International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

EFFICIENT IMPLEMENTATIONS OF REDUCED PRECISION REDUNDANCY (RPR) MULTIPLY AND ACCUMULATE (MAC)

[1] A Jyothirmayi, [2] Kandhagatla Mahathi, [3] Pothanaboina Rohitha

[1] Assistant Professor, Department of Electronics and Communication Engineering, BhojReddy Engineering College for Women, Hyderabad, Telangana, India.

[2,3,4] Students, Department of Electronics and Communication Engineering, BhojReddy Engineering College for Women, Hyderabad, Telangana, India.

Abstract

Multiply and Accumulate (MAC) is one of the most common operations in modern computing systems. It is for example used in matrix multiplication and in new computational environments such as those executed on neural networks for deep machine learning. MAC is also used in critical systems that must operate reliably such as object recognition for vehicles. Therefore, MAC implementations must be able to cope with errors that may be caused for example by radiation. A common scheme to deal with soft errors in arithmetic circuits is the use of Reduced Precision Redundancy (RPR).RPR instead of replicating the entire circuit, uses reduced precision copies which significantly reduce the overhead while still being able to correct the largest errors. This paper considers the implementation of RPR Multiply and Accumulate circuits. First, it is shown that the properties of signed integer multiplication (two´s complement format) can be used to make RPR more efficient. Then its principles are extended to the MAC operation by proposing RPR implementations that improve the error correction capabilities with a limited impact on circuit over head.The proposed schemes have been implemented and tested. The results show that they can significantly reduce the Mean Square Error (MSE) at the output when the circuit is affected by a soft error and the implementation overhead of the proposed schemes is low.

1.INTRODUCTION

Multiply and Accumulate (MAC, also known as multiply and add) is one of the most commonly used operations in computing systems. It is for example, the key element of matrix multiplication and it is also used in new algorithms like Deep Learning Neural Networks (DNNs).Stochastic Computation has been proposed for implementing artificial neural networks with reduced hardware and power consumption, while retaining in most cases the required accuracy and still improving the processing speed. MAC is extensively used in digital signal processing and its fused implementation has been extensively analyzed as part of the floating point IEEE 754 Standard.Matrix multiplication (MM) has a high complexity due to the large number of required multiplication and addition operations; however, its computational regularity can be exploited using a systolic

array. Different approximate schemes have been presented. In exact full adder cell s are utilized in a MAC-based processing element (PE) for the Baugh-Wooley multiplier and/or the final adder as circuits implementing the two computational steps required for matrix multiplication using an integer format an approximate design has been proposed for the Discrete Cosine Transform (DCT) using signed integers in the transformation matrix. The DCT processing is used for image convolution on a pixel basis ofan image by employing multiple approximate circuit and algorithm-level techniques such as truncation and voltage scaling. The integer version of MAC is widely used for example, in image processing.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

In many applications, MAC operations are part of a system that needs to operate reliably, like for example cloud computing infrastructure or vehicle security systems. In those cases, the implementations must be able to detect and correct errors to ensure that they do not cause any major failure. There are many types of error that can affect electronic circuits including manufacturing defects, ageing, electromagnetic disturbances and radiation induced soft errors. In particular, soft errors caused by radiation are challenging and hard to define. Detected in advance as it is the case with manufacturing defects or ageing. A soft error typically changes the value of a circuit node and thus can produce an incorrect result for the MAC operation.

II.LITERATURE SURVEY

In this chapter we will be observing the analysis and summary of previously done projects or reports. We will be observing Accelerating pipelined integer and floating point accumulations in configurable hardware with delayed addition techniques, A 175- MV multiply accumulate unit using an adaptive supply voltage and body bias architecture, Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators and Applications, A Stochastic Computational Multi-Layer Perceptron Supporting Backward Propagation Algorithm, Razor Based Programmable Truncated Multiply and Accumulate, Energy- Reduction for Efficient Digital Signal Processing, Design of an online multiply-add module for recursive digital filters and A Correctly Rounded Mixed- Radix Fused-Multiply-Add.

