ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Modified Booth Encoding Multiplier for both Signed and Unsigned Radix Based Multi-Modulus Multiplier

M.Shiva Krushna
M.Tech, VLSI Design,
Holy Mary Institute of Technology And Science,
Hyderabad, T.S, India.

K.Kanthi Kumar,
Associate Professor In Department of ECE,
Holy Mary Institute of Technology & Science,
Hyderabad, T.S, India.

INTRODUCTION

Power dissipation is recognized as a critical parameter in modern VLSI design field. To satisfy MOORE'S law and to produce consumer electronics goods with more backup and less weight, low power VLSI design is necessary.

Fast multipliers are essential parts of digital signal processing systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors today, especially since the media processing took off. In the past multiplication was generally implemented via a sequence of addition,

Subtraction, and shift operations. Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product.

The basic multiplication principle is twofold i.e. evaluation of partial products and accumulation of the shifted partial products. It is performed by the successive additions of the columns of the shifted partial product matrix. The 'multiplier' is successfully shifted and gates the appropriate bit of the 'multiplicand'. The delayed, gated instance of the multiplicand must all be in the same column of the shifted partial product matrix. They are then added to form the product bit for the particular form. Multiplication is therefore a multi operand operation. To extend the multiplication to both signed and unsigned numbers, a convenient number system would be the representation of numbers in two's complement format.

OBJECTIVE OF THE PROJECT

Multiplication operation involves two steps one is producing partial products and adding these partial products. Thus, the speed of a multiplier hardly depends on how fast generate the partial products and how fast we can add them together of partial products to be generated are of less than it is indirectly means that we have achieved the speed in generating partial products. By using Modified Booth's algorithm, to speed up the addition among the partial products by using fast adder architectures and also we can perform the operation on both signed and unsigned numbers

METHODOLOGY

Modified Booth Encoding (MBE) multiplier for both signed and unsigned numbers multiplication. The already existed Modified Booth Encoding multiplier and the Baugh-Wooley multiplier perform multiplication operation on signed numbers only, whereas the array multiplier and Braun array multipliers perform multiplication operation on unsigned numbers only. Thus, the requirement of the modern computer system is a dedicated and very high speed unique multiplier unit for signed and unsigned numbers. Therefore, this topic presents the design and implementation of MBE multiplier. The modified Booth Encoder circuit generates half the partial products in parallel. By extending sign bit of the operands and generating an additional partial product the MBE multiplier is obtained.

DEIGN OF MULTIPLIER-ACCUMULATOR MAC:

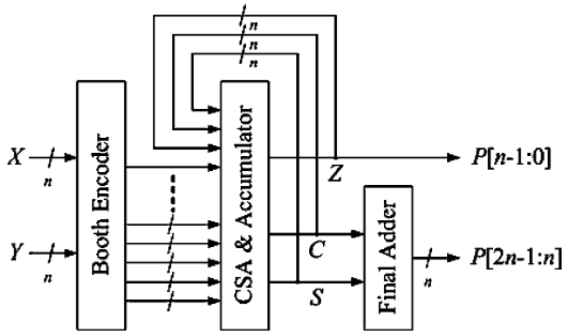
In the majority of digital signal processing (DSP) applications the critical operations are the multiplication and accumulation. Real-time signal processing requires high speed and high throughput Multiplier-Accumulator (MAC) unit that consumes low power, which is always a key to achieve a high performance digital signal processing system. The purpose of this work is, design and implementation of a low power MAC unit with block enabling technique to save power. Firstly, a 1-bit MAC unit is designed, with appropriate geometries that give optimized power, area and delay. The delay in the pipeline stages in the MAC unit is estimated based on which a control unit is designed to control the data flow between the MAC blocks for low power. Similarly, the N-bit MAC unit is designed and controlled for low power using a control logic that enables the pipelined stages at appropriate time. The adder cell designed has advantage of high operational speed, small Gate count and low power.

