# LOW-POWER, HIGH-AREA APPROXIMATION MULTIPLIERS: A DEVELOPMENT PERSPECTIVE

[#1]**BOORLA SANTHOSH,** ***Associate Professor,***

***Department of Electronics and Communications Engineering,***

[#2]**MADIREDDY SANDHYA RANI,** ***Assistant Professor,***

***Department of Electronics and Communications Engineering,***

**MOTHER THERESA COLLEGE OF ENGINEERING AND TECHNOLOGY, PEDDAPALLY, TS.**

**Abstract:** There is a requirement for data compression in several fields, including signal processing, digital image analysis, multimedia, and image processing. The use of approximation computation is widespread in mathematics. A potential catalyst for the introduction of additional high-speed areas is the development of multimedia programs capable of transmitting and receiving massive volumes of data quickly. Due to their robust nature, these circuits can easily recover from malfunctions. They are also renowned for their impeccable numerical precision. Improved system performance is attributed to the application. Eliminating lag time and power consumption is a priority. The two compressors we provide are more compact, faster, more powerful than our own, and they achieve the same level of precision. All designs were thoroughly analyzed and forecasted with regards to AOC, area, delay, power (PDP), margin of error (ER), range of error (ED), and CMOS technology at 45 nm. Precision compressors save 57.20% on energy and 56.80% on time as compared to a 4:2 compressor. Compressors often use 8- and 16-dada multipliers. In terms of precision, boosters are on par with cutting-edge tools. The proposed framework will be compatible with error-handling techniques, allowing for the application of features like image improvement and transmission.

***Keywords*:** **Signal** – processing, Digital image analysis, Multimedia, Error tolerant, AOC, CMOS.

## 1. INTRODUCTION

For tasks like data mining and multimedia signal processing, less precise techniques can be applied. They can be substituted for the same items. Mathematical approximation and working with errors are two areas that are getting a lot more attention in the realm of study. More and more individuals are utilizing these apps, and they continue to expand in size. It is possible to alter digital signals with a simple transistor-based adder. The incomplete output of a multiplier could be added to with a full adder. The complexity of fixed-width multiplication circuits can often be reduced by employing truncation. A variable correction component corrects any errors introduced by quantization as soon as the size is shrunk.

Power consumption can be reduced by employing approximate multipliers during bit capture. Partial outputs can be generated with less complicated circuitry by eliminating the less crucial inputs. Multiple adder circuits are required to accommodate partial products. The unfinished product reduction tree reveals that there are a total of four Dadda 8 x 8 multipliers and two Dadda 4 x 2 compressors. The proposed compressors are harmful to MRE because they generate non-zero outputs when given inputs of zero. The focus of this analysis shifts to an alternative scenario. Accuracy improves as a result. The first bit of each input is used by Sequential Shift Machines (SSMs) to generate m-bit output segments. It follows that replacing n by n with m x m is equivalent to n x n. Partial products that begin with j can be discarded with the help of an n-bit multiplier. In this scenario, j can range from 0 to n-1, and k from 1 to the smaller of (n-j) and (n-1). Each element of a Karnaugh map can be multiplied by two to provide a number divisible by both 4 and 8. Low energy consumption on the part of Wallace tree elements means that even modest preventative interventions can provide positive outcomes. One type of adder regulates how several subproducts are combined. A precise multiplier requires 26% more power than an approximate multiplier.

An 8-bit Wallace tree multiplier can be emulated on a computer by a technique called voltage overscaling (VOS). It can cause issues if routes have to wait longer than expected because of a power outage. It has been done before, using techniques like approximation adders and partial product compression, to simplify logic. The probability presented here are illogical. The systematic approximation is used to calculate the probability statistics of partial products.

You can make educated guesses by employing full-adders, half-adders, or 4-2 compressors. Once the error was corrected, the arithmetic optimization process continued. Approximation arithmetic units are more efficient and need less space and power because of their simpler logic. By carefully guessing, they improve their precision. In comparison to traditional PSNR techniques, the proposed multipliers provide superior image processing results. The ED measures how far off an approximation of an input is from the true value of that input.