Accelerating pipelined integer and floating point accumulations in configurable hardware with delayed addition techniques

When an arithmetic calculation is carried out in a RISC microprocessor, each instruction typically has two source operands and one result. In many computations, however, the result of one arithmetic instruction is just an intermediate result in a long series of calculations. For example, dot product and other long summations use a long series of integer or floating-point operations to compute a final result. While FPGA designs often suffer from much slower clock rates than custom VLSI, configurable hardware allows us to make specialized hardware for these cases; with this, we can optimize the pipelining characteristics for the particular computation.

A typical multiplier in a full-custom integrated circuit has three stages. First, it uses Booth encoding to generate the partial products. Second, it uses one or more levels of Wallace tree compression to reduce the number of partial products to two. Third, it uses a final adder to add these two numbers and get the result. For such a multiplier, the third stage, performing the final add, generally takes about one-third of the total multiplication time. If implemented using FPGAs, stage 3 could become an even greater bottleneck because of the carry propagation problem is it hard to apply fast.Possible is the hardwired ripple-carry adder, the minimum delay of such an adder (in a y9 speed grade XC4000xla part) is more than four times the delay of an SRAM- based, 4-input look-up table that forms the core of the configurable logic blocks. Since this carry propagation is such a bottleneck, it impedes pipelining long series of additions or multiplies in configurable hardware; the carry propagation lies along the critical path, it determines the pipelined clock rate for the whole computation. Our work removes this bottleneck from the critical path so that stages 1 and 2 can run at full speed. This improves the performance of dot-products and other series calculations.

III.PROBLEM STAEMENT

Aim of our project is Reduced Precision Redundancy (RPR) as applicable to the commonly used operation of Multiply and Accumulate (MAC) in signed integer format. The proposed schemes utilize compensating error features in the multipliers and adders when signed integer processing is executed. Novel designs for MAC are proposed; initially the thresholds used for detecting errors within the RPR schemes have been determined analytically and checked by simulation. Then both the soft error tolerance and the implementation cost have been analyzed. The results show that a reduction of 40 to 60 percent on the Mean Square Error (MSE) can be achieved with only a marginal impact on the implementation cost in terms of delay and hardware overhead. As application, the computation of the DCT has beenpresented; the results confirm an improvement of PSNR of at least 5 dB on average and increasing with the reduced precision. There are many techniques to

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

protect a design against soft errors. Probably, the most common one is the use of replication in the form of Dual Modular Redundancy (DMR) to detect errors or Triple Modular Redundancy (TMR) to correct them. A major issue of modular redundancy is its implementation cost. For example, TMR requires circuit area and power of more than three times that of the unprotected design.TMR has also been utilized in a stochastic computational multi-layer perception design for neural networks by relying on backward propagation. A TMR based probability estimator and divider are employed. By utilizing a TMR voting structure in the processing element and divider, the error due to stochastic fluctuations in the binary search process for training purposes is significantly reduced. As a result, the latency and energy consumption are also reduced. The analysis in has confirmed that TMR will be required also for machine learning and the new computational systems required for its execution. For some arithmetic circuits, their properties can be used to reduce the protection cost. In the case of multipliers, for example residue codes and parity prediction have been used to provide 1an efficient protection. In both cases, the protection targets error detection, by checking the modulo of the output in the first case or its parity in the second case. To implement correction, two copies of the multiplier would be needed using the residue or parity checking to identify the correct one when they are different. This means that a cost that is still larger than twice the unprotected multiplier is needed. An alternative to complete redundancy such as triplication or duplication is to use Reduced Precision Redundancy (RPR). RPR uses a full precision version of the circuit and two reduced precision ones with an implementation of a voting logic to correct errors.

