ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 8, Issue 3, March 2020

Approximate Radix-8 Booth Multipliers for Low Power and High Performance Operation

M.Kaviyarasi[1], M.Kamalnathan[2]

PG Student, Department of VLSI Design, Gnanamani College of Technology, Namakkal (Dt), Tamilnadu, India [1],

Assistant Professor, Department of ECE, Gnanamani College of Technology, Namakkal (Dt), Tamilnadu, India [2]

ABSTRACT-In CMOS technology and VLSI design, the power consumption becomes a major problem. In existing system, ArrayBased Approximate Arithmetic Computing (AAAC) has the building block of Error Compensation Unit (ECU).In this existing system multiplier and Squarer applications are designed to obtain optimal trade off between area, delay and energy consumption of AAAC circuits. In this system it occupies a larger area by which it increases the power consumption and delay. By the use of truncation it can improve the speed of the computation, but it has limitation that it does not consider the minimum bit error.To overcome this, Error Compensation unit is designed using logic formulation to minimize delay and reduce area and also reduce power consumption compared to existing method. Multiplier and Squarer are coded in VHDL and simulated in ModelSim and synthesized in EDA tool Xilinx_ISE 9.2i. Then this technique will be implemented in Spartan 3E.

KEYWORDS: Booth multiplier,squarer, compressor, encoder.

I.INTRODUTION

Very-large-scale integration (VLSI) is the process of creating an integrated circuits(IC) by combining thousands of transistors into a single chip. Multiplication is a basic arithmetic operation which is present in many part of the digital computer especially in signal processing systems such as graphics and computation system. It requires more hardware resources and processing time than addition and subtraction requires. There is a continuous development in VLSI technologies. As the scale of integration keeps growing, more and more sophisticated signal processing system are being implemented on a VLSI chip. These signal processing application not only demand great computation capacity but also consume considerable amount of energy.Multipliers play an important role in today's digital signal processing and various other applications. With 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 even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.Squares are a special case of multiplication where both inputs are identical. Since the two input are identical, many optimizations can be made in the implementation of a dedicated squaring unit. While speed and area remain to be the major design tools. The higher speed results to enlarged power consumption, thus, low power architecture will be the choice of the future. Low power design directly leads to prolonged operation time in these portable devices.Multiplication can be done serially or parallel. Theoretically multiplication can be done by repeated addition. Consider multiplication A x B where A is multiplier and B is multiplicand, if we add B to itself A times the sum will be the product of A x B. Practically this process is very slow so never been used. This method involves generating intermediate products and then adding them properly taking into account the weight of each bit while moving from LSB to MSB.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 8, Issue 3, March 2020

II.DESIGN OF BOOTH MULTIPLIER AND SQUARER

A.Booth multiplier

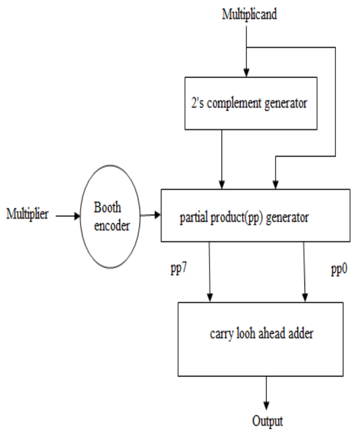

Fig 1.Booth Multiplier

It is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly. For the standard add-shift operation, each multiplier bit generates one multiple of the multiplicand to be added to the partial product. If the multiplier is very large, then a large number of multiplicands have to be added. In this case the delay of multiplier is determined mainly by the number of additions to be performed. If there is a way to reduce the number of the additions, the performance will get better. Booth algorithm[2] is a method that will reduce the number of multiplicand multiples. Since a k-bit binary number can be interpreted as K/2 -digit radix-4 number, a K/3-digit radix-8 number, and so on, it can deal with more than one bit of the multiplier in each cycle by using high radix multiplication.

B.Modified Booth Multiplier and Squarer

Booth encoding is a method of reducing the number of partial products required to products the multiplication result. To achieve high-speed multiplication, algorithm has been proposed and used. This type of fast multiplier operates much faster than an array multiplier operates much faster than an array multiplier for longer operands because it's time to compute is proportional to the logarithm of the word length of operands.

Table1.Recoding Table

Block	Re-coded digit	Operation
000	0	0
001	+1	+1
010	+1	+1
011	+2	+2
100	-2	-2
101	-1	-1
110	-1	-1
111	0	0

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 8, Issue 3, March 2020

III. DESIGN OF AAAC

A. AAAC

Error-Free Computing Unit (EFCU) with n-bit inputs and an m-bit output (left) with its approximate counterpart modelled using the proposed AAAC model (right).