Because it is relatively invariant with respect to circuit size, ED (NED) is useful for verifying and tuning approximation adders. Both the existing multiplier layout and the proposed one make use of MRE (the old-fashioned kind of error analysis). In the next paragraphs, you will find an in-depth explanation of the report's structure. The second part of the paper examines the proposed architecture, while the third part compares and contrasts the proposed multiplier with current approximation multipliers in terms of design and error metrics, and the fourth part discusses the application of the multipliers in image processing software and draws conclusions.

## 2. LITERATURE SURVEY

### Approximate Adders for approximate multiplication

It is becoming increasingly difficult to foresee the requirements of future CMOS applications. The gap might be narrowed considerably by employing some cutting-edge design methods. There has been a recent uptick in the value of precise analysis. Approximate computing is optimal because it makes advantage of low-power methods. Bugs in the program are used for this purpose. There has been a lot of progress made in the field of approximation computation over the previous decade. Nonetheless, most of this study has been on adders, which are not concrete entities but rather representations of them. Discussed here are the merits and drawbacks of today's estimation enhancers. Measuring how well a design has been analyzed and estimated by a computer is possible.

### Approximate Wallace-Booth Multiplier

Approximation computation allows for more effective computer processing at the nanoscale. It's fascinating to see what happens when computers make mistakes in math. Two novel compressors are described in the study, each with a compression ratio of roughly 4:1. Error rates and normalized error distances are two examples of computational imprecision that these systems may be able to tolerate because of reduced transistor counts, reduced delay, and reduced power consumption. A Dadda multiplier can be expanded by a factor of four with the use of compressors. In the field of image processing, approximation multipliers have been proved to perform well in several simulations. Multiplication of images using two multiplicand designs performs admirably when the mean squared error and peak signal-to-noise ratio of the sample images are both more than 50dB.

### Two variants of approximate multipliers

As we've established, simple calculations that can be applied in a variety of contexts can increase productivity and decrease energy consumption. This multiplier has a 4-2 compressor, Booth encoding, and approximation tree as standard equipment. signed multiplication strategies with 8, 16, or 32 bits can be evaluated and researched. This study provides and analyzes results from the 45 nm model. With respect to energy consumption, delay, and effectiveness, the Wallace-Booth approximation multiplier outperforms both the precise multiplier and other approximate multipliers mentioned in the recent literature. The results from this analysis support the merit of the proposed strategy.

## 3. EXISTING METHOD

Due to its ability to do additions without carrying and adaptability to a variety of needs, the redundant binary (RB) approach is ideally suited for fast multipliers. Two distinct techniques, Radix-4 Modified Booth Encoding (MBE) and RB, are used to generate Error Correction Words (ECWs). To accommodate the RB partial output, the RB multiplier has an extra row added. When RBPP is applied in the MBE multiplier's second step, performance improves. RBMPPG is a modified bits generator we developed to speed up the RBPP accumulation process. Restricted Boltzmann Machine Probabilistic Graphical Models (RBMPPG) are used in the design of multipliers, and they eliminate more incorrect product rows than RB MBE designs. This means that when each multiplier operand is at least 32 bits in length, these designs based on RBMPPG require less space and produce less heat than the present NB multiplier designs. There will likely be a 5% increase in latency, according to calculations. Alternating between several RB multipliers can reduce the power-delay product by as much as 59%.

Computers with low processing speeds are sufficient for data mining and multimedia signal processing. Concerning blunders, if you will. Products that are difficult to tell apart can be substituted for them. There is still a lot of work to be done on finding practical uses for approximation mathematics that are resilient to error. There is a continuing rise in the number of people who download and utilize these apps. It is possible to alter digital signals with a simple transistor-based adder. Full adder architectures manage the summation of multiplier partial products. As well as fixing multiplier problems, truncation is often used to simplify circuits.