In general, a multiplier uses Booth's algorithm and array of full adders (FAs), or Wallace tree instead of the array of FA's., i.e., this multiplier mainly consists of the three parts: Booth encoder, a tree to compress the partial products such as Wallace tree, and final adder . Because Wallace tree is to add the partial products from encoder as parallel as possible, its operation time is proportional to, where is the number of inputs. It uses the fact that counting the number of 1's among the inputs reduces the number of outputs into. In real implementation, many (3:2) or (7:3) counters are used to reduce the number of outputs in each pipeline step. The most effective way to increase the speed of a multiplier is to reduce the number of the partial products because multiplication precedes a series of additions for the partial products. To reduce the number of calculation steps for the partial products, MBA algorithm has been applied mostly where Wallace tree has taken the role of increasing the speed to add the partial products. To increase the speed of the MBA algorithm, many parallel multiplication architectures have been researched .Among them, the architectures based on the Baugh–Wooley algorithm (BWA) have been developed and they have been applied to various digital filtering calculations.

PROPOSED MAC ARCHITECTURE

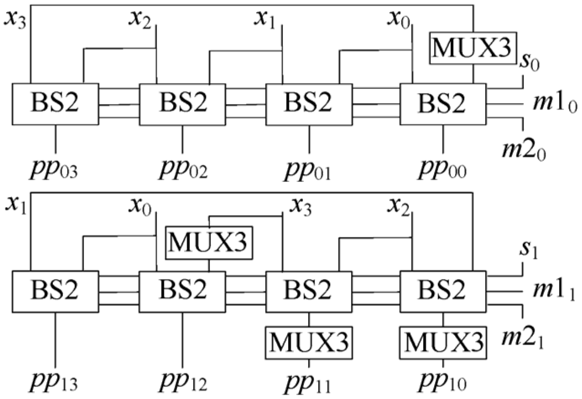
In this section, the expression for the new arithmetic will be derived from equations of the standard design. From this result, VLSI architecture for the new MAC will be proposed. In addition, a hybrid-typed CSA architecture that can satisfy the operation of the proposed MAC will be proposed.

Basic Concept: If an operation to multiply two N –bit numbers and accumulates into a 2N -bit number is considered, the critical path is determined by the 2 -bit accumulation operation. If a pipeline scheme is applied for each step in the standard design of Fig. 1, the delay of the last accumulator must be reduced in order to improve the performance of the MAC. The overall performance of the proposed MAC is improved by eliminating the accumulator itself by combining it with the CSA function. If the accumulator has been eliminated, the critical path is then determined by the final adder in the multiplier. The basic method to improve the performance of the final adder is to decrease the number of input bits. In order to reduce this number of input bits, the multiple partial products are compressed into a sum and a carry by CSA. The number of bits of sums and carries to be transferred to the final adder is reduced by adding the lower bits of sums and carries in advance within the range in which the overall performance will not be degraded. A 2-bit CLA is used to add the lower bits in the CSA. In addition, to increase the output rate when pipelining is applied, the sums and carrys from the CSA are accumulated instead of the outputs from the final adder in the manner that the sum and carry from the CSA in the previous cycle are inputted to CSA. Due to this feedback of both sum and carry, the number of inputs to CSA increases, compared to the standard design and [17]. In order to efficiently solve the increase in the amount of data, CSA architecture is modified to treat the sign bit. The below diagram is the parallel MAC structure. In parallel MAC both partial product addition and accumulation take place at same time.



Parallel MAC unit based on modified booth algorithm

In order to efficiently solve the increase in the amount of data, CSA architecture is modified to treat the sign bit. The below diagram is the parallel MAC structure. In parallel MAC both partial product addition and accumulation take place at same time. The following figure shows the parallel partial product generator for producing the partial products.