V OBJECTIVE

To evaluate the proposed schemes, first the thresholds used for detecting errors have been verified by simulation.Then both the soft error tolerance and the implementation cost have been analyzed. To that end, the proposed schemes have been implemented in HDL and For the MAC operation, the reduced precision redundancy can be implemented in different ways. One option could be to use a RPR multiplier followed by a RPR This one stage approach is illustrated in this second option reduces the voting logic and thus seems more attractive to reduce the implementation.

MAC Unit The main concerns of the signal processing system are multiplications and additions, the multiplier and accumulator is the basic building block for any of the digital signal processing systems. For this, MAC architecture represents the various functions of the different blocks for the high speed throughout the system for an efficient processing of the signals. Basically, the MAC is a DSP processor which is used for high-speed signalling such that the different blocks of the MAC unit should also perform the high-speed operations. This can be achieved by considering the different parameters such as power, speed and area. The block diagram of the MAC architecture is represented. Here an N- input bits of two sets are fetched from memory and given as input to the multiplier block which performs the multiplication and produces the 2N-bit output as input to the resister block which stores the data and sends the data to next level as input to the to the adder. This adder performs the adding operation by adding the output from the register block with the previously accumulated value that is stored in the accumulator register. Thus, the output from the adder is given to the accumulator register and the overall output is taken from the output of the accumulator register which is stored in the feedback register for next step. In the architecture of the MAC unit the blocks is considered to be of a high speed and low power consumption. The Multiplier and adder circuit is designed by considering the various

architectures and the final MAC unit is designed by using those individual blocks. This significantly lowers the overhead because the reduced precision copies are significantly smaller than the full precision implementation.

The price to pay is that small errors cannot be corrected because the reduced precision may induce small differences in output values that are difficult to discriminate. This howeveris not a problem for many applications in which most failures are due to large errors. In fact, it has been recently shown that the same reasoning can also apply to new circuits like for example

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Deep Learning Neural Networks (DNNs) for training purposes. Another issue whenRPR is used for simple structures is that the voting logic is significantly more complex than for TMR. This occurs because the difference between the full precision and reduced precision copies needs to be computed and compared to a threshold. Therefore, a subtraction and comparison are needed. This logic may be as complex as some operations, for example addition. The optimization of RPR for addition has been explored in which it was shown thatthe voting logic could be simplified for adders. In this work, we consider different RPR implementations of integer Multiply and Accumulate/Add (MAC) in two´s complement (signed) format.

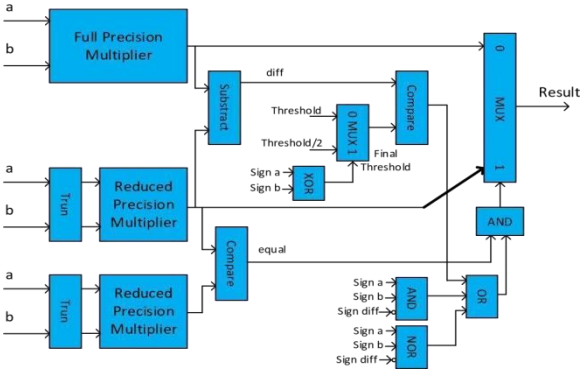
We first analyze signed multiplication to show that it is possible to optimize the RPR implementation. This is done by exploiting the dependence of the truncation error on the sign of the operands. In a second part, this technique is extended to the MAC operation using different schemes. To evaluate the benefits of the proposed schemes, both multipliers and MAC operations have been implemented and tested. The results show that the proposed schemes can improve the error correction capabilities of RPR for a given truncation, while requiring a low overhead in terms of implementation. Alternatively, the schemes could be used to increase the truncation and thus reduce the implementation cost while preserving the error detection capability

