JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

IMPLEMENTATION OF COMPRESSOR BASED MULTIPLIER FOR HIGH SPEED DSP APPLICATIONS

T. Rikhitha
PG Scholar
Narayana Engineering college,Nellore,A.P

ABSTRACT

Compression of data is a common practice in the signal - processing & digital image analysis fields, and is often employed for multimedia & image processing purposes. Approximate computation is a prominent design approach in arithmetic. New high-speed areas may be opened up by high-speed multimedia applications. Error-tolerant circuits that use approximate computations. At the same time, these applications give great production at a reduced cost of accuracy in addition, the system's complexity is reduced as a result of their implementations. Power consumption and latency in the system's architecture are the main factors. It is proposed that two compressors be designed and analyzed that have smaller size, less delay, and more power than the present systems, allwhile maintaining precision that is equivalent to the current systems.

All designs were tested and forecasted on Area, Delay, Power (PDP), Margin of Error (ER), Range of Error (ED), & Accurate Output Count (AOC) before being implementedin 45 nm CMOS technology (the AOC). In comparison to an exact 4:2 compressor, the suggested 4:2 compressor with approximation has an overall decrease of 56.80 percent and 57.20 percent Power and delayed reduction of 73.30 percent Dadda multipliers of 8 × 8 and 16 x 16 are used in the suggested compressors. There's a comparable level of precision in these multipliers compared to current technology. Image smoothing & propagation are two examples of error-tolerant applications that will benefit from the architecture now under consideration.

I.INTRODUCTION

Exact computing units aren't necessarily essential in applications like data mining and multimedia signal processing. They may be substituted with a similar item. Research into error-tolerant applications using approximation computation is on the increase. These applications rely heavily on adders and multipliers. In digital signal processing, approximate complete adders are suggested at the transistor level. Partial product accumulation in multipliers is handled by their suggested full-adder design.

The use of truncation in fixed-width multiplication designs is common to simplify the circuitry. In order to compensate again for quantization error induced by the reduced portion, a variable correction component is added. Accumulation of bits is critical in termspf power usage when using approximation methods in multipliers. If the least relevant bits of the inputs can be trimmed, then partial products may be formed in order to decrease hardware complexity. In partial product accumulations, the suggested multiplier saves just a few adder circuits. A partial product reduction trees of four 8 x 8Dadda multiplier variations is given and applied with two types of roughly 4-2 compressors.

Mean Relative Error (MRE) is greatly affected by the suggested compressors in that they produce nonzero output for zero-valued inputs, which is a severe negative. In this brief, an estimated design is presented to address the current issue. As a result, accuracy is improved. When a Segment Multiplier (SSM) is used, the leading 1 bit of each operand is used to generate an output of m-bit segments. Then, instead of n multiplication, mm multiplication is used. A Partial Product Perforation (PPP) multiplier in an n-bit multiplieromits subsequent partial products beginning at position j, where j [0, n-1] and k [1, min (n-j, n-1)]. Multiplying an element in the Karnaugh map by 2 is used as a building block to produce 4 4 and 8 8 multiplications. Power-efficient Wallace tree multipliers might benefit from an inaccurate counter design.

The multiplier partial product accumulation ishandled by a new approximation adder. Compared to an accurate multiplier, a 16-bit approximation multiplier achieves a 26% decrease in power.Voltage Over-Scaling (VOS) is used to approximate an 8-bit Wallace tree multiplier, Errors may be caused by lowering the supply voltage, which produces routes that do notmatch delay restrictions. In the past, logic complexity reduction has been achieved by simply applying approximation adders and compressor to the partial products. Various probabilities are included into the partial products in this short. Systematic approximation is used to examine the likelihood statistics of the changing partial products.

Approximation is suggested using half-adder, full-adder, and 4-2 compressors. Thecomplexity of the arithmetic units has been lowered, but the error value has also been taken into consideration. Systemic approximation improves accuracy, while lower logic architecture of approximation arithmetic units reduces power and area consumption. Image processing applications benefit from improved Peak Signal-to-Noise Ratio (PSNR) values achieved by the suggested multipliers over previous designs. The arithmetic separation between a proper output and an approximation output for just a given input may be Characterized as the Error Distance (ED). In order to assess and normalize approximation adders, it is argued that ED (NED) is a virtually invariant metric regardless of the approximate circuit's dimensions. For both current and proposed multiplier designs, the conventional error analysis, MRE, is discovered .

