High Speed and Low Power Error Recovery Approximate Multiplier for Image Processing Applications

B. Aparna¹, Mr. B. Girirajan²

¹PG Scholar, Department of ECE, SR University, Warangal, Telangana, India.
²Assistant Professor, Department of ECE, SR University, Warangal, Telangana, India.

Abstract. In approximating, the requirement for accurate results is ignored because of some other applications with better performance in terms of area or delay. Multipliers are key arithmetic circuits in many of these applications including digital signal processing (DSP). Here in this paper we propose an approximate multiplier circuit with two various architectures where one is designed by modifying the design of the circuit by adding AND-OR logic approximation in the partial product generation stage and a dual quality adder at the final stage of the multiplication for error recovery. The other one is by adding the same AND-OR logic approximation in the partial product reduction stage where it is applied to only least significant part and accurate adders are used in the remaining for final outcome. In addition, many errors make little noticeable variations in practice, for instance image processing owing to human perceptual restrictions. Error-tolerant algorithms and their utilization inspired the progress of inexact multipliers which trade-off between power efficiency, area and speed. The proposed approximate multipliers have been shown that both have a lower area and one with better accuracy and error recovery other with delay than an exact Wallace multiplier and existing approximate designs. Functional analysis has shown that on a statistical basis, the proposed multipliers have considerable error distances and thus, they achieve a high accuracy and better area. To show the effectiveness of the work we had implemented and image processing application with the help of Xilinx System generator. The total designs are done and implemented in Xilinx ISE 14.7 with Verilog HDL coding.

Keyword: Approximate Multiplier, Error Recover, Image Processing Applications, Low Power

1. Introduction

Multiplication is one of the basic arithmetic operations in various VLSI applications. The major factors in VLSI (very large scale integration) speed, power, area and delay. Multipliers are most widely used arithmetic and logic unit, DSP processors, FIR filters, floating point units. The basic building blocks of Multipliers are using by full adders and half adders. These are having low power and less delay. Different types of multiplication algorithms are used to achieve minimum power and less delay. In Digital Signal Processing (DSP), which performance improved with loss of accuracy for application of approximate circuits? In existing system, a approximate multiplier with low power and small critical path is deliver for high performance DSP applications. An approximate adder is developed in previous only with Luminance (Luma) concept. AM1 and AM2 are described for error recovery, which error has hand over from approximate adder and then multiplexing. In proposed system, to avoid the error correcting block, by applying bit weight based compression technique we designed a multiplier By using this method we reduce the area of the unsigned multiplier circuits with better accuracies. Some technologies, including image processing and recognition, which are increasingly popular, are intrinsically tolerant of small imprints. Such programs are computer-intensive and their basic arithmetical function provides a way of changing computational precision for reduced power consumption. Approximate computation is an effective approach for error-resistant applications, as it can compensate for variance in energy, as it plays a significant role in such applications at present. We exhibit quicker and power-efficient column compression multiplication with extremely little overheads by utilizing a blend of two strategies: segment of the fractional items
into two sections for independent vertical segment compressor and speeding up of the last expansion by utilizing new hybrid architectures of adders is discussed in. A detailed scrutiny of adder circuits for power efficient multipliers is presented in. In trending VLSI technology QCA adders and multipliers are also in high demand. A reconfigurable energy efficient CMOS adder circuit is discussed in. Various error-tolerant programs have various reliability criteria, as do different software system levels. The energy is reduced if high precision is not required when multiplication precision is calculated. In other words, estimated multipliers should be reconfigured dynamically in order to meet the varying precision specifications of various process phases and implementations. Vedic Mathematics got profound space in designing high speed and accurate multipliers. Functional approximation deals with the design of approximation of arithmetic units such as adders and multipliers, most of the approximate multipliers designed at the higher abstraction levels i.e. RTL, application and gate. In general multipliers are classified as serial and parallel. Parallel multipliers are faster than serial, and the architecture of parallel multipliers are of two different types, they are 1. Array Multipliers and 2. Tree Multipliers. Array multipliers are faster, but they are complex in design, they give large footprint. Tree multipliers are faster than array, they lead to small footprint. It consists of shift and add logics. Approximation of multipliers is obtained by three techniques. i. Approximation of operands. ii. Approximate multipliers designed at the higher abstraction levels i.e. RTL, application and gate. The major characteristics and the limits of works in regarding estimated multipliers is summarized, the higher speed, minimization of silicon region used and optimization of device parameters is required. Features dealing with techniques of logical extraction techniques. To achieve energy efficiency and higher speed, minimization of silicon region used and optimization of device parameters is required. The major characteristics and the limits of works in regarding estimated multipliers is summarized, the multiplier conditions, for example, will enable the less important parts to be removed. More reduction in energy can be obtained by removing more pillars, but at the expense of increase in mistakes. One of the other methods for better energy efficiency is by modular re-designing of combinational logic circuit with low complexity [4]. Multipliers could be constructed with tiny estimated multipliers for energy efficiency, but at the cost of mistakes spreading as the magnitude of the multiplier grows. This is achieved by structuring tiny sets in a hierarchical model. The paper [5] suggests the technique of software-based perforation technique. It suggests acquiring the optimum number of partial products by trading off precision in area and power. There are many area and power efficient methods suggested by modifying the Functional behavior ranging from transistor level to architectural level. The automated architecture approaches discussed in [6] uses current layout stream to synthesize the estimated circuits by using the system operation models and applying performance boundaries. The DWT needs many mathematical functions to be performed for its execution. As the numbers of calculations are more, it is required to know the hardware necessities for such compound calculations. Generally there are 2methods for performing DWT. One method is Convolution based method where filter bank structures are implemented. The second one is Lifting scheme which decreases the number of computations and avoids the use of multiple operations. The benefit of DWPT is that it is conceivable to choose the optimum representation of the signal according to some standard.

