

# A New Multiplier - Accumulator Architecture based on High Accuracy Modified Booth Algorithm

J. Y. Yaswanth babu, P. Dinesh Kumar

**Abstract**— In this paper, a new MAC architecture is developed for high speed performance. The performance can be improved by developing a new carry save adder which is designed by combining multiplication with accumulation. The overall performance will be improved because of merging the accumulator, which has largest delay, into CSA. The CSA tree uses modified Booth algorithm(MBA) which provides the high accuracy instead of using radix 2 modified booth algorithm in present technique. Least significant bits are generated in advance to reduce the number of inputs to the final adder by propagating carries to the least significant bits by CSA. Instead of the final adder output, the intermediate results, sum and carry, are accumulated. The MAC architecture is synthesized with 180nm standard CMOS library using cadence SOC encounter.

**Index Terms**— Carry save adder, digital signal processing, modified booth algorithm, multiplier-and-accumulator.

## I. INTRODUCTION

Real time signal processings like large capacity data processing, audio signal processing are most advanced techniques in present multimedia and communication systems. Filtering, inner products and convolution are some important factors in digital signal processing. The essential elements for these operations are MAC and multiplier[5]. Multipliers are the fundamental arithmetic unit and influence the overall system's performance and power dissipation. Discrete cosine transform[6] or Discrete wavelet transform are nonlinear functions which are used in digital signal processing. They can be accomplished by repetitive application of multiplication and addition. Multiplication and addition arithmetics determines the execution speed of the entire calculation. The multiplier requires the longest delay in digital system.

The critical path can be found by multiplier. For high speed multiplication, the modified booth algorithm is commonly used. Due to long critical path for multiplication, the high speed multiplication cannot be completed successfully. A Multiplier uses booth algorithm and array of full adders. Instead of array of full adders, wallace tree can be used. Multiplier using booth algorithm mainly consists of three parts. They are booth encoder, tree, final adder. Booth

encoder is used to generate partial products. Tree is used to compress the partial products such as wallace tree[5]. Tree should add partial products as parallel as possible. The operational time of tree is proportional to  $o(\log_2 N)$ , N is number of inputs. Counting number of inputs reduces the number of outputs into  $\log_2 N$ .

The most effective way to increase the performance of the multiplier is to reduce the number of partial products. Wallace tree is used to increase the speed of adding the partial products, but cannot be used to reduce partial products. To reduce partial products Modified Booth algorithm[2] is used. So many parallel multiplier architectures are developed for increasing the speed of MBA.

There are so many advanced MAC architectures. Among them Elguibaly [3] developed an advanced type of MAC. In this architecture accumulation is combined with the carry save adder tree[3] which is used to compress partial products. In this architecture, critical path is reduced by eliminating the adder for accumulation and decreasing the number of inputs to the final adder. Output rate should be improved due to the use of the final adder for accumulation.

A new architecture for a high speed MAC is proposed. In this MAC, computations of multiplication and accumulation are combined and a hybrid-type CSA structure is proposed to reduce the critical path and to improve the output rate. In CSA, a carry look-ahead adder is used to reduce the number of bits in the final adder. Intermediate calculation results are accumulated in the form of sum and carry instead of the final adder outputs to increase the output rate by optimizing the pipeline efficiency.

This paper is organized as follows. In section II, basic MAC is described. In section III, Modified Booth encoder is described. In section IV, proposed MAC is described. In section V, experiment results are described.

## II. MAC

### A. Introduction to MAC

Basic MAC consists of four steps, Modified Booth encoding, partial product compression, final addition, accumulation Modified Booth encoding is used to generate partial product from multiplicand and multiplier. partial product compression is used to sum all the partial products. final addition is used to give the final multiplication result by adding sum and carry. Final step is accumulation of the multiplied result. General MAC architecture is shown in Fig.

*Manuscript received Mar , 2013.*

*J Y Yaswanth Babu, E.C.E, Karunya University, Coimbatore, India.  
P Dinesh Kumar, E.C.E, Karunya University, Coimbatore, India,*

1. From the figure it is clear that general MAC contains four steps for multiplication and accumulation. The number of partial products is proportional to the number of bits. The time required to add them serially is also proportional to N. In order to increase the speed of multiplication, the number of partial products should be reduced. To reduce the number of partial products, modified Booth encoding, which reduces the partial products by a factor of 2, is used.