VI PROPOSED SCHEMES

The proposed scheme is based on the observation that for signed integers in two's complement format, the truncation error in a multiplication depends on the sign of the operands. This is clearly seen by remembering that truncation errors in two´s complement can only be negative and looking at the expression of the truncation.
When the two operands (a and b) have the same sign, the errors introduced by truncation add. However, when the operands have different signs, errors will compensate. Therefore, the difference between the full precision and the reduced precision copies will be larger in the first case. This can be exploited by using different thresholds depending on the sign of the operands.
In more detail, the magnitude of the largest error when operands have different signs occurs when one operand takes the maximum (in magnitude) value and its truncation error is zero, while the truncation error for the other operand is maximum (to minimize thecompensation).For a negative value, this would be:

$$(-2 + 1) \cdot (-2-1) \cong 2 \text{ plus or minus } 1$$

and a similar error is obtained for a positive value. This is half the worst case error seen for the general case previous section. Therefore, when the signs are different, one option would be to use a threshold that is half the value of the general threshold.
A second improvement can be made when both operands have the same sign, as then, the difference between the full precision results and the reduced precision results can only have one sign. For example, if both are positive, the difference can only be positive. This means that we can check the sign to detect errors. For example, a negative difference when the operands are both positive is not possible.
This enables the detection of errors that would go undetected in the traditional implementation.The proposed RPR multiplier that incorporates both improvements is shown in Fig. It can be seen that the threshold is divided by two when the signs of the operands aredifferent (the output of the XOR is one). In that case, the maximum error due to truncation is also smaller. Additionally, when the signs of the operands are the same andthe differencebetween the full and reduced precision multipliers has also the opposite sign,an error is detected. This is because truncation should produce a difference of the same sign. Therefore,a difference of the opposite sign can only be due to an error. In those cases, if the two reduced precision copies generate the same result, then the error has occurred in the full precision multiplier and the output from the reduced precision ones should be used.



Proposed Reduced Precision Redundancy for a Multiplier

Multiplication and Accumulate

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

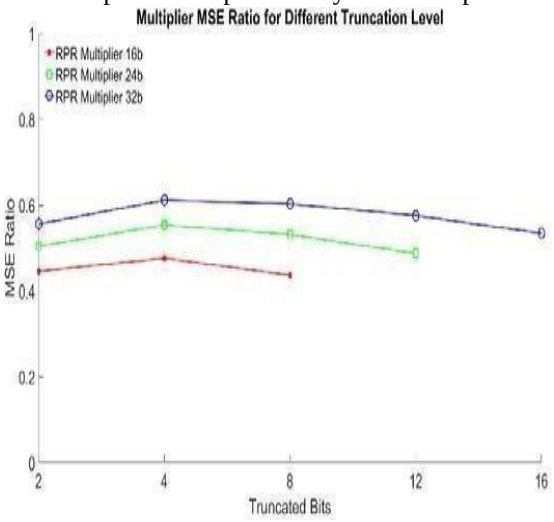
In the case of MAC, the design can be further improved by compensating the truncation error of the multiplication with the adder. The flowchart of the proposed optimizations for the MAC operation is shown in Fig . It can be seen that the sign of the operands is exploited to reduce the error both in the multiplication (left) and addition (right). This new scheme is illustrated in Fig the carry-in signal of the adder is set based on the sign of the multiplication operands. So, when the operands are positive, truncation in the multiplicationwill introduce a positive error thus setting the carry-in of the adder to one will introduce an error of opposite sign such that both errors will tend to compensate. This means that the worst case error for positive operands will be slightly smaller.

Evaluation

To evaluate the proposed schemes, first the thresholds used for detecting errors have been verified by simulation. Then both the soft error tolerance and the implementationcost have been analyzed. To that end, the proposed schemes have been implemented inHDL and mapped to the Free PDK 45nm library.