II.LITERATURE SURVEY

- **Approximate Adders for approximate multiplication**

Document It's becoming more difficult for CMOS technology to keep up with the demands of future applications. This gap may be narrowed greatly with the help of numerous viable design methods. Accurate computation is one of them, although it has received the most attention in recent years. Accuracy is sacrificed for speed and energy efficiency in approximation computing, which harnesses the inherent fault of applications and provides highly energetic hardware and software implementations (e.g., performanceand energy). There has been a lot of study on approximate computing over the last decade, but much of it has focused on adders, which are hardware abstractions. There is a comparative study of the current state of the art in approximation adders. Both traditional design measurements and approximation computing metrics are available for comparison.

Approximate Compressors for Multiplication

At the nanometer scale, approximate computing is a promising paradigm for digital processing. For computer arithmetic designs, inexact computing is especially intriguing. In order to be used in a multiplier, the authors discuss the study and construction of two novel compressors with an approximate 4-2 compression ratio. In order to fulfil circuit- based figures of merit, these designs depend on various compression characteristics that allow imprecision in calculation (measured by error rate and so-called normalized error distance) to meet (number of transistors, delay and power consumption). A Dadda multiplier may be multiplied using four distinct techniques for using the suggested approximation compressors. There is substantial decrease in power consumption, delay and number of transistors compared to that of an exact design; two of the suggested multiplicand designs provide performance of system for image multiplication to respect to average normalized error distance as well as peak signal-to-noise ratio (and over 50dB again for considered image examples).

Approximate Wallace-Booth Multiplier

The potential benefits of improved performance and reduced power consumption offeredby approximate and in exact computing have lately gained a lot of attention. There are three parts to this approximation multiplier: a Booth encoder, a 4-2 compressor, and an approximation tree structure. For 8x8, 16x16, & 32x32-bit signed multiplication schemes, the approximate design is built and tested. This project presents and discusses simulationfindings for 45 nm technology. The suggested approximate multiplier produces considerable improvements in energy usage, latency, and combined metrics when compared to an accurate Wallace-Booth multiplier and other approximate multipliers available in the technical literature. These findings support the feasibility of the concept as presented.

- **Two variants of approximate multipliers**

For error-tolerant applications, approximate computing may reduce design complexity while increasing performance and power efficiency. A novel method for approximating multipliers is explored. Probability terms may be added by altering the multiplier's partial products. The chance of accumulating changed partial products has an effect on the logic complexity of approximation. Two different kinds of 16-bit multipliers make use of the suggested approximation. The results of the synthesis show that two suggested multiplierssave 72 percent and 38 percent of the power of an exact multiplier, respectively. When compared to other approximation multipliers, they are more precise. Image processing isused to test the suggested multipliers, and one model gets the best peak signal power of all of them.

Energy-constrained devices are increasingly being required to serve a wide range of Digital Signal Analysis (DSP) and classifying applications. Such applications often use fixed-point arithmetic to execute matrix multiplications while allowing for certain computational mistakes. As a result, multiplication efficiency must be improved. Finally, the exhibited computational mistake has no significant influence on DSP quality or classification accuracy.

III.EXISTING METHOD

Modified Booth encoding

Two signed binary values in 2's complement notation is multiplied using Booth'smultiplication method. In 1950, while working on diffraction at Birkbeck University in Bloomsbury, London, Andrew Donald Booth came up with the algorithm. Algorithm for shifting was developed by Booth, who utilized desk computers that were more efficient in shifting than at adding. In the field of computer architecture, Booth's method is of interest.

An implicit bit underneath the most significant bit, y1 = 0, is examined using Booth's procedure for the N-bit multiplication Y in sign two's complement form. Bits Yi and yi1are examined for every bit Yi, scale from zero to N - 1 for I. The product accumulation P remains unaffected if these two items are equivalent. The multiplicand times 2i is given to P when Yi is zero or one, and the multiply and times 2i is removed from P when Yi isone or zero. The signed product is P's ultimate value.

However, the representations of a multiplicand & product are not defined and may be anynumber system that allows the addition and subtraction of numbers; as is the case with the multiplier. The sequence of the stages is left to the reader's imagination, as indicated before. This is usually done by shifting the P accumulator rightward in small stages, beginning at I = 0, and then working its way up from there, starting with the lowest N bitsof P. Then, the multiplying by 2i is often substituted. In terms of these specifics, there areseveral modifications and optimizations available. Strings of 1s inside the multiplier are typically characterized as being converted to high-order+1 and low-order-1 at the endpoints of the string using this process. In this case, there is no elevated +1 and the total consequence is that the appropriate value is interpreted as negative.

