

Analysis of inexact Computing of Truncated Multiplier in Image Multiplication

Indira.V¹, Madhavan.J²

PG Student, Dept. of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India¹

Associate Professor, Dept. of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India²

ABSTRACT: In this project inexact computing is particularly attractive for computer arithmetic designs. The Implementation of 8X8 truncated multipliers can be used in the image multiplication application. This multiplier is mechanically truncating the output and reduces the power consumption and also comparing to other multipliers. The truncated multiplier shows much more reduction in device utilization as compared to standard multiplier. The Significant reduction in delay, area and power consumption can be achieved by using truncated multipliers instead of standard parallel multipliers. The proposed work of this project can be implemented by VHDL using Xilinx 14.2.

KEYWORDS: Truncated multiplier, Compressor, VHDL, Image multiplication

I. INTRODUCTION

The common computer arithmetic software and engineering problems be realized using digital logic circuits, as a result its functionality has a higher level of constancy and accuracy. On the other hand, several programs for illustration multimedia systems and image/signal processing can accept approximate outcomes but still generate results are good enough for human perception. Erroneous design and algorithms are efficient for utilization these kinds of applications. Using incorrect design for error resilient system has benefit of reduction in circuit complexity and its area, cost, better performance and power efficient compared with logic circuits [1].

Multiplication is one of the most area consuming arithmetic operations in high-performance circuits. As a importance many research works deal among low power design of high speed multipliers. Multiplication involves two basic operations, the generation of the partial products and their sum, performed using two kinds of multiplication algorithms, serial and parallel [2]. The Serial multiplication algorithm uses sequential circuits with feedbacks: inner products are sequentially produced and computed. Parallel multiplication algorithm frequently uses combinational circuits and do not contain many feedback structures.

Multiplication of two bits produces an output which is twice that the original bit. It is usually needed to truncate the partial product bits to required precision to reduce area cost. Fixed-width multipliers, a subset of truncated multipliers, compute only n most considerable bits (MSBs) of the 2n-bit product for $n \times n$ multiplication and use extra correction/compensation circuits to reduce truncation errors. In previous related papers, to reduce the truncation error by adding error compensation circuits.

II. MULTIPLIER

For digital signal processing and applications involving signal and image processing, multipliers and adders form an important part. So, speed the multipliers and adders affect rapidity the operation.

For high speed applications, faster multipliers are not compulsory. Several techniques to increase multiplier speed are proposed. Multiplication is done by adding partial product terms. Multiplication involves mainly 3 steps:

- Partial Product Generation
- Partial product reduction schemes to two rows
- Final addition of two rows of partial product by using a carry propagate adder

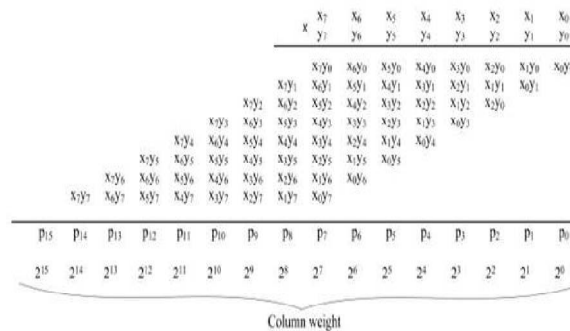


Fig.1 Standard parallel 8 x 8 bit multiplier

Multipliers play an important role in today's digital signal processing and various other applications. By way of advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or great combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation design. An efficient multiplier should have following characteristics:

Speed: Multiplier should perform operation at high speed.

Area: A multiplier should occupy less number of slices and LUTs.

Power: Multiplier should consume less power.

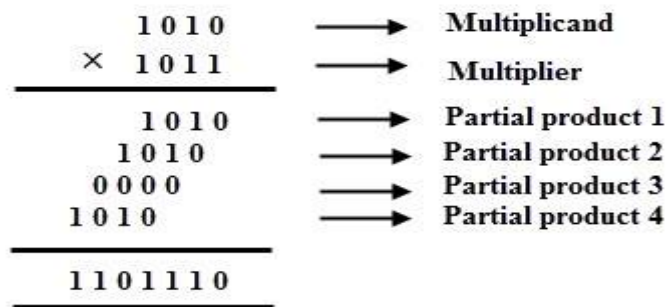


Fig.2 Manual calculation of binary multiplier

III. PREVIOUS DESIGN OF MULTIPLIER

Design the multiplier based on approximate Full adder are proposed. Intuitively to design an approximate 4-2 compressor, it is possible to substitute the exact full-adder cells in Fig. by an approximate full-adder cell. However, this be not very efficient, because it produce at least 17 incorrect results out of 32 possible outputs, i.e., the error rate of this not exact compressor is more than 53 percent (where the error rate is given by the ratio of the number of erroneous outputs over the total number of outputs). Two different designs are proposed next to reduce the error rate; these designs offer significant performance development compared to an exact compressor with respect to delay, number of transistors and power consumption. The common implementation of a 4-2 compressor is accomplished by utilizing two approximate full-adder (FA) cells.