Threshold Configuration

The thresholds used for error detection are those obtained in the analysis presented in section2 and section 3 for the traditional and proposed implementations respectively. More precisely,2N+K in the first case and 2N+K-1 in the second case when operands have different signs and2N+K with sign checking when they have the same sign. To check that those values are indeed correct, simulations have been done with different thresholds for voting and a variable referred to as the inexact ratio is measured. The inexact ratio is defined as the fraction of cases for which the output is taken from the reduced precision version in an error free condition. As an example, the inexact ratio versus the threshold value for N=16 is illustrated in Figure 4. It can be seen that there is one threshold value (referred to as critical) from which the inexact ratio becomes zero both for a traditional RPR multiplier and for the proposed scheme. In this case, this value is 220. The same simulations have been done for other word widths(N) and truncation levels (K) and the results are shown in Table 1. It can be seen that in allcases, the values obtained are given by 2N+K which was the value obtained in the previous sections by analyzing the worst case error due to truncation (in the proposed scheme, that threshold is divided by two when operands have different signs).The Multiply- Accumulate (MAC) operation is a mathematical calculation used in digital signal processing, image processing, and other fields. It involves multiplying two numbers together and adding the result to an accumulator value. The MAC operation is particularly useful for performing



complex mathematical calculations quickly and efficiently. It is often used in applications such as filtering, convolution, and correlation. The MAC operation can be implemented in hardware or software, and thereare many variations of the basic operation that are used in different applications. A multiplier is a digital circuit that performs multiplication of two binary numbers. It is a fundamental building block in digital signal processing, image processing, and otherapplications that involve complex mathematical calculations.

In the rest of the simulations, the threshold values given are used. The valuesobtained for the MAC operation were the same.

Proposed Reduced Precision Redundancy for Multiplyand

Accumulate

Instead of performing these operations separately, a multiplying accumulator allows the multiplication and accumulation to be performed simultaneously, effectively reducing the number of clock cycles required for the computation. By using a multiplying accumulator, the computational efficiency can be significantly improved, especially in applications such as digital filters, Fast Fourier Transform (FFT), and matrix operations, where a large number of multiplications and accumulations

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

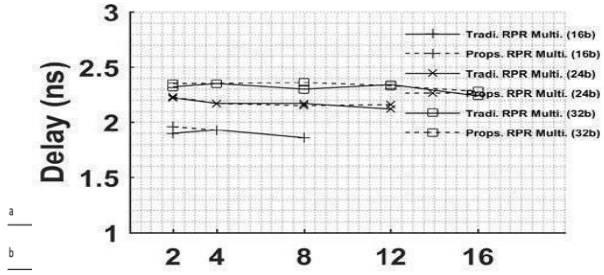
are involved. It's important to note that both reduced precision redundancy and multiplying accumulators involve trade- offs.

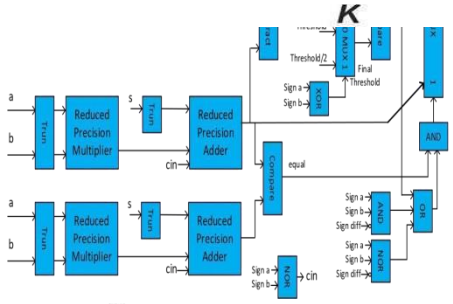
While they can improve efficiency and performance, they may introduce some loss of accuracy or precision in the computations. Therefore, careful consideration and analysis are required to determine the appropriate level of precision reduction and the optimal use of multiplying accumulators based on the specific application requirements and the acceptablelevel of error or loss of precision.

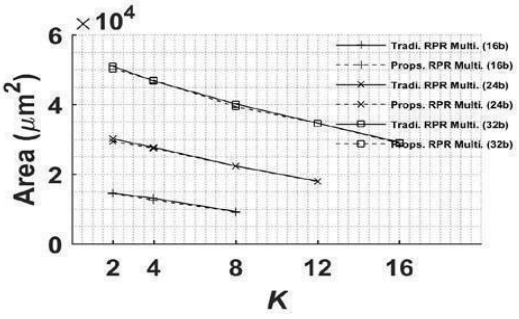
Soft Error Tolerance

To evaluate the impact of soft errors on the traditional and the proposed RPR schemes, random errors that are uniformly distributed have been injected in both implementations. Then, the square error has been measured. The Mean Square Error (MSE) across all injections is used compare the implementations the MSE ratios between the proposed are given.