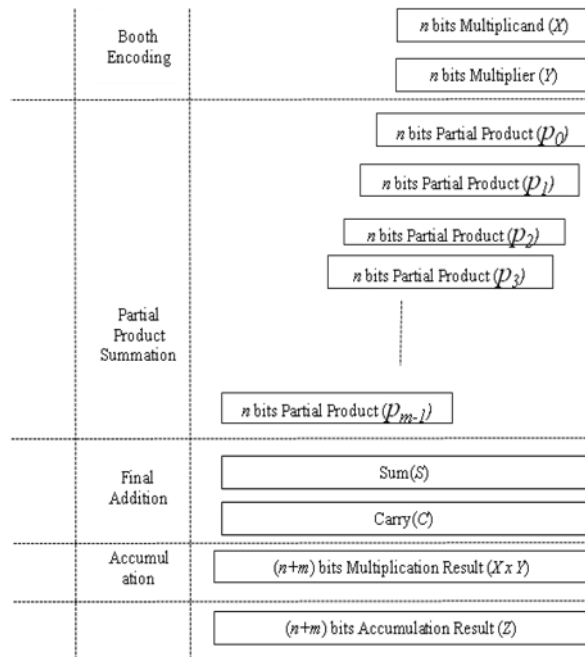


Fig. 1. Basic steps of MAC.

**B. Modified Booth Encoder**

Modified Booth encoder[2] groups multiplier into overlapping group of three bits. Based on the table I, Booth encoder and partial product generation circuits is developed as shown in the Fig. 3. These two circuits are used to choose multiple multiplicands  $-2A, -A, 0, A,$  and  $2A$ . Multiple multiplicands are used to generate partial products. The number of partial products will be reduced to half. Shifting  $A$  left one bit gives  $2A$ . The outputs of Modified Booth encoder,  $neg_i, two_i, one_i, zero_i,$  are the inputs of the partial product generation. Based on the Table I values, the partial product  $p_{i,j}$  is generated.

**C. Proposed MAC Architecture**

The critical path will be proportional to the number of output bits of the multiplication of two N-bit numbers. To improve the performance of the MAC, the delay of the accumulator should be reduced. This can be achieved by applying a pipeline scheme to each step in the MAC. If the accumulator is removed and combined with CSA, overall performance can be improved. The critical path can be determined by final adder if accumulator is removed. If the number of inputs to the final adder is reduced, then the final adder performance will be improved.

TABLE I

MODIFIED BOOTH ENCODING TABLE

$b_{2i+1}$	$b_{2i}$	$b_{2i-1}$	Operation	$neg_i$	$two_i$	$one_i$	$zero_i$
0	0	0	+0	0	0	0	1
0	0	1	+A	0	0	1	0
0	1	0	+A	0	0	1	0
0	1	1	+2A	0	1	0	0
1	0	0	-2A	1	1	0	0
1	0	1	-A	1	0	1	0
1	1	0	-A	1	0	1	0
1	1	1	-0	1	0	0	1

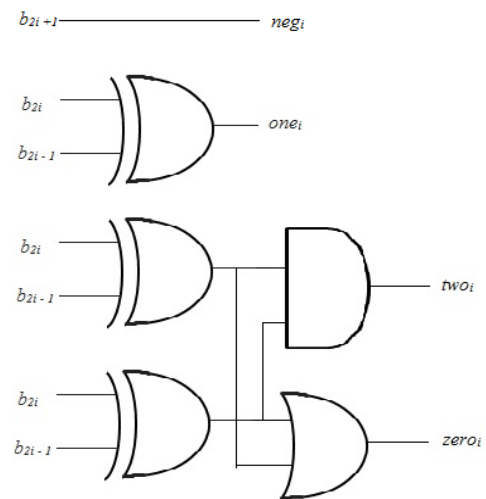


Fig. 2. Modified Booth encoder.