To compensate for the quantization error introduced by the reduction in size, a variable correction term is introduced. Approximation approaches result in a linear increase in multiplier power consumption as the number of partial products added increases. Partial outputs can be generated with less complicated circuitry by eliminating the less crucial inputs. In earlier iterations, complete adders and compressors had their circuitry reduced for better performance. The probability presented here are illogical. The systematic approximation is used to calculate the probability statistics of partial products. Half-adding, full-adding, or compressing by a factor of 4-2 will get you near. There may be some errors still present, but they have been minimized as much as possible by not using unnecessary mathematical terminology. Approximation arithmetic units are more efficient and consume less power since they lack the logic of traditional arithmetic units. Careful estimation helps them get closer to the mark. In terms of PSNR, the proposed multipliers outperform the existing techniques. The error distance (ED) is the difference between the true answer and the estimated response. Normalized Euclidean Distance (NED) is one frequent metric used to evaluate approximation adders; it remains constant regardless of the passage of time. A mean-squared error (MRE) study is performed to evaluate the new multiplier design against the old. Technology doesn't need to be perfect to mine data or handle multimedia signals. Products that are difficult to tell apart can be substituted for them. Mathematical approximation and functions are still being studied by researchers for their potential to detect flaws in a variety of contexts. These programs continually evolve to become more powerful and feature-rich.

As a result, the RBR system wastes bits that could be utilized to indicate an additional digit in the binary representation. Binary integers can be expressed in a wide variety of ways due to this. Since each digit in an RBR takes up two bits, two's complement differs from RBRs in this fundamental way. Numerous advantages make RBRs a preferable alternative to classical binary forms. The requirement for a second firearm to be carried undercover is eliminated by using a red dot sight (RBR). When using the Rosberg representation, arithmetic operations speed up while bitwise operations take longer. The sign of the number represented by the digit may not always be the same as the sign of the digit. The individual digits that compose signed RBR numbers carry significant meaning. Using relative bearing references (RBRs) simplifies the creation of geographic coordinates. For an RBR digit, two bits are required because each bit represents a single digit. One way to understand the repetition of some digits is to use a translation table. Each bit pair in this table corresponds to a positive or negative integer value.

Integer values can be calculated from any representation by adding the digits and multiplying each digit by its weight, such as the common binary format. You can get twice as much for that home over there on the right. Most RBRs are capable of processing negative numbers. One binary digit cannot distinguish between positive and negative values when a number is recorded repeatedly. RBRs can use a variety of representations for numbers.

The "canonical" representation of integers is often contrasted with the non-adjacent form or two's complement.

## Modified Booth encoding

Booth's multiplication algorithm can be used to multiply together two binary values expressed in 2's complement notation. As he was studying diffraction at Birkbeck College in 1950, Drew Donald Booth came up with the idea. As a result of his work on a desktop program that improved the speed and efficiency of data transmission, Booth no longer needs to develop mobile applications. Booth's approach to incorporating PCs into the design process is novel.

In order to understand what y1 = 0 and its implied bit, which is the most significant bit, mean, we can use Booth's N-bit multiplication Y in sign two's complement. With i ranging from 0 to N-1, it checks each byte, yi, against its predecessor, yi1. Collection P won't be altered if the products in question are interchangeable. If yi is either 0 or 1, then increasing the multiplicand by a factor of 2i will result in a greater P. Alternatively, if yi is zero or one, then a 2i-multiplication of the multiplicand reduces P. For maximum profit, safeguard that which you value most.

Any calculator will display the multiplicand, product, and multiplier when performing an addition or subtraction. The order of the steps is up to the observer, as we have already mentioned. The P accumulator is incremented bit by bit with the N least significant bits of P when I equals zero via a right shift operation.

It is usual practice to transform P into a product of 2i. These particulars are amenable to a wide variety of adjustments.

Typically, at this point, the high-order value of the multiplier 1 strings is modified to +1 and the low-order value is modified to -1. No positive increment is possible, hence a negative integer must be subtracted instead.