Multiplier and the traditional RPR multiplier are presented when the number of bits is 16, 24 and 32. It can be seen that the ratio is in the 0.4 to 0.6 range. Therefor e, the proposed scheme can reduce the MSE by 40 to 60 per cent. The reductions are larger for smaller word widths and they also



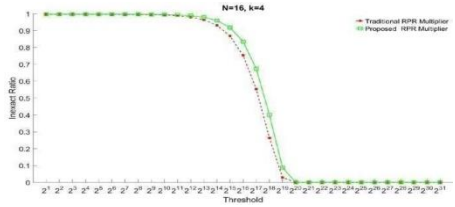




improve as truncation increases. The results for the MAC operation. It can be observed that they are very similar to those of the multiplier. Onereasonfor this phenomenon is that the probability of the multiplier experiencing a soft erroris significantly larger than the probability of an adder experiencing a soft error due to the complexity difference between these two arithmetic circuits. Another reason is that the truncation in the multiplication ends to dominate the error.

Implementation cost

The area, delay, power and PDP of the RPR multipliers



analyzed in the previous subsectionare plotted. It can be observed that the delay does not decrease significantly by increasing

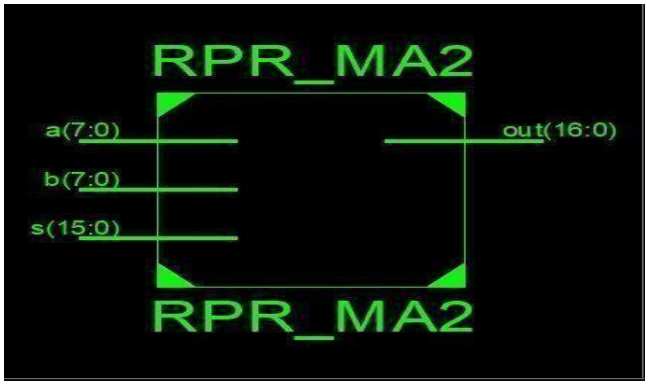
K. The proposed RPR multiplier design is slightly slower than the traditional RPR multiplier design. However, the power and area decrease rapidly with increasing K. The difference between the proposed and traditional RPR multiplier designs is negligible as expected, because the proposed RPR multiplier design just has a few more gates than the traditional RPR multiplier design. The PDP difference of the proposed and traditional RPR multiplier designs is also marginal. In summary, the overall delay and hardware cost of the proposed and traditional design are nearly the same.

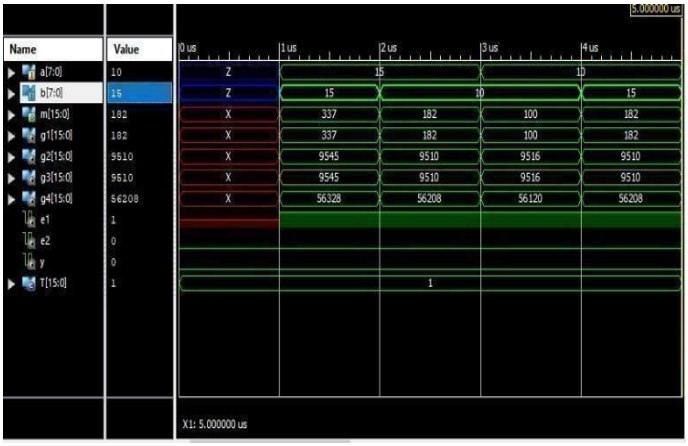
Fig 5.5 Area for RPR multipliers protected with different levels of truncation