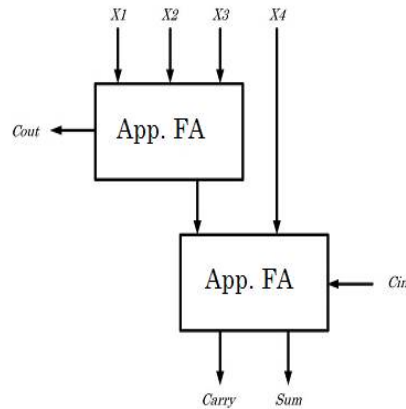


Fig.3 Compressor based on the app. FA

A. PARTIAL PRODUCT GENERATION

A carry save adder (CSA) tree to reduce the partial products, matrix to an addition of only two operands. A carry propagation adder (CPA) is the final computation of the binary result.

In the design of a multiplier, the second module plays a pivotal role in terms of delay, power consumption and circuit complexity. Compressors have been widely used to speed up the CSA tree and decrease its power dissipation, so to achieve fast and low-power operation. The use of approximate compressors in the CSA tree of a multiplier results in an approximate multiplier.

An 8*8 unsigned Dadda tree multiplier is considered to evaluate the impact using the proposed compressors in approximate multipliers. The proposed multiplier are used to the first part AND gates to generate all partial products. In the second part, the approximate compressors proposed in the previous section are utilized in the CSA tree to reduce the partial products. The last part is an exact CPA to work out the final binary result.

The two most famous reduction methods are Wallace tree and Dadda tree reductions. Wallace tree reduction manages to compress the Partial Product as early as possible, whereas Dadda reduction only performs compression whenever necessary without increasing the number of carry-save addition (CSA) levels. Here we design a multiplier based on Wallace tree only.

IV. PROPOSED WORK OF TRUNCATED MULTIPLIER

A multiplier is used to multiply two images on a pixel by pixel basis, thus combination the two images into a single output image. It shows two examples: both input images and the resulting output image are provided. The average NED and the peak signal-to-noise ratio that is based on the mean squared error (MSE) are computed to evaluate the quality of the output image and compare it with the output image generated by a literal multiplier.

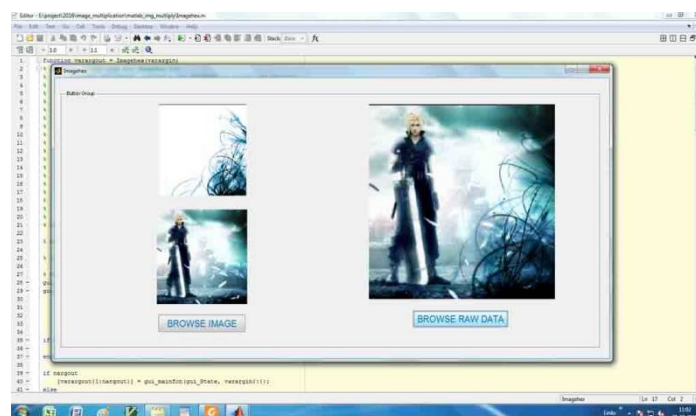


Fig.4 GUT

In proposed architecture we can multiply 8x8 bits, and the bits are reduced in step by step manner. Then the number of stages to reduce the final bit width is without increasing the error. In normal truncated multiplier design, the architecture produces the output with some truncation error. But in the proposed design of truncated multiplier the truncation error is not more than 1, so the precision of the final result is improved. It shows proposed truncated multiplier.