IV.PROPOSED METHOD

Digital filters play a critical role in DSP. DSP's popularity is mostly due to its ability to perform at such a high level. Separation and restoration are the two primary functions ofa filter. When the signal is tainted by interference, noise, or other signals, signal separationis required. Consider, for example, an EKG gadget for monitoring a baby's cardiac activitywhile still in the womb. The mother's breathing and pulse will likely distort the raw signal. These signals must be filtered out so that they may be evaluated separately. When a signal has been distorted in some manner, signal restoration is utilized. It's possible to improvean audio recording that was recorded with low standards by filtering it. Using a lens thatis not properly focused or a camera that is shaken is another example. Analog or digital filters may be used to combat these issues. What's the best one? There is a wide dynamicrange in the amplitude and frequency response of analogue filters. In contrast, the performance levels that may be obtained with digital filters are much better. Filters that use digital technology can outperform analogue ones by a factor of a thousand times. Filtering issues may now be tackled in a whole new way thanks to this discovery. There are a number of restrictions with analogue filters, such as a lack of precision in resistors and capacitor stability. Digital filters, in contrast, are so excellent that the filter's performance is often neglected. The focus now moves to the signal's inherent limits and theoretical difficulties surrounding its manipulation. In many digital signal processing (DSP) techniques, a variable is multiplied by a set of known constant coefficients. Multiplication is the costliest operation in DSP algorithms when compared to other frequent operations such as addition, subtraction, delay elements, etc. It's a trade-off between the quantity of silicon in the integrated circuit and how quickly the calculation can be done. Due to the fact that each operation must be performed within the same period of time, multiplication uses more logic resources than most other operations, even if the same number of logic resources are available. Any time you want to multiply two differentvariables, you'll need a generic multiplier.

It is possible to use binary multiplication's features when multiplying by a known constant, such that we end up with a less costly logic circuit that is functionally similar tomerely asserting the constant on a general multiplier. Because multiplication is costly, employing a cheaper solution for merely multiplication might result in considerable savings when looking at the full logic circuit. According on the application, multiplication may be the primary operation. Throughout this thesis, we will suggest a number of algorithms that can be executed in software, but the solutions they provide allow for efficient hardware implementation of constant coefficient multiplication. Algorithms likethis look for suitable hardware realizations based on a set of constant coefficients.

It is suggested that the proposed & existing approximation multipliers have their design and performance metrics thoroughly analyzed. Image processing applications make use of the suggested multipliers, and the results are shown. Implementing multipliers involves three steps: creating partial products, creating a reduction tree from those partial products, and then combining those reduction tree sums and carry rows into a final result. The second phase is more demanding on the battery. The reduction tree step is whenapproximation is used in this short.

Implementation of Multiplier comprises three types:

- Generation of partial products
- Partial products reduction tree, and finally,
- A Vector merges Addition to produce final product from the sum and carry rowsgenerated from the reduction tree.

The second phase is more demanding on the battery. The reduction tree step is when approximation is used in this short. Using an 8-bit unsigned1 multiplier, the suggested approach for approximating multipliers is shown. Let _7m=0 and _7n=0 be two 8-bit unsigned input operands.

Multiple Constant Multiplication (MCM)

An extension of SCM is the problem of multiplying a variable x by several constantst$_1$, . . ., T$_n$ in parallel in a so-called multiplier block shown in Figure 4.5. Since intermediate results of the constant decompositions may be shared, a multiple constant multiplier block may be decomposed into fewer operations than the sum of the single constant decompositions' operation counts. The problem of finding the decompositionwith the fewest operations is known as Multiple Constant Multiplication (MCM). The potential savings from sharing intermediate results increase with the number of constants.The MCM problem is particularly relevant for the multiplier less implementation of digital Finite Impulse Response (FIR) filters [Bull and Hordocks 1991], but also for matrix-vector products with a fixed matrix, which includes linear signal trans-forms [Urschel et al. 2004; Chen et al. 2002; Liang and Tran 2001], such as the discrete Fouriertransform or the discrete cosine transform. In an n-tap FIR filter, every input sample is multiplied by all n taps. Discrete Fourier and trigonometric transform algorithms, on theother hand, involve 2x2 rotations, which require the simultaneous multiplication by two constants.

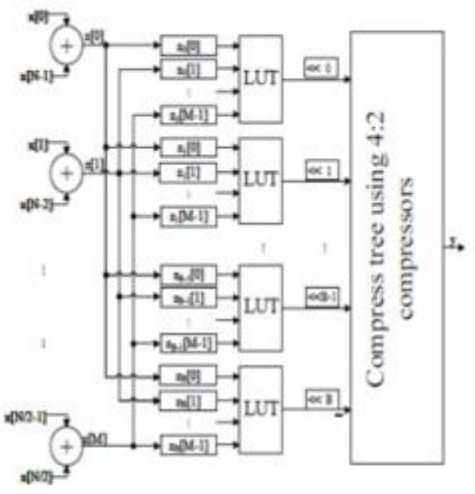


Fig. 1: Proposed block diagram.