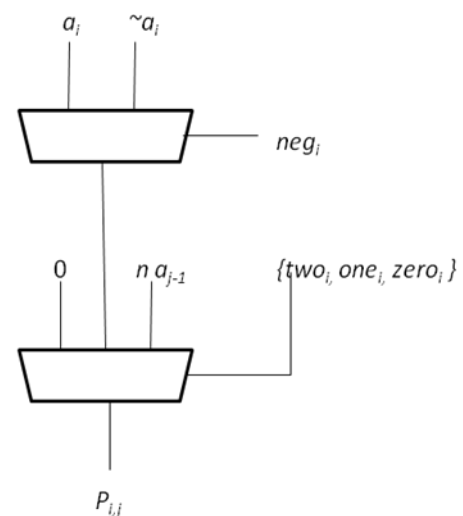


Fig. 3. Partial product generation.

The number of partial products are compressed into sum and carry by CSA. This reduces the number of input bits to the final adder. By adding the number of lower bits of sums and carries will reduce the number of bits of sum and carries that to be transferred to the final adder. The lower bits of the CSA are added by using a 2-bit CLA. Instead of accumulating the output of the final adder, accumulating the sum and carries of the CSA will increase the output rate when the pipelining is applied. The number of inputs to the CSA will be increased due to the feedback of the CSA outputs.

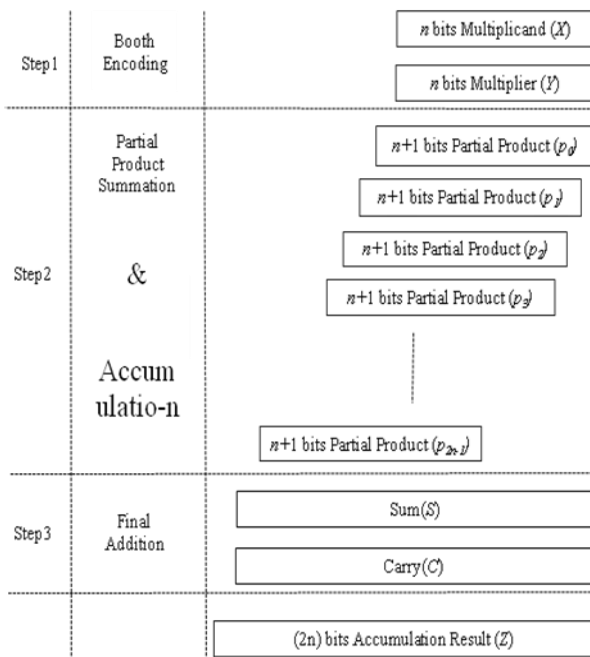


Fig. 4. Arithmetic operation of Proposed MAC

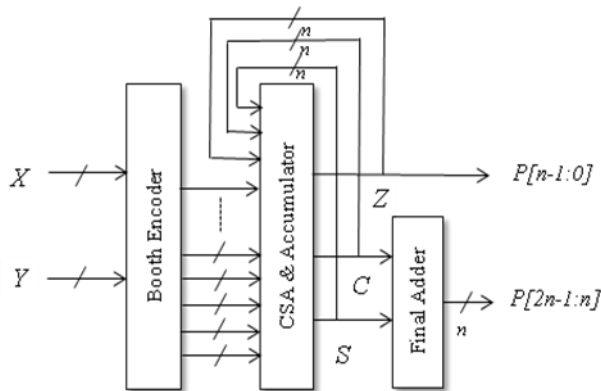


Fig. 5. Hardware architecture of the proposed MAC

The basic MAC is rearranged and a new MAC is proposed as shown in the Fig. 4. The number of steps in the proposed MAC is reduced to three, where the number of steps in basic MAC is four. Partial product summation and accumulation are combined together as shown. The execution time will be less for this type of MAC.