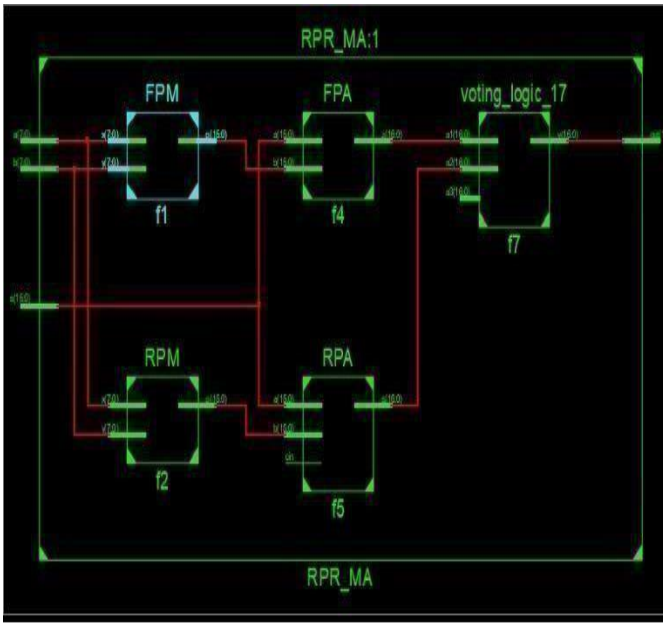
Delayfor RPR multipliers protected with different levels of truncation

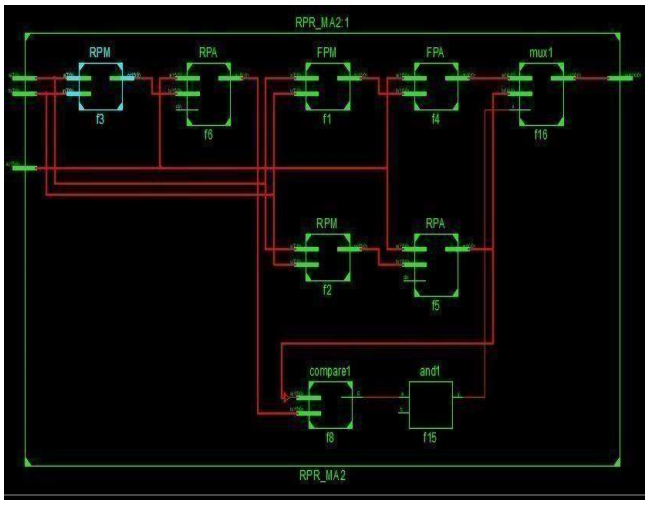
International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

VII RESULTS





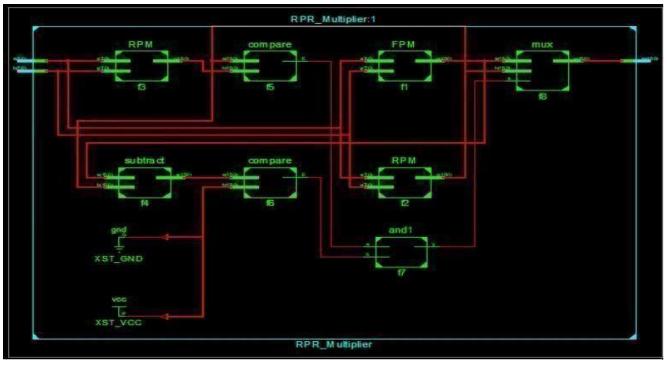




VIII CONCLUSION

The conclusion of multiplication and accumulator reduced precision redundancy (often referred to as "rounding") depends on the specific context and requirements of the system or application being considered. Here are a few general points to consider: Precision Trade- off, reducing the precision of multiplication and accumulator operations can lead to computational efficiency by reducing the complexity and resource requirements of the hardware. However, it also introduces a loss of precision in the calculations, which can affect the accuracy of the results.

Impact on Result Accuracy: The impact of reduced precision on result accuracy depends on the specific application and the level of precision required. In some



cases, a slight loss of precision may be acceptable and may not significantly affect the overall performance or outcome. However, in other cases, such as scientific calculations or financial applications, maintaining high precision is crucial, and reducing it can lead to

unacceptable errors.