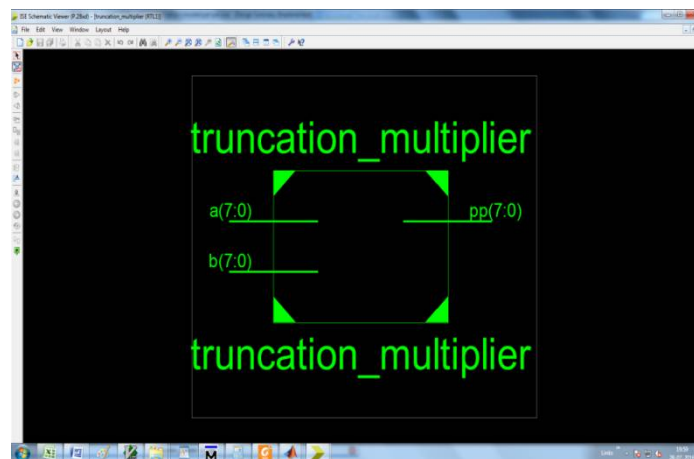


Fig.5 RTL View

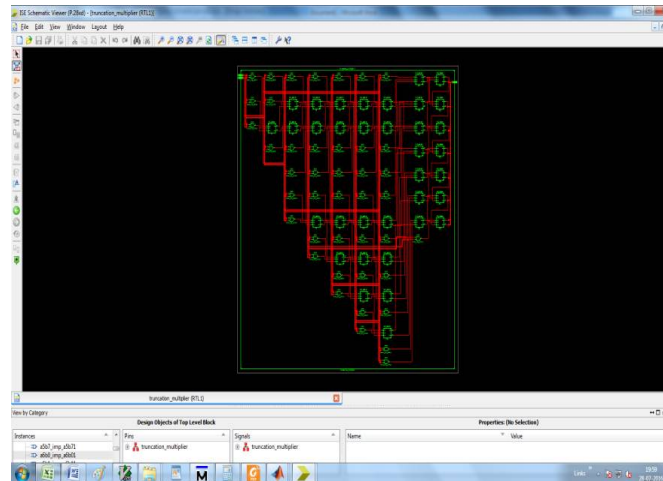


Fig.6 Overall architecture

We have presented hardware design and implementation of FPGA based on parallel architecture for standard and truncated 8x8 multipliers utilizing VHDL. Both the design was implemented on Xilinx Spartan 3AN XC3S700AN device. The desire is to present a virtual study of the standard and truncated 8x8 multipliers. The truncated multiplier as compared to standard multiplier shows much more reduction in device utilization.

V.SIMULATION RESULT

The truncated multiplier for general $M \times N$ unsigned multiplication with a full product of $M+N$ bits truncated to P bits. In other words, there are $T = M + N - P$ bits truncated. First, perform deletion in stage 1 to remove the PP bits, as long as the magnitude of the total deletion error is not more. Bits $[Col]$ represent the number of PP bits of column.