**Step 1: Booth Encoder and Partial Product Generator stage (BEPPG stage):**

Booth encoders significantly impact a product's performance in a certain area. The number of bits that can be conserved has a direct bearing on the performance, efficiency, and power consumption of the RB summation tree or multiplier. Eighteen CRBBE-4 modules run the initial multiplier step. Five items now need to be completed. Using bit manipulation on the multiplicand, we generate sixteen columns of RBPPG partial products.

**Step 2: Redundant Binary Adder summing tree stage (RBA summing stage):**

The sum of all the separate products is 128 bits. These four bits are added together by the redundant binary adders 1, 2, 3, and 4. One hundred twenty-eight bit segments were generated by the RBA.

**Step 3: Redundant binary to NB conversion**

**stage(RB-to- NB stage):**

When translating, the whole number of RBs are converted to NBs. There is a distinct delay pattern revealed by the Rbs result, allowing for bulk adjustments to be made to the numbers based on their arrival times. As long as carries are calculated in the same way, the sum can predict the result of the next integers.

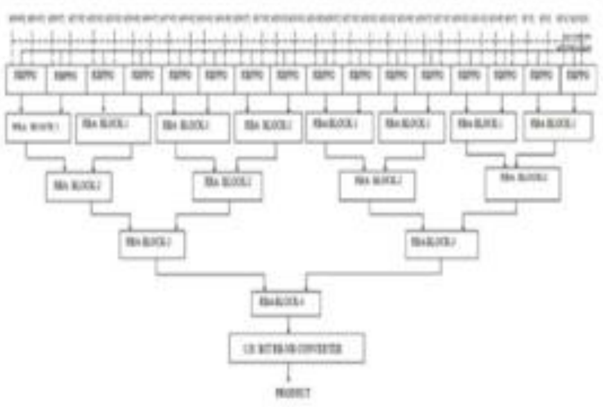


Fig 3.1: Block Diagram 8*8 bit High Performance Redundant Binary Multiplier

Binary multiplication is a mathematical operation that digital electronics like computers can perform between two binary integers. There are only binary adders in the system. A variety of digital repeater implementations exist in the field of computer science. Variables are often calculated and used in combination during the process. For the sake of making multiplication of whole numbers in base-10 easier, the way in which elementary school children are taught binary numbers should be altered.

Between the years 1947 and 1949, Arthur Alec Robin was an apprentice and then a research engineer at English Electric Ltd. As part of his dissertation work at Manchester, he was tasked with developing a hardware multiplier for the state-of-the-art Mark 1 computer. Until the late 1970s, most microcomputers couldn't perform multiplication. Computer programmers utilize something called a "multiplication algorithm" to reorganize and combine partial findings. As a common method, loop unwinding was frequently employed. The "multiply routines" performed by mainframe computers were actually merely shifts and additions disguised as "multiply instructions."

Original microprocessors lacked a multiplicative instruction. The MSC-51 series and the 6809 are examples of high-end eight-bit microprocessors from the 1980s. The two can rapidly reproduce themselves. Among the Atmel families, the ATMega, ATTiny, and ATXMega all use the cutting-edge AVR 8-bit processor.

More and more transistors were packed onto each board as integration progressed. A semiconductor may contain enough of these arithmetic adders to process all the intermediate values simultaneously. A decision was made to eliminate the necessity for individual adders in each CPU.

Early digital signal processors included a single multiply-accumulate unit that occupied the majority of the semiconductor and was utilized for quick multiplication.