2. Literature Survey

The literature review attempts to discuss the work carried out in the areas of estimated multiplier circuit design. The methods attempted could be classified as functional behavioral or timing changes. Methods like extreme voltage scaling can be used for timing optimization [1]. Approximations can incorporate in to the significant structure squares (for example number juggling circuits) which are utilized in circuit, or could be brought into Boolean or elevated levels of depictions of discretionary circuit [2]. Surmised number juggling centers around planning fundamental number-crunching units, for example, estimated multipliers and adders that could substitute or expand careful number-crunching units of programmable processors (like GPUs and CPUs) or in custom circuit (for example ASICs and quickening agents). As a rule, it is normal that rough number juggling will convey higher force investment funds in customized circuits contrasted with CPU, where information move and directions of control structure the majority of calculations. Energy consumed can be reduced by operating at low nominal voltage but at the expense of mistakes induced by timing. The errors are not strictly bound and hence integration of additional circuit for rewarding mistake is needed. As the timing mistakes are activated by long transport lines that affect the largest portion of finished design, by modifying traditional multiplier catering for the smooth degradation should be able to quantify these effects of timing breach. Secondly, to facilitate accurate execution and compatibility of Boolean functions require modifications to the features dealing with techniques of logical extraction techniques. To achieve energy efficiency and higher speed, minimization of silicon region used and optimization of device parameters is required. The major characteristics and the limits of works in regarding estimated multipliers is summarized, the multiplier conditions, for example, will enable the less important parts to be removed. More reduction in energy can be obtained by removing more pillars, but at the expense of increase in mistakes. One of the other methods for better energy efficiency is by modular re-designing of combinational logic circuit with low complexity [4]. Multipliers could be constructed with tiny estimated multipliers for energy efficiency, but at the cost of mistakes spreading as the magnitude of the multiplier grows. This is achieved by structuring tiny sets in a hierarchical model. The paper [5] suggests the technique of software-based perforation technique. It suggests acquiring the optimum number of partial products by trading off precision in area and power. There are many area and power efficient methods suggested by modifying the Functional behavior ranging from transistor level to architectural level. The automated architecture approaches discussed in [6] uses current layout stream to synthesize the estimated circuits by using the system operation models and applying performance boundaries. The DWT needs many mathematical functions to be performed for its execution. As the numbers of calculations are more, it is required to know the hardware necessities for such compound calculations. Generally there are 2methods for performing DWT. One method is Convolution based method where filter bank structures are implemented. The second one is Lifting scheme which decreases the number of computations and avoids the use of multiple operations. The benefit of DWPT is that it is conceivable to choose the optimum representation of the signal according to some standard.
The representation that uses less coefficients is the best in image compression. Canonic Signed Digit (CSD) representation had the competence to do multiplications with less number of adders. It is different representation of a binary number with less number of 1’s and -1’s digits. One important applications of CSD is in multiplication operation, where it takes less number of subtractions & additions to produce the product. V.Gupta et. al (2013), [8] proposed logic complexity at transistor level. K.-J in 2004 gave an error correction method for modified booth fixed width multiplier [9]. H.R Mahdiani in 2010 presented a new method of bio-inspired approximate hardware arithmetic with improvements in cost and performance. H.A.Moghadam et al, in 2015 [10] proposed an approximate multiplier which can trade off accuracy and energy at design time for DSP & recognition applications. On the basis of survey of approximate computation, proposed approximate SAR unit for DWPT using CSD representation.