The AAAC model consists of three units:

A) Low-Precision Computing Unit (LPCU)
B) Error Compensation Unit(ECU)
C) Combine Unit (CU).

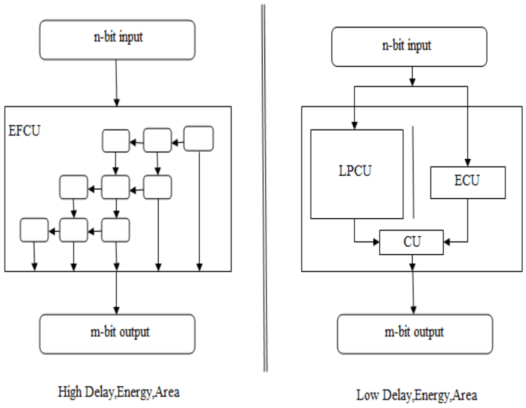

Fig 2.Block Diagram

The LPCU in the AAAC circuit produces a low-precision approximate output, for example, based upon truncation or a fraction of the input bits, with lowered energy, delay and/or area overheads compared with the error-free EFCU. To re-duce the error produced by the LPCU, a low-cost ECU may be included for error comparison. Finally, the CU combines the error compensation produced by the ECU with the result outputted by the LPCU, generating the final output of the AAAC unit with reduced approximate error. The generality of the AAAC model[7] lies in the fact it reflects the key computing principles behind a wide range of array based arithmetic units, for example, approximate adders, multipliers and squarers. For instance, many approximate adders employ carry prediction from low input bits, which can be thought as a particular way of implementing the ECU. Similarly, error compensation is a common scheme in approximate multipliers and squarers.

Approximate multipliers and squarers have been a focus of a great deal of past and ongoing work. Two types of approximate multipliers exist: approximate AND-array multipliers, which utilize AND gates for partial product generation and approximate Booth multipliers, which use the modified Booth algorithm to reduce the number of partial products.

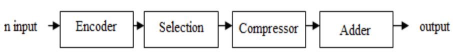

Fig.2 Error Free Booth Multiplier

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 8, Issue 3, March 2020

B. Fixed-Width 16x16 Booth Multipliers

Booth multipliers[1] are ideal for high speed applications and the Radix-4 Modified Booth multipliers, are most widely applied. The main blocks of a Radix-4 Modified Booth multiplier half number of partial products needed for array multipliers with each product being one of the following: 0, A, 2A, -A, -2A.

Table II

Booth's Radix-4 Algorithm Table

x_i	x_{i-1}	x_{i-2}	Operation	Comments
0	0	0	+0	String of zeroes
0	1	0	+A	A Single 1
1	0	0	-2A	Beginning of 1's
1	1	0	-A	Beginning of 1's
0	0	1	+A	End of 1's
0	1	1	+2A	End of 1's
1	0	1	-A	A Single 0
1	1	1	+0	String of 0's

Fixed-width Booth multipliers[3] to refer to approximate Booth multipliers, that operate on two –bit inputs while outputting only an -bit product.Full-precision 16x16 Booth multiplier where each dot row (PP_0 to PP_7)is a partial product. The 16 dots (bits) in each PP_i are denoted by $pp_{i,15},pp_{i,14}...pp_{i,0}$ from left to right, is the correction constant required to generate the negative partial product, and s_i is sign of the i^{th} partial product[6]. The vertical dashed line splits the array at the position of the binary (radix) point. A fixed-width multiplier outputs an integer output by approximating the carry-out produced by the fractional part of the array, which is also labeled as the truncation part (TP). On the other hand, the contribution of the bits left of the binary point, i.e., ones in the accurate part (AP), is not approximated.

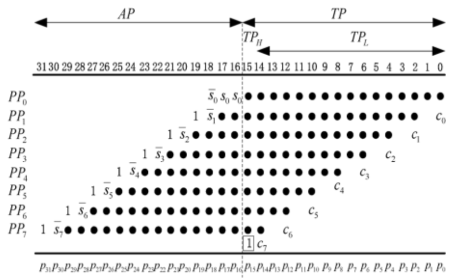

Fig.3 Partial Product Diagram for Fixed-Width 16x16 Booth Multipliers(N=16)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 8, Issue 3, March 2020

Direct-truncated Booth multipliers (DTM), which are an extreme case of fixed-width multipliers [4], output an -bit integer product by simply neglecting the bits in the TP part of the array without forming them in the first place, thus potentially producing a large error. As another extreme, post-truncated Booth multipliers (PTM) form the complete partial product array, compress all the bits, compute with full precision, add an extra "1" to the n-1th column to exactly round the carry-out to the nth column, and finally output the exact -bit integer part of the final product (with rounding).