Hardware architecture for proposed MAC is shown in the Fig. 6. X and Y are two N-bit numbers. These two are applied to Modified Booth encoder to generate partial products. As a result (n+1)- bit partial products are generated. These partial products are summed by using CSA. CSA generates n-bit S, C, Z as shown in the Fig.6. S is the sum, C is the carry and Z is the result of adding the lower bits of the sum and carry. These values are feedback to the CSA for next accumulation. The final adder result, P[2n-1:n] is generated by adding the sum and carry. The output generated at Z is P[n-1:0]. Combining both P[2n-1:n] and P[n-1:0] will be the final multiplier result.

D. Proposed CSA Architecture

Proposed CSA architecture is shown in the Fig. 6. In Fig.6.,  $S_i$  is sign expansion and  $N_i$  is to convert 1's complement to 2's complement number. Feedback sum and carry correspond to  $S[i]$  and  $C[i]$  respectively. The sum of the lower bits of all partial products correspond to  $Z[i]$ .  $Z'[i]$  is the previous result.  $P_j[i]$  correspond to partial products. The number of partial products are  $N/2$ . CSA contains four rows of full adders for partial products. One more row of full adder for accumulation. White squares in the Fig. 6. represent full adders and the grey square represent the half adder. The rectangular symbol represents 2-bit CLA. CLA contains five inputs. CLA is the critical path in CSA due to the number of bits. CLA is used to sum the lower bits of partial products. This will reduce the number of inputs to the final adder.

After applying pipeline scheme the proposed MAC will give better performance than the previous structures[3].The pipelined hardware architecture of both the proposed a MAC architecture and existing architecture[3] are shown in the Fig. 7. The difference between existing and proposed structures is, proposed structure contains feedback from CSA. In Existing structure feedback is from final adder.

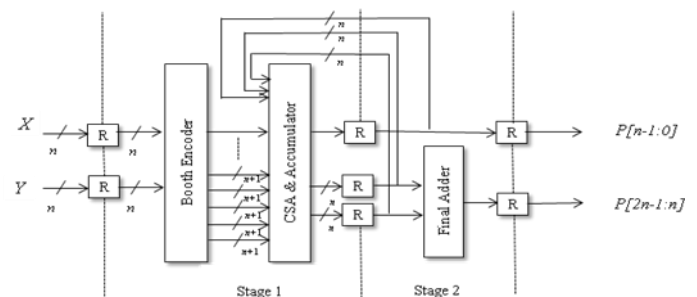


Fig. 6. Pipeline architecture of proposed MAC.

III. EXPERIMENTAL RESULTS

In this section, proposed MAC and existing MAC is synthesized and compared. Table II shows the Delay comparison of proposed and [3] architecture. The Booth encoding technique used is same for both structures. So the delay will be same. Comparing CSA of both structures, from table II it is clear that proposed CSA shows better performance.

TABLE II  
DELAY COMPARISON OF PROPOSED AND [3] ARCHITECTURE

Step	[3]	Proposed
		8 bit
Step1	Booth Encoding 1.3 ns	Booth Encoding 1.3 ns
Step 2	CSA 2.004 ns	CSA 2.004 nss

In the table III, hardware source of both proposed and [3] are compared. The gate size of Full adder and half adder have less area than the adders of [3]. Proposed CSA area is