Parallel Partial Product Generator

Multiplier and Accumulator Unit

MAC is composed of an adder, multiplier and an accumulator. Usually adders implemented are Carry- Select or Carry-Save adders, as speed is of utmost importance in DSP (Chandrakasan, Sheng, & Brodersen, 1992 and Weste & Harris, 3rd Ed). One implementation of the multiplier could be as a parallel array multiplier. The inputs for the MAC are to be fetched from memory location and fed to the multiplier block of the MAC, which will perform multiplication and give the result to adder which will accumulate the result and then will store the result into a memory location. This entire process is to be achieved in a single clock cycle (Weste & Harris, 3rd Ed). Figure 2 is the architecture of the MAC unit which had been designed in this work. The design consists of one 16 bit register, one 16-bit Modified Booth Multiplier multiplier, 33-bit accumulator using ripple carry and two16-bit accumulator registers. To multiply the values of A and B, Modified Booth multiplier is used instead of conventional multiplier because Modified Booth multiplier can increase the MAC unit design speed and reduce multiplication complexity. Carry save Adder (CSA) is used as an accumulator in this design. Apparently, together with the utilization of Wallace tree multiplier approach, carry save adder in the final stage of the Modified Booth multiplier and Carry save Adder as the accumulator, this MAC unit design is not only reducing the standby power consumption but also can enhance the MAC unit speed so as to gain better system performance. The operation of the designed MAC unit is as in Equation 2.1. The product of Ai X Bi is always fed back into the 34-bit Carry save accumulator and then added again with the next product Ai x Bi. This MAC unit is capable of multiplying and adding with previous product consecutively up to as many as eight times. Operation: Output = Σ Ai Bi (2.1) In this paper, the design of 16x16 multiplier unit is carried out that can perform accumulation on 34 bit number. This MAC unit has 34 bit output and its operation is to add repeatedly the multiplication results. The total design area is also being inspected by observing the total count of gates [Hardwires]. Power delay product is calculated by multiplying the power consumption result with the time delay.

A radix-4 modified Booth's algorithm:

Booth's Algorithm is simple but powerful. Speed of MAC is dependent on the number of partial products and speed of accumulate partial product. Booth's Algorithm provide us to reduced partial products. We choose radix-4 algorithm because of below reasons.

➢ Original Booth's algorithm has an inefficient case.

The 17 partial products are generated in 16bit x 16bit signed or unsigned multiplication.

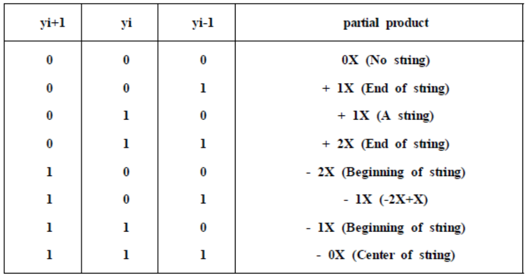
➢ Modified Booth's radix-4 algorithm has fatal encoding time in 16bit x 16bit multiplication.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Radix-4 Algorithm has a 3x term which means that a partial product cannot be generated by shifting. Therefore, 2x + 1x are needed in encoding processing. One of the solution is handling an additional 1x term in wallace tree. However, large wallace tree has some problems too.

A radix-4 modified Booth's algorithm: Booth's radix-4 algorithm is widely used to reduce the area of multiplier and to increase the speed. Grouping 3 bits of multiplier with overlapping has half partial products which improves the system speed. Radix-4 modified Booth's algorithm is shown below:

- X-1 = 0; Insert 0 on the right side of LSB of multiplier.
- Start grouping each 3bits with overlapping from x-1
- If the number of multiplier bits is odd, add a extra 1 bit on left side of MSB
- generate partial product from truth table
- when new partial product is generated, each partial product is added 2 bit left Shifting in regular sequence.



yi+1	yi	yi-1	partial product
0	0	0	0X (No string)
0	0	1	+ 1X (End of string)
0	1	0	+ 1X (A string)
0	1	1	+ 2X (End of string)
1	0	0	- 2X (Beginning of string)
1	0	1	- 1X (-2X+X)
1	1	0	- 1X (Beginning of string)
1	1	1	- 0X (Center of string)

Radix 4 Booth Encoding

x: multiplicand y: multiplier

Sign or zero extension:

Our MAC supports signed or unsigned multiplication and the produced result is 64bit which are stored in 2 special 32bit register. First MAC receives a multiplicand and multiplier but just 16bit operands are signed number in Booth's radix-4 algorithm. Hence, extension bit is required to express 16bit signed number. The core idea of this is that 16bit unsigned number can be expressed by 33bit signed number. The 17 partial products are generated in 33bit x 33bit case (16 partial products in 32bit x 32bit case). Here is an example of signed and unsigned multiplication. When x(multiplicand) is 3bit 111 and y(multiplier) is 3bit 111, the signed and unsigned multiplication is different. In signed case x × y = 1 (-1 x -1 = 1) and in unsigned case x × y = 49 (7 x 7 = 49).