C. 16-Bit Fixed-Width Squarers

Fig.4 shows the full 8-partial squaring array (PS_0 to PS_7) for a full-precision 16-bit squarer, where the input is denoted by $A(a_{n-1}...a_0)$. Here, we use the method in to implement squarers instead of using Booth algorithm are more energy-efficient and faster since most partial products[8] bits are implemented by simple AND operation of two input bits instead of more complex Booth encoding and selection blocks. The squarer design process is very similar to the one presented for the proposed multipliers.

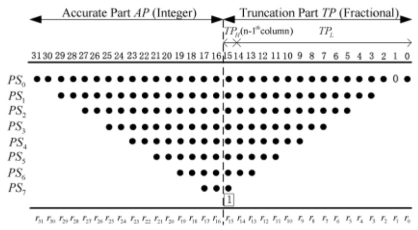

Fig.4 Squaring Diagram for 16-Bit Fixed-Width Squarers.

IV.RESULT ANALYSIS

The results are obtained if one of the negative operand is in 2's complement It also speeds up multiplication process by analysis multiple bits of multiplier at a time.

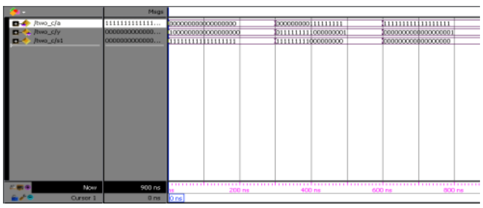

Fig 7.Waveform of 2's complement

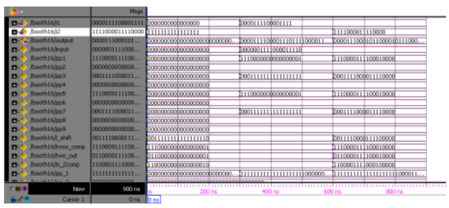

Fig 8. Waveform of 16 x 16 Booth Multiplier

Figure 7 shows the waveform of 2's complement and the waveform of 16 x16 booth multiplier is show in figure 8.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 8, Issue 3, March 2020

Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added. The addition can be performed with normal carry propagate adder.

Synthesis Report:

Table-IV: Comparision of values

Methods	Operation	Delay(ns)	Number of LUT's	Number of Slices	Power (mW)
Existing	Multiplier	12.66	1644	1263	424
	Squarer	6.14	925	1075	412
Proposed	Multiplier	10.24	1200	683	364
	Squarer	3.076	562	551	360

The Table IV shows the comparison of delay, number of slices and number of LUT's of the Fixed width Booth multiplier and Squarer of existing and proposed method.

V. CONCLUSION

In this project general model is presented for Array-based approximate arithmetic computing to guide the design of approximate Booth multipliers and squarers, the impact of area,delay have been studied and analyzed. Simulation results have been calculated. To further reduce energy consumption and area EUC model is designed. The proposed approach has led to significant performance improvement for a number of approximate multiplier and squarer design.

REFERENCES

[1]Chen Y.H. (2014), 'An accuracy-adjustment fixed-width booth multiplier based on multilevel conditional probability', IEEE Transaction Very Large Scale Integration (VLSI) System, Vol 23, No.1, pp. 203-207.
[2]Huang H.A. et Al. (2006), 'A self-compensation fixed-width Booth multiplier and its 128-point FFT applications', Proc. IEEE ISCAS, pp. 3538-3541.
[3]Jeena Maria Cherian and Sireesha B. (2014), 'Design of high-accuracy fixed-width modified booth multiplier', International Journal of Computer Science and Information Technology, Vol.3, pp.238-290.
[4]Jou S. J. (2003), 'Low-error reduced-width Booth multipliers for DSP applications', IEEE Trans. Circuits Syst. I, Vol. 50, No. 11, pp. 1470–1474.
[5]Juang T.S and Hsiao S.F. (2005), 'Low-error carry-free fixed-width multipliers with low-cost compensation circuits', IEEE Trans. Circuits Syst.II, vol. 52, no. 6, pp. 299–303.
[6]Kuang S.R. et al (2009) , 'Modified Booth multipliers with a regular partial product array' , IEEE Trans. Circuits Syst. II,Vol. 56, No. 5, pp. 404–408.
[7]Shano B. and Li P.(2014), 'A model for array-based approximate arithmetic computing with application to multiplier and squarer design' ,Low Power Electron Design, pp. 9-14.
[8]Oklobdzija V.G. et al (1996),'A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach,' IEEE Trans. Comput.,Vol. 45, No.3,pp 294–306.