3. Proposed System

An approximate adder is developed in previous only with Luminance (Luma) concept. AM1 and AM2 are described for error recovery, which error has hand over from approximate adder and then multiplexing. In proposed system, to avoid the error correcting block, by applying bit weight based compression technique we designed a multiplier By using this method we reduce the area of the unsigned multiplier circuits with better accuracies. In the Partial product accumulation stage we can add the partial products by a Dual quality adder thus we can have a choice to select the output with 2 combinations.

As all the bits of a number does not have equal importance especially in applications which can tolerate this error, Our proposed estimated multiplier exactly uses the above concept i.e the way which allows dissimilar significance of bits of the number. Accordingly, the proposal is to restrain the quantity of bits applied via cautiously choosing a scope of bits for every one of the two multiplier operands. Subsequently, in equipment cost, this methodology lessens a huge multiplier to a significant smaller center multiplier and few controlling rationale that is required to recognize and in course they choose scope of bits of both of the operands to littler center multiplier.
A new algorithm for multiplication can be shown by an illustration in Fig. 2. First, the operands of the input are divided into a multiplication component comprising several higher significant bits and a non-multiplication part containing the lower significant bits. No need to be equal in length of each element. The process of multiplication occurs at the point where the bits are separated and move concurrently in the two opposite directions. In the case of Fig.2, the two 12-bit input operands, the “101110011011” (2971) multiplicand and the “010011 00 1001” (1225) multiplier, are splitted into two equal sections each consists of 6 bits. The lower order bits (non multiplication part), here a special procedure is intended for not producing partial product and removes the carry propagation path. Every bit location of the non-multiplication component from left to right is checked and if one or both of the two operand bits are “1,” the testing method is terminated and from that bit location all the bits are placed as “1.” As normal, the operation is performed from (LSB to MSB) right to left as the higher significant bits of an operands at the input which begins the multiplication portion. Hence, the circuit is designed in a conventional manner. Here, we maintained the conventional topology because the MSB’s are weighted more than the LSB’s. These two operations perform simultaneously, multiplying the MSBs, the power consumption and overall delay is reduced.

To Reduce the Delay of the Existing Circuitry we implemented another method where we have less area and delay compared to existing technique of approximation. And is done in two stages called partial product generation with Reduction and Final summation. The partial products generated are reduced by OR gate logic Clustering and in the second stage the reduced partial products are to be added in hybrid logic where right or least significant half is fully approximated part and the left part is added with accurate adder thus we can have better accuracy of the output.
\[
\text{SUM} = P \oplus Q \oplus R \oplus S \oplus C_{in} \\
C_{OUT} = (P \oplus Q) C_{in} + (P \oplus Q)^{1} P
\]

Compressors are the most dominant building blocks for a rapid multiplier. It provides an improvement of partial product addition at the outlay of smallest power dissipation. Instead of fully summoning partial products with the support of the CSA / Ripple adder tree, a compressor structure can accomplish the same function in much less period and at the same moment remove the problems of upturn of area high power consumption. This addition of partial products can not compensate as much for the reduction of delay allied with the critical path in conventional method comprises of half adders and full adders than counters or compressors.

\[
C_{CARRY} = (P \oplus Q \oplus R \oplus S) C_{in} + (P \oplus Q \oplus R \oplus S)^{1} S
\]

The purpose for compressor's apparent preference over counters is the advantages it provides in terms of total number of transistors used, power and delay allied with the critical path (mainly includes XORs). Both MUXs and XORs can be incorporated in the compressor configuration. The Compressors which reduce stages of the n product. A 4:2 compressor generates in a similar way and a carry for the next stage. Moreover, the carry in (Cin) of the new compressor produce (Cout). A 4:2 precise compressor consists two complete adder circuits. That describes a compressor configuration that uses two 2:1 multiplexers and three XOR-XNOR gates. The compressor logic equation outputs are as follows