## 4. PROPOSED METHOD

Important tools for digital signal processing (DSP) are digital filters. DSP's widespread acceptance stems from the fact that it provides desirable results. Separation and facilitation of recovery are the primary functions of filters. Irrelevant information and noise should be eliminated if the information is inaccurate or imprecise. To determine a young patient's heart rate, an electrocardiogram (ECG) can be performed. There is a risk that the signal will be garbled due to the mother's respiration and heart rate. It is impossible to investigate a hypothesis without first isolating its components. Whenever a signal isn't functioning properly or has been damaged, it must undergo repair so that it can function normally again. Sound quality is improved when a cassette with poor recording is filtered. You can employ a customized camera or a lens with a defocused focal plane. Such issues can be addressed by employing either analog or digital filters. Which approach, if any, yields superior outcomes? In terms of frequency range and amplitude, analog filters are extremely flexible. Digital filters, meanwhile, are vastly superior. There is no comparison between digital and analog filtering quality. Modern, state-of-the-art filtration algorithms can be credited to this breakthrough. Resistance and capacitance components in analog filters are notorious for being imperfect and shifting. The power of digital filters, however, is often overlooked. Disruptions to signals and how to fix them are common topics of conversation. Variables are often multiplied by fixed coefficients in DSP applications. The most time-consuming part of DSP algorithms is the process of introducing, erasing, and reintroducing delay components. There is a trade-off between the efficiency and size of an integrated circuit's silicon footprint. Despite the same number of operations, multiplication still takes more thought and reasoning than addition. When two variables of different dimensions are multiplied together, this mathematical function determines the result. It is possible to build a low-cost logic circuit that always returns the same value by using binary multiplication and a known multiplier. Due to the expensive nature of multiplication, a more efficient approach in the logic circuit could help bring down overall expenses. Sometimes, the primary operation will be a multiplication.

The focus of this research is on developing software that will simplify hardware-based coefficient multiplication by constants. With a known set of coefficients as a starting point, this method can be used to find practical hardware implementations.

Think about both the suggested multiplier and the one you're using right now to get an approximation of the true value. Images enhanced with multipliers in image-editing software are displayed. Construct a reduction tree from some partial products you've made. In order to multiply, you must join the nodes that represent the total and the carry. There is a lot of strain on the battery during the second phase. In this paragraph, it is used as a method of attack in the reduction tree.

There is a lot of strain on the battery during the second phase. In this paragraph, it is used as a method of attack in the reduction tree. To illustrate how to acquire the multiplier, an 8-bit unsigned multiplier is used in this example. Both _7m and _7n are unsigned 8-bit inputs, so let's assume that.

Similar to the word "am,n," "mnin" means "new." See the diagram in Figure 1. To get these findings, we simply do a logical AND operation on n and m bits. Booth multipliers and other variants of signed multiplication are incompatible with the proposed approximation method.

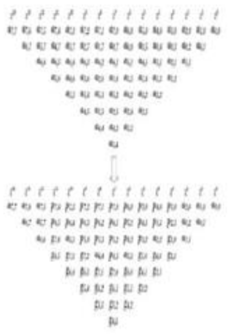


Figure 4.1: Transformation of generated partial products into altered partial Products.

It has been calculated that there is a one-in-four chance that the letters "am" and "n" will appear in the same sentence. When arranging the components am,n and am,m in vertical columns, more than three partial products can be generated for the purposes of sending and receiving messages. Modifications are made to both the outgoing and incoming signals of PM,n and gm,n. When the values in column 3 (originally weighted 23) are changed to those in column 11 (originally weighted 211), the am and a values are exchanged for pm and gm. One example of a grid segment displaying many goods is depicted in Figure 1. The modified and baseline partial product matrices are shown in Figure 1.

There is a one in a hundred chance of Gm happening, which is much lower than am's one in sixteen chance. It is eight times more likely that PM,n will be a modified partial product than gm,n. Reconstruct the partial product matrix with these new values for the variables.

Embedded system development has its greatest challenge in the effort to control time and power consumption. Embedded systems use digital signal processing to deal with images and movies, despite the fact that this places a heavy burden on the CPU. Due to time and resource constraints, hardware implementation of algorithms is typically the best option for embedded devices. Many methods, including image processing, linear transformations, digital filtering, and others, involve multiplying a single variable by a large number of fixed values. If these factors are improved, the plan's energy efficiency and geographical reach will be maximized. We call this process "multiple constant multiplication" (MCM).