Error Analysis: When considering reduced precision in multiplication and accumulation, it is important to perform an error analysis to determine the potential impact on the final results. This analysis involves evaluating the error introduced at each step and assessing its propagation through out the computation. Various techniques, such as Monte Carlo simulations or analytical error propagation models, can be employed to estimate the overall impact on result accuracy.

System Requirements: The decision to employ reduced **precision in multiplication and accumulator operations should align with the system's requirements and** constraints. If the application permits a loss of precision without significant consequences and prioritizes computational efficiency, reducing precision can be a viable option. However, if high accuracy is crucial or if the application involves critical calculations, maintaining full precision may be necessary. Ultimately, the conclusion regarding the use of reduced precision in multiplication and accumulator operations depends on a careful considerationof the specific application, the desired level of precision, and the trade- offs between computational efficiency and result accuracy. It is essential to perform a thorough analysis and evaluation to make an informed decision in each particular case.

References

[1]. Z. Luo and M. Martonosi “Accelerating pipelined integer and floatingpoint accumulations in configurable hardware with delayed addition techniques.” IEEE Transactions on Computers,Bol. 49, No. 3, pp. 208-218, 2014.

[2]. J. T. Kao; M. Miyazaki; A. R. Chandrakasan “A 175-MV multiplyaccumulate unit using an adaptive supply voltage and body bias architecture,” IEEE Journal of Solid- State Circuits, Vol. 37, No. 11, pp. 1545-1554, 2020.

[3]. G. Li, S. Kumar, S. Hari, M. Sullivan, T. Tsai, K., Pattabiraman, J. Emer, and S.W. Keckle,“Understanding Error Propagation in Deep Learning Neural Network (DNN) Accelerators andApplications”, In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 8:1–8:12, 2017.

[4]. Y. Liu, S. Liu, Y. Wang, F. Lombardi and J. Han, “A Stochastic Computational Multi- LayerPerceptron Supporting a Backward Propagation Algorithm,” IEEE Transactions on Computers(in press).

[5]. M. de la Guia Solaz and R. Conway “Razor Based Programmable Truncated Multiply and Accumulate, Energy-Reduction for Efficient Digital Signal Processing,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol 23, No. 1, pp. 189 – 193, 2019.

[6]. R. H. Brackert, M. D. Ercegovac, and A. N. Willson “Design of an online multiply- add module for recursive digital filters”, In proceedings of 9th IEEE Symposium on Computer Arithmetic, pp. 34-41, 2016.

[7]. C. Jeangoudoux and C. Lauter “A Correctly Rounded Mixed-Radix Fused-Multiply- Add”In Proceedings of the 25th IEEE Symposium on Computer Arithmetic, pp 17-24, Amherst, 2020.

[8]. K. Chen, J. Han and F. Lombardi, “Design and Analysis of an Approximate 2D Convolver,”in Proceedings of the IEEE International Symposium on DFT in VLSI and Nanotechnology Systems, pp. 31-34, Storrs, October 2019.

[9]. K. Chen, J. Han and F. Lombardi "Matrix Multiplication by an Inexact Systolic Array" in Proceedings of the ACM/IEEE Symposium on Nano Architectures, pp 151-156, Boston,July 2021.

Asmita Havelia,”A Novel Design for High Speed Multiplier for Digital Signal Processing Applications”, Dept. of Electronics, ASET, Amity University Lucknow, India.

Shamim Akhter, “VHDL Implementation of Fast N X N Multiplier Based On Vedic Mathematics, Jaypee Institute of Information Technology University, Noida,201307 UP,India,2007 IEEE

Ramesh Pushpangadam, Vineeth Sukumaran, Rino Innocent, Dinesh Sasikumar, Vaisakh Sundar,”High Speed Vedic Multiplier for Digital Signal Processors ”, Dept.of Ece, College of Engineering, Munnar, PB NO 45 County Hills, Idukki, Kerala, India