Figure 4.5 is an example of a multiplier block, which implements the parallel multi- plication by 23 and 81 using only 3 add/subtract operations, although the separate optimal decompositions of 23 and 81 each require 2 add/subtract operations.

The different problem of multiplexed multiple constant multiplication was considered in [Tummel shammer et al. 2004]. In this case, the multiplier block contains multiplexers that are switched by control logic to achieve multiplication by different constants. This way sequential multipliers can be fused.

The MCM algorithms can be divided into four general classes:

- Digit-based recoding;
- Common sub-expression elimination (CSE) algorithms;
- Graph-based algorithms;

Digit-based recoding includes simple methods like CSD and the binary method mentioned in Section 4.1. They generate the decomposition directly from the digit representation of the constant. These methods are the fastest and the worst-performing; however, a more recent approach [Coleman 2001] uses different number systems to yield considerably better solutions. The main advantage of digit-based recoding is their low computational cost, typically linear in the number of bits. As a consequence, these methods can be easily applied to constants with thousands of bits.

Common sub-expression elimination (CSE) algorithms are direct descendants of digit- based recoding methods. The basic idea is to find common sub patterns in the representations of the constants after the constants are converted to a convenient numbersystem such as CSD. Examples for this method include [Pasco et al. 1999; LeFevre 2001; Hartley 1996]. The disadvantage, however, is that the performance of these algorithms depends on the number representation. Further, even though the considered CSE problem is NP-complete [Gary and Johnson 1979; Downey et al. 1980], its optimal solution doesin general not provide the optimal MCM solution. More recently, [Dempster and Macleod2004] proposes searching over alternative number representations to find considerably improved solutions using a CSE algorithm.

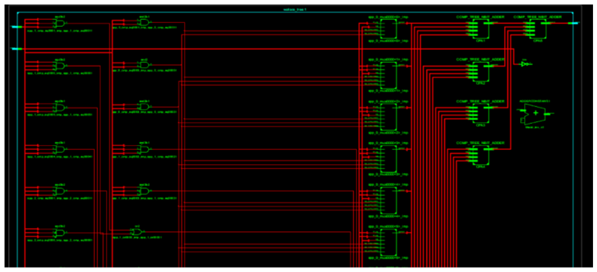
Graph-based algorithms are bottom-up methods that iteratively construct the graph (as in Fig.4.5) representing the multiplier block. The graph construction is guided by a heuristic that determines the next graph vertex to add to the graph. Graph-based algorithms offer more degrees of freedom by not being restricted to a particular representation of the coefficients, or a predefined graph topology (as in digit-based algorithms), and typically produce solutions with the lowest number of operations. Examples of graph-based algorithms include [Bull and Hordocks 1991], RAG-n [Dempster and Macleod 1995], and [Bernstein 1986]. This project proposes a new graph-based algorithm.

Today, to the best of our knowledge, RAG-n yields the solutions with the smallest number of add/subtract operations among all algorithms. Graph-based algorithms are expected tooutperform other methods since they have fewest restrictions. How-ever, RAG-n relies on a lookup table of optimal single constant decompositions, which is currently limited to 19 bits as mentioned before.

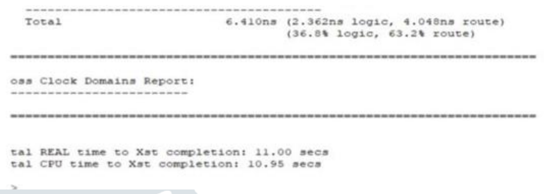
This project first presents a general formal frame-work that captures the common structure of graph-based MCM algorithms. A crucial component in this framework is ournotion of "A-distance," an extension of the concept of adder distance introduced in [Dempster and Macleod 1995], and its exact or heuristic estimation. We use the framework to develop a new graph-based MCM algorithm that outperforms the best available algorithms with respect to the number of add/subtract operations in the obtained multiplier blocks. In particular, we achieve an up to 20% lower average operation count than the best previous algorithm RAG-n. At the same time, our new algorithm is not bit width limited like RAG-n, and can thus be used to generate multiplier blocks for all practically relevant bit widths. Finally, we perform a detailed runtime analysis of our new Algorithm and other graph-based algorithms used for benchmarks. This analysis was notprovided in the original project. Besides reducing the number of add/subtract operations,it is often desirable to optimize for other metrics, for example, the critical path of the MCM block, or the register pressure in the generated code. Examples of such work include [Dempster et al. 2002] and [Kang et al. 2001]. This project does not consider thistype of optimization; however, the structure of our algorithm enables its adaptation to other target metrics.