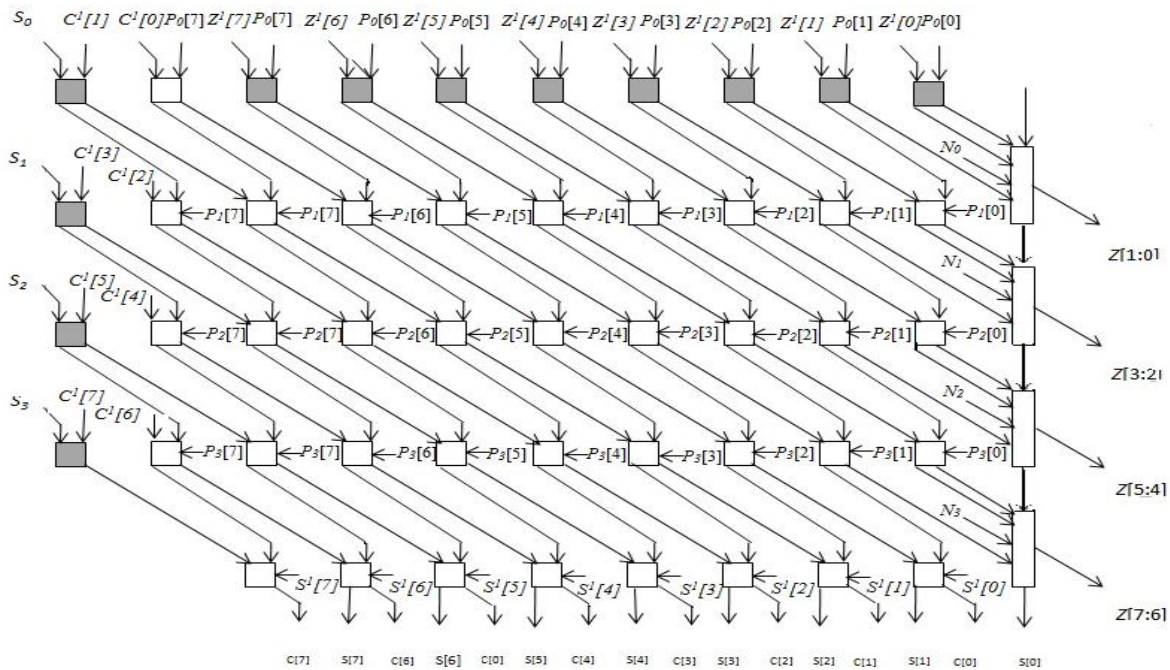


Fig. 6. CSA Architecture.  
less than [3].

TABLE III  
AREA OF PROPOSED AND [3]

Component	Proposed (8 bit)	[3] (8 bit)
FA	1955.92	2165.486
HA	475.675	731.808
CSA	2551.349	3030.35

Now, the proposed is using the modified booth Encoding[1] in the MAC instead of using radix booth encoding. In table IV, pre-layout comparison of both existing and proposed is compared. From table IV, it is clear that performance of proposed MAC is high than existing. Area, power, instances is relatively same.

Post-layout analysis is also done and the layout for 8 bit of proposed MAC is shown in the fig.7.

TABLE IV  
PRE-LAYOUT COMPARISON OF EXISTING AND PROPOSED.

Parameters	Existing	Proposed
Delay	5.5 ns	3.58 ns
Power	0.153 mW	0.76 mW
Area	6603	6091
Instances	280	285

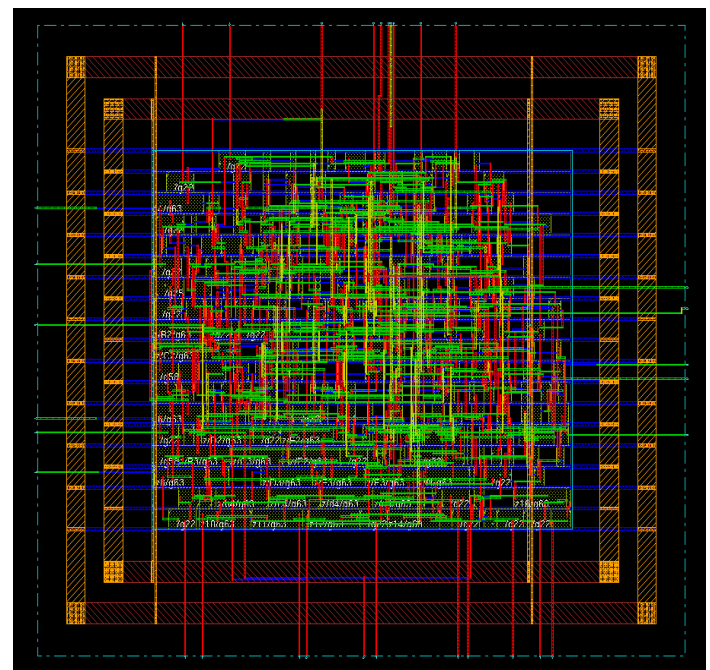


Fig.7. Layout of proposed MAC.