The Minimum Cost Multicommodity (MCM) problem can be solved well with the use of graph dependency and CSE methods.

The difficulties of computing tix quickly when there are few variables are investigated.

While ti and x are held constant, the value of x can take on a wide range of numbers between 1 and n. In order to improve the efficiency of hardware filters and transformations for digital signal processing, it would be helpful to use MCM, which stands for "multiple constant multiplication," instead of more expensive multipliers. As an alternative to constant multiplication, software engineers may choose for continuous addition and shifting processes. Since integer multipliers have a lesser throughput than adders, certain embedded computers may not even have an adder, let alone a multiplier. This suggests that it could be important to find ways to make it more efficient. In the realm of computer science, a central mathematical problem is the Minimum Cost Matching problem (MCM).

A new approach to solving the MCM issue is presented. In terms of the amount of additions and subtractions required to find a solution, our method regularly exceeds all other published methods. New features have been built into the improved app. Further investigation into the topic will allow us to offer a more in-depth critique. With this, we can more precisely define our contribution and situate it amongst related research.

A. **Single Constant Multiplication (SCM)**

The product y = tx, where t is an integer or fixed-point constant, can be broken down by adding, deleting, and rearranging the bits.

I’d appreciate it if you could elaborate on your question or provide further background. Single constant multiplication (SCM) is shown to be NP-complete by Cappello and Steiglitz (1984). Choosing the decomposition that calls for the fewest operations is part of the problem. To keep things simple, let's pretend the numbers are integers and think of fixed-point multiplication as an integer multiplier followed by a right shift. Like the additive chain problem [Knuth, 1969], the SCM problem can be solved by simply multiplying the solution by a constant and then adding another constant. When people have the ability to adapt, the problem's core elements and the approaches used to address them undergo profound changes.

By exchanging the ones in the binary constant t for shifts and then adding the resulting products, division may be broken down. Calculating the product of three numbers yields the value 71: (6x)+(2x)+(1x) = 1000112x, where x is a positive integer. It is possible to reduce multiplication to a series of addition and subtraction steps. This is accomplished by moving the zeros and deducting the value 2n1 (the nearest constant made up entirely of ones) from the result.

Whether the circumstance is extremely negative or extremely positive, the answer requires 2b + O(1) additions and subtractions because of the combination of the two methods. Specifically, t's bitwidth is denoted by b. (x7 x)54x = 712 is a reduction of the equation 72x = 100012x.

Due to its capacity to represent negative numbers, addition and subtraction are typically performed with the canonical signed digit (CSD) representation of a number [Avizienis 1961]. It is possible to lower the final case's optimal number of phases from three to two.

The solution is 1001001CSDx, which is equal to 1000112 when x6 is added to x3 minus x1.

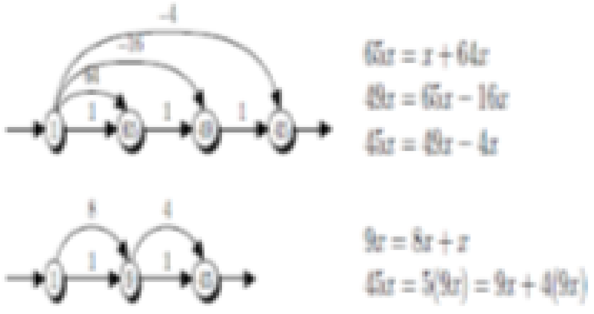
According to a 1999 research by Wu and Hasan, CSD can cost anywhere from 2b + O(1) in the worst case to 3b + O(1) under normal conditions.

The difficulty in evaluating if CSD achieves the ideal split of add/subtract operations stems from the lack of knowledge regarding the precise worst-case and average costs of CSD. In order to determine the most efficient ways to partition constants as large as 12 bits, Dempster and Macleod (1994) developed a thorough search procedure. They also showed how to solve 12-bit constants with shifts no larger than b + 1. Even though shifts can be restrictive, Gustafsson et al.

