] V. Leon, G. Zervakis, D. Soudris, and K. Pekmestzi, "Approximate hybrid high radix encoding for energy-efficient inexact multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 3, pp. 421– 430, Mar. 2018.

[7] I. Alouani, H. Ahangari, O. Ozturk, and S. Niar, "A novel heterogeneousapproximate multiplier for low power and high performance," IEEE Embedded Syst.Lett., vol. 10, no. 2, pp. 45–48, Jun. 2018.

[8] R. Marimuthu, Y. E. Rezinold, and P. Mallick, "Design and analysis of multiplier using approximate 15–4 compressor," IEEE Access, vol. 5, pp. 1027– 1036, 2017.

[9] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Austin, TX, USA, Nov. 2015, pp. 418–425.

[10] H. J. Ko and S. F. Hsiao, "Design and application of faithfully rounded and truncated multipliers with combined deletion, reduction, truncation, and rounding," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58, no. 5, pp. 304–308,May 2011.