V.SIMULATION RESULTS

Proposed Internal Block Diagram



Delay



```
   -----------------------------------------
   Total                        6.410ns (2.362ns logic, 4.048ns route)
                                        (36.8% logic, 63.2% route)

=================================================================

oss Clock Domains Report:
------------------------

=================================================================


tal REAL time to Xst completion: 11.00 secs
tal CPU time to Xst completion: 10.95 secs

>
```

Simulation

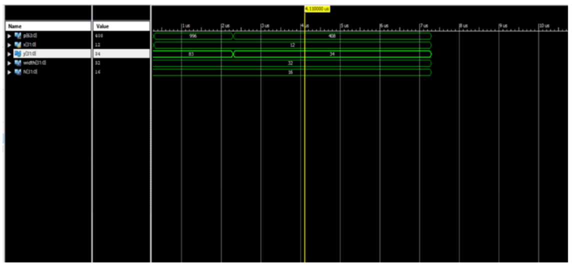


Table 1: Comparison Between Existing &Proposed level compressor

Parameter	Existing Level multiplier	Proposed Level multiplier
Delay	23.31 ns	6.410 ns

VI.CONCLUSION

Using signals generated and propagated, this project proposes efficient approximationmultipliers. A basic OR gate is used to create changed partial products for approximation. Half-adder, full-adder, and 4-2 compressors are all recommended to minimize the leftover partial products in the final product. In Multiplier1, approximations are applied to all n bits, but in Multiplier2, they are applied just to the n 1 least significant bit. The space and power consumption of Multiplier1 and Multiplier2 are significantly reduced compared to exact designs. Both Multiplier1 and

Multiplier2 save 87 and 58 percent, respectively, from accurate multipliers in APP compared to previous approximation solutions. When compared to previous approximation multiplier designs, they are proven to be more precise. Using the suggested multiplier designs, output quality may be maintained while substantial power and space are saved. the delay is reduced in proposed Multiplier compared to existing Multiplier.

VII.FUTURE SCOPE

We also extend this project as distribution Arithmetic based FIR filter can be implemented further.

REFERENCES

[1] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Computer-Aided Design Integer. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[2] E. J. King and E. E. Swartz lander, Jr., "Data-dependent truncation scheme for parallel multipliers," in Proc. 31st Asilomar Conf. Signals, Circuits Syst., Nov. 1998, pp. 1178–1182.

[3] K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Pari, "Design low-error fixed-width modified booth multiplier," IEEE Trans. Very Large-Scale Integer. (VLSI) Syst., vol. 12, no.5, pp. 522–531, May 2004.

[4] H. R. Mamdani, A. Ahmadi, S. M. Fakhri, 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.

[5] A. Momani, J. Han, P. Montecchio, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Transcom., vol. 64, no. 4, pp. 984–994,Apr. 2015.

[6] S. Narayana Moorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy- efficient approximate multiplication for digital signal processing and classification applications," IEEE Trans. Very Large-Scale Integer. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[7] G. Gervais, K. Tsunamis, S. Xydis, D. Souris, and K. Pelmets, "Design-efficient approximate multiplication circuits through partial product perforation," IEEE Trans. Very Large-Scale Integer. (VLSI) Syst., vol. 24, no. 10, pp. 3105–3117, Oct. 2016.

[8] P. Kulkarni, P. Gupta, and M. D. Sreckovic, "Trading accuracy for power in a multiplier architecture," J. Low Power Electron., vol. 7, no. 4, pp. 490–501, 2011.

[9] C.-H. Lin and C. Lin, "High accuracy approximate multiplier with error correction," in Proc. IEEE 31st Int. Conf. Compute. Design, Sep. 2013, pp. 33–38.

[10] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in Pocono. Exhibit. (DATE), 2014, pp. 1–4.

[11] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "Accommodating and analysis of circuits for approximate computing," in Proc. IEEE/ACM Int. Conf. Compute. - Aided Design (ICCAD), Oct. 2011, pp. 667–673.

[12] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Compute., vol. 63, no. 9, pp. 1760–1771, Sep. 2013.

[13] S. Suma net al., "Image enhancement using geometric mean filter and gamma correctionfor WCE images," in Proc. 21st Int. Conf., Neur