(2002) were able to successfully extend their analysis to include constants of up to 19 bits, leading to optimal results. It's a diagram, and it's shown in Figure 1. While the exponential worst-case cost of the intended decomposition is still being investigated, O(b) complexity or less is the goal in the best-case scenario. There are 300 randomly chosen constants with bitwidths ranging from 2 to 19, and the provided data contains the average number of additions and subtractions (y-axis) for these values. The goal of this information is to make comparisons (along the x-axis) among the three different decomposition methods.

The three-add-two-subtract (CSD) and two-add-two-subtract (optimal) techniques of 45-times multiplication are depicted in Figure 1. The edges depicting scaling factors (2-power) represent transformations, while the nodes representing sums and differences represent additions and powers of two, respectively. If the value on the scale is negative, subtraction must be done.

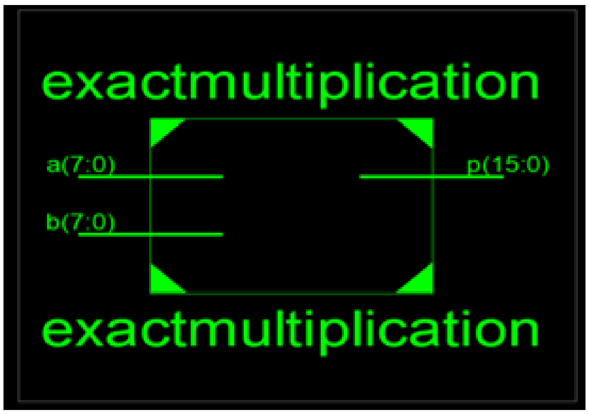
The optimal decomposition takes advantage of a different graph topology than the inefficient CSD decomposition does. Unsatisfactory results have been found when using CSD or other digit-based methodologies to evaluate one type of network architecture. To find the best ways to break down a network, exhaustive search methods, as described in [Dempster and Macleod 1994; Gustafsson et al. 2002], examine every conceivable configuration.



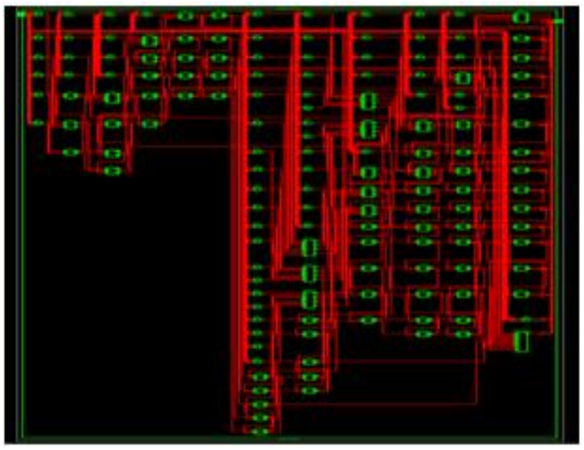
The CSD method (top) and the optimum method (bottom) for multiplying 45 are shown below. Edges labeled with a power of two signify scaling operations, while nodes labeled with outputs denote arithmetic operations. If the value on the scale is negative, subtraction must be done.