IV. CONCLUSION

A new MAC architecture is proposed for high performance. The proposed architecture is synthesized using cadence SOC Encounter. Area of different stages of proposed architecture is compared. Then from results, it is clear that the proposed architecture shows better performance of 35% than the existing architecture.

ACKNOWLEDGMENT

The authors would like to thank Karunya University for providing effective tool Cadence.

## REFERENCES

- [1] Young-Ho Seo and Dong-Wook Kim, "A New VLSI Architecture of Parallel Multiplier-Accumulator Based on Radix-2 Modified Booth Algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 2, pp 201-208, Feb 2010.
- [2] Jiun-Ping Wang, Shiann-Rong Kuang, " High Accuracy Fixed Width Modified Booth Multipliers for lossy Applications," *IEEE Trans, Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 1, pp 52-60, Jan 2011.
- [3] F. Elguibaly, " A fast parallel multiplier-accumulator using the modified Booth algorithm," *IEEE Trans. Circuits Syst.*, vol. 27, no.9, pp 902-908, Sep. 2000.
- [4] A. D. Booth, " A Signed binary multiplication technique," *Quart. J. Math.*, vol. IV, pp. 236-240, 1952.
- [5] J. J. F. Cavanagh, *Digital Computer Arithmetic*. New York: McGrawHill, 1984.
- [6] A. R. Moondi, *Computer Arithmetic Systems*. Englewood Cliffs, N J: Prentice-Hall, 1994.
- [7] A. R. Cooper, "Parallel architecture modified Booth multiplier," *Proc. Inst. Electr. Eng. G*, vol. 135, pp. 125-128, 1988.
- [8] N. R. Shanbag and P. Juneja, "Parallel implementation of a 4 X 4-bit multiplier using modified Booth's algorithm," *IEEE J Solid- State circuits*, vol.23, no.4, pp. 1010-1013, Aug. 1988.
- [9] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, " A 54 X 54 regular structured tree multiplier," *IEEE J. Solid-State Circuits*, vol. 27, no.9, pp.1229-1236, Sep. 1992.
- [10] J.Fadavi- Ardekani, "M X N Booth encoded multiplier generator using optimized Wallace trees," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1, no.2, pp.120-125, Jun 1993.
- [11] N. Ohkubo, M. Suzuki, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki, and Y. Nakagome, " A 4.4 ns CMOS 54 X 54 multiplier using pass-transistor multiplexer," *IEEE J. Solid-State Circuits*, vol.30, no.3, pp. 251-257, Mar. 1995.
- [12] F. Elguibaly and A. Rayhan, "Overflow handling in inner-product processors," in *proc. IEEE Pacific Rim Conf. Commun., Comput., Signal Process.*, Aug. 1997, pp.117-120.
- [13] A. Fayed and M. Bayoumi, " A merged multiplier-accumulator for high speed signal processing applications," *Proc. ICASSP*, vol. 3, pp. 3212-3215, 2002.
- [14] P. Zicari, S. Perri, P. Corsonello, and G. Cocorullo, " An optimized adder accumulator for high speed MACs," *Proc. ASICON 2005*, vol.2, pp.77-760, 2005.
- [15] T. Sakurai and A.R. Newton, " Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol.25, no.2, pp. 584-594, Feb. 1990.
- [16] K. Choi and M. Song, "Design of a high performance 32 X 32-bit multiplier with a novel sign select Booth encoder," in *proc. IEEE Int. Symp. Circuits and Systems*, 2001, vol. 2, pp. 701-704.
- [17] K. J. Cho, K. C. Lee, J. G. Chung, and K. K. Parhi, " Design of Low-error fixed-width modified Booth multiplier," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 5, pp. 522-531, May 2004.
- [18] M. A. Song, L. D. Van, and S. Y. Kuo, " Adaptive low-error fixed-width Booth multipliers," *IEICE Trans. Fundamentals*, vol. E90-A, no.6, pp. 11180-11187, Jun. 2007.

**J Y Yaswanth Babu** Completed B.Tech and pursuing M.tech in Karunya University, Coimbatore.

**P Dinesh Kumar** completed M.E. and working as Lecturer in Karunya University, Coimbatore.