4. Results and Discussion

The total designs are done and implemented in Xilinx ISE 14.7 with Verilog HDL coding. The hybrid logic used in the proposed multiplier method where the partial product accumulation is done using kogge-stone adder in the second stage of parallel prefix adder, which helps to error correction

Fig. 4 Simulation results of 16 bit multiplier in Xilinx software

Approximate multiplier the generation and reduction of error uses individual adders which increases delay as well as area. In the existing approximate multiplier in order to obtain the area and delay the accuracy is sacrificed. The Approximate multiplier when simulated for 16 bits it exhibits the accuracy comparably low, so in order to obtain better accuracy in proposed system, we avoid the error correcting block, by applying bit weight based compression technique.
Here, we are using 16-bit multiplier for foreground separation based on their threshold value which is related to that gray image. The results for the image processing can be done through Xilinx system generator and results are shown in below figures. The PSNR value of the image multiplication using the proposed method is obtained as 33.98.

Table 1 comparative Power, Delay, Area results for Wallace tree multiplier and Combinational multiplier

| Multipliers         | Power (mw) | Delay (ns) | Area  |
|---------------------|------------|------------|-------|
| Wallace tree multiplier | 40%        | 48%        | 46%   |
| Approximate multiplier | 26%        | 44%        | 28%   |

Fig.6 Comparison Analysis of Wallace Tree multiplier and Combination multiplier for Image Compression
5. Conclusion

Proposed High Speed and Low Power Error Recovery Approximate Multiplier designed executed using simulation software. Inexact unsigned multipliers for both error and circuit parameters are assessed as comparative. Among the inexact multipliers considered, truncation is an efficient way to reduce the complexity of the circuit. The design can be scaled to enable the users to trade accuracy and power as desired. Reduction of errors due to unbiased circuit’s usage can be used for computations within applications. Xilinx Tool were used to analyzed multipliers error and Timing Area & power consumption characteristics. Our multiplier design also was evaluated the design against an image processing applications. Verilog code for approximate multiplier integrated into it. The simulations were performed using Xilinx and ModelSim tools, and synthesis performed using Xilinx tools. It was found that application with Approximate multiplier fared comparatively with an inbuilt multiplier. The power consumed by the multiplier was found to be 5.91mW and the delay was found to be 6.54ns. Further this multiplier design can also be incorporated in typical DSP applications.

References

1. Noor Mahammad sk, Mohamed Asan Basiri m, “An efficient VLSI architecture for lifting based 1D/2D DWT”, Microprocessor and Microsystems(2016).
2. Dr.T.Srinivasulu, G.KiranMaye, “A survey on VLSI Architectures for Wavelets”,IJERA,Vol 7,Issue 10,Oct 2017.
3. Sujatha, Pushpawati Changlekar, P.Anita, “ Implementation of Binary Canonic signed digit multiplier using ASIC”,IJERA,Vol 3,issue 1,2013.
4. D. Mohapatra, V. Gupta, Raghunathan, & K. Roy, “Low-power digital signal processing using approximate adders,” IEEE Trans. Computer-Aided Design Integration Circuits Systems, vol. 32, no. 1, pp. 124–137,Jan. 2013.
5. K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, “Design of low-error fixed-width modified booth multiplier,” IEEE Trans.VLSI Syst., vol. 12, no. 5, pp. 522–531,May 2004.
6. H. R. Mahdiani, Fakhrain A. Ahmadi, S. M., and C. Lucas, “Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” IEEE Trans. Circuits Syst. I, Reg. Papers,vol. 57, no. 4, pp. 850–862, Apr. 2010
7. S. Narayanaanmorthy, Z. Liu H. A. Moghaddam, T. Park, and N. S. Kim, “Energy-efficient approximate multiplication for digital signal processing and classification applications,” IEEE Trans.(VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015
8. A. Momeni, J. Han, P. Montuschi, and F. Lombardi, “Design and analysis of approximate compressors for multiplication,” IEEE Trans.Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.
9. J. Liang, J. Han, and F. Lombardi, “New metrics for the reliability of approximate and probabilistic adders,” IEEE Trans. Comput., vol. 63, no. 9, pp. 1760–1771, Sep. 2013.
10. K.-J. Cho, K.-C. Lee, J.-G. Chung, & K. K. Parhi, “Design of low-error fixed-width modified booth multiplier,” IEEE Trans VLSI Syst., vol. 12, no. 5, pp. 522–531,May 2004