## 5. SIMULATION RESULTS

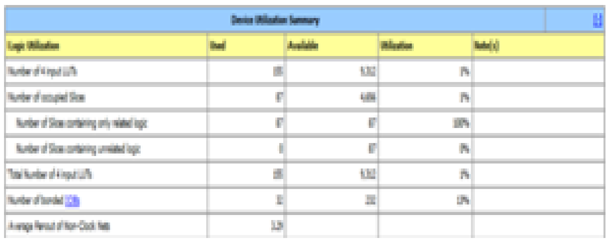
### RTL



### INTERNAL BLOCK DIAGRAM

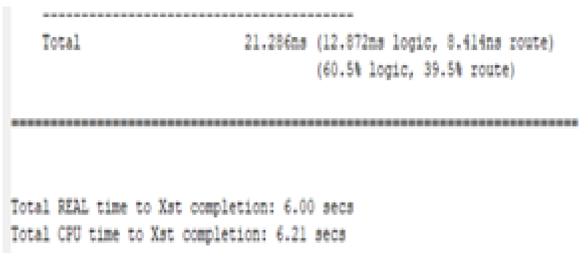


### Area



| Device Utilization Summary | | | | |
|---|---|---|---|---|
| Logic Utilization | Used | Available | Utilization | Note(s) |
| Number of 4 input LUTs | [illegible] | 9,312 | 1% | |
| Number of occupied Slices | 67 | 4,656 | 1% | |
| Number of Slices containing only related logic | 67 | 67 | 100% | |
| Number of Slices containing unrelated logic | 0 | 67 | 0% | |
| Total Number of 4 input LUTs | [illegible] | 9,312 | 1% | |
| Number of bonded IOBs | 32 | 232 | 13% | |
| Average Fanout of Non-Clock Nets | 3.29 | | | |

### Delay



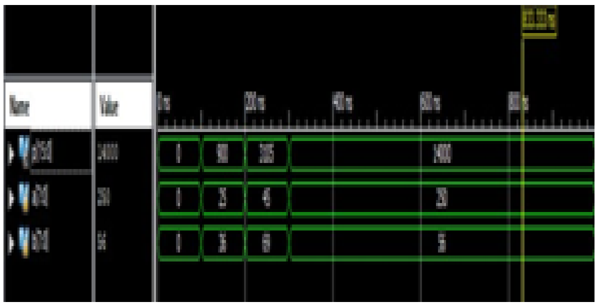
```
   ----------------------------------------
   Total                    21.286ns (12.872ns logic, 8.414ns route)
                                     (60.5% logic, 39.5% route)


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


Total REAL time to Xst completion: 6.00 secs
Total CPU time to Xst completion: 6.21 secs
```

### Power

## Simulation



## 6. CONCLUSION

This work introduces a method for generating and transmitting data simultaneously, which can be used to multiple estimates. Estimation of the changed partial products is performed using a standard OR gate. Half-adders, full-adders, and 4-2 compressors can be used to reduce the number of extra bits in the output respectively. The n-bit set is approximated by Multiplier1, whereas the least significant bit is approximated by Multiplier2. Compact and low-power, Multiplier1 and Multiplier2 outperform their higher-precision counterparts. Savings of 87% and 58%, respectively, are achieved by multipliers 1 and 2 as compared to earlier approximation methods for accurate multipliers in APP. In comparison to its forerunners, these approximation multiplier designs provide higher levels of accuracy. Energy and space efficiency are both maximized in the proposed multiplier designs without sacrificing output quality.

## REFERENCES

1. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digitalsignal processing using approximate adders," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137,Jan. 2013.
2. E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncationscheme 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. Parhi, "Design low-error fixed-width modified booth multiplier," IEEE Trans. VeryLarge Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 522– 531,May 2004.
4. H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspiredimprecise computational blocks for efficient VLSI implementation ofsoft- computing applications," IEEE Trans. Circuits Syst. I, Reg. Papers,vol. 57, no. 4, pp. 850–862, Apr. 2010.
5. A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design andanalysis of approximate compressors for multiplication," IEEE Trans.Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.
6. S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim,"Energy-efficient approximate multiplication for digital signal processingand classification applications," IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.
7. G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi,"Design-efficient approximate multiplication circuits through partialproduct perforation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,vol. 24, no. 10, pp. 3105–3117, Oct. 2016.
8. P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy forpower 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 errorcorrection," in Proc. IEEE 31st Int. Conf. Comput. Design, Sep. 2013,pp. 33–38.
10. C. Liu, J. Han, and F. Lombardi, "A low-power, high-performanceapproximate multiplier with configurable partial error recovery," in Proc.Conf. Exhibit. (DATE), 2014, pp. 1–4.