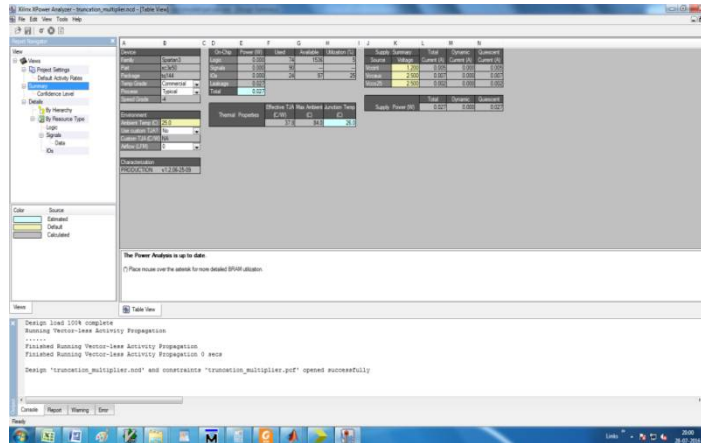


Fig.7 Power report

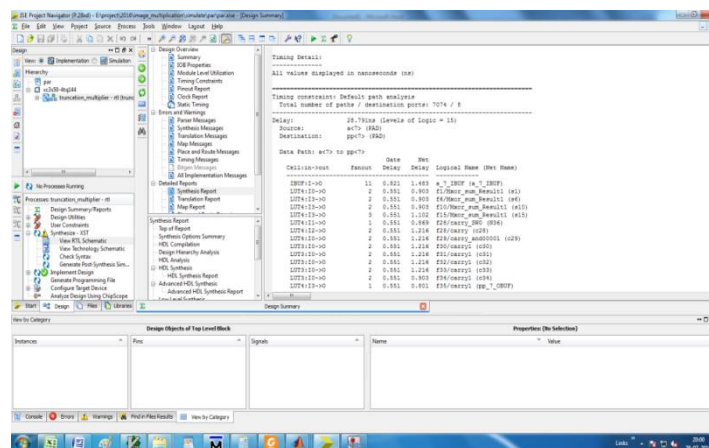


Fig.8 Delay report

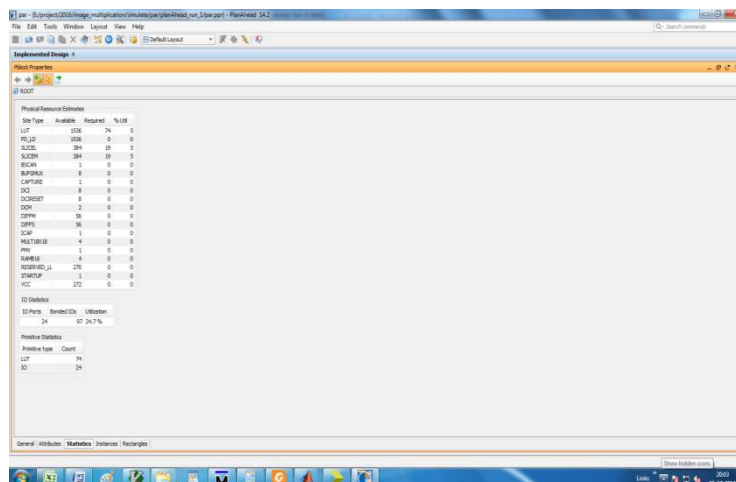


Fig.9 Area report

VI. CONCLUSION

Computer arithmetic offers significant operational compensation for inexact computing; an extensive literature exists on approximate adders. Inexact computing is an emerging paradigm for computation at nano-scale. We analyze the performance, area, power of the truncated multiplier in the Xilinx. Finally we get the multiplied image and the low energy/area multiplier architecture design with power consumption of 0.027W and delay of 28.791ns. In future use different multiplier for low energy and high speed performance in the image multiplication.

REFERENCES

- [1] E. J. King and E. E. Swartzlander Jr., "Data dependent truncated scheme for parallel multiplication," in Proc. 31st Asilomar Conf. Signals, Circuits Syst., 1998, pp. 1178–1182.
- [2] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in a multiplier architecture," J. Low Power Electron., vol. 7, no. 4, pp. 490–501, 2011.
- [3] C. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," IEEE Trans. Circuits Syst., vol. 51, no. 10, pp. 1985–1997, Oct. 2004.
- [4] D. Radhakrishnan and A. P. Preethy, "Low-Power CMOS pass logic 4-2 compressor for high-speed multiplication," in Proc. IEEE 43rd Midwest Symp. Circuits Syst., 2000, vol. 3, pp. 1296–1298.
- [5] Z. Wang, G. A. Jullien, and W. C. Miller, "A new design technique for column compression multipliers," IEEE Trans. Comput., vol. 44, no. 8, pp. 962–970, Aug. 1995.
- [6] J. Gu and C. H. Chang, "Ultra low-voltage, low-power 4-2 compressor for high speed multiplications," in Proc. 36th IEEE Int. Symp. Circuits Syst., Bangkok, Thailand, May 2003, pp. v-321–v-324.
- [7] M. Margala and N. G. Durdle, "Low-power low-voltage 4-2 compressors for VLSI Applications," in Proc. IEEE Alessandro Volta Memorial Workshop Low-Power Design, 1999, pp. 84–90.
- [8] B. Parhami, Computer Arithmetic: Algorithms and Hardware Designs, 2nd ed. London, U.K.: Oxford Univ. Press, 2010.
- [9] K. Prasad and K. K. Parhi, "Low-power 4-2 and 5-2 compressors," in Proc. 35th Asilomar Conf. Signals, Syst. Comput., 2001, vol. 1, pp. 129–133.
- [10] M. D. Ercegovac and T. Lang, Digital Arithmetic. Amsterdam, The Netherlands: Elsevier, 2003.
- [11] D. Baran, M. Aktan, and V. G. Oklobdzija, "Energy efficient implementation of parallel CMOS multipliers with improved compressors," in Proc. ACM/IEEE 16th Int. Symp. Low Power Electron. Design, 2010, pp. 147–152.
- [12] D. Kelly, B. Phillips, and S. Al-Sarawi, "Approximate signed binary integer multipliers for arithmetic data value speculation," in Proc. Conf. Design Architect. Signal Image Process. 2009, pp. 97–104.