Carry save Adder:

When three or more operands are to be added simultaneously using two operand adders, the time consuming carry propagation must be repeated several times. If the number of operands is ‘k’, then carries have to propagate (k-1) times (Weste & Harris, 3rd Ed). In the carry save addition, we let the carry propagate only in the last step, while in all the other steps we generate the partial sum and sequence of carries separately. A CSA is capable of reducing the number of operands to be added from 3 to 2 without any carry propagation. A CSA can be implemented in different ways. In the simplest implementation, the basic element of carry save adder is the combination of two half adders or 1 bit full adder (Weste & Harris, 3rd Ed)

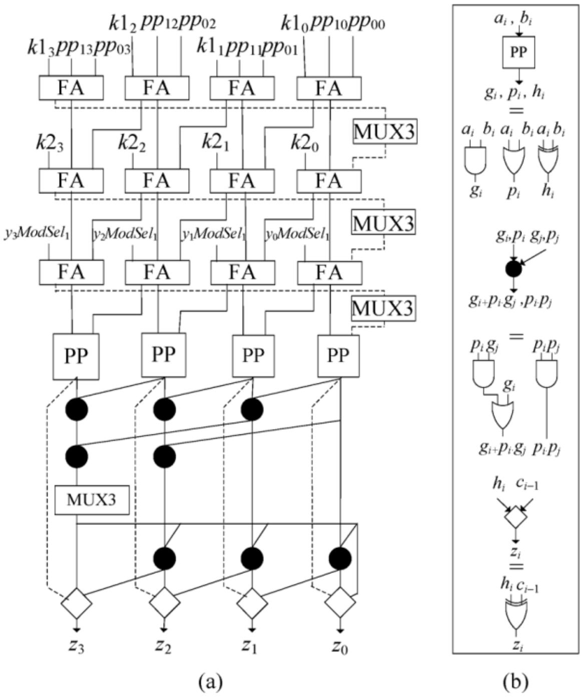
Circuit Design Features:

One of the most advanced types of MAC for general-purpose digital signal processing has been proposed by Elguibaly. It is an architecture in which accumulation has been combined with the carry save adder (CSA) tree that compresses partial products. In the architecture proposed in , the critical path was reduced by eliminating the adder for accumulation and decreasing the number of input bits in the final adder. While it has a better performance because of the reduced critical path compared to the previous MAC architectures, there is a need to improve the output rate due to the use of the final adder results for accumulation. Architecture to merge the adder block to the accumulator register in the MAC operator was proposed to provide the possibility of using two separate N/2-bit adders instead of one -bit adder to accumulate the –bit MAC results. Recently, Zicari proposed an architecture that took a merging technique to fully utilize the 4–2 compressor .It also took this compressor as the basic building blocks for the multiplication circuit.

Block Diagram of MAC:

In this paper, a new architecture for a high-speed MAC is proposed. In this MAC, the computations of multiplication and accumulation are combined and a hybrid-type CSA structure is proposed to reduce the critical path and improve the output rate. It uses MBA algorithm based on 1's complement number system. A modified array structure for the sign bits is used to increase the density of the operands. A carry look-ahead adder (CLA) is inserted in the CSA tree to reduce the number of bits in the final adder. In addition, in order to increase the output rate by optimizing the pipeline efficiency, intermediate calculation results are accumulated in the form of sum and carry instead of the final adder outputs.

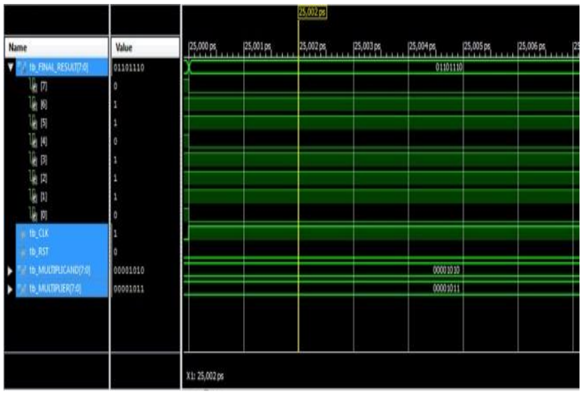
A multiplier can be divided into three operational steps. The first is radix-2 Booth encoding in which a partial product is generated from the multiplicand X and the multiplier Y . The second is adder array or partial product compression to add all partial products and convert them into the form of sum and carry. The last is the final addition in which the final multiplication result is produced by adding the sum and the carry. If the process to accumulate the multiplied results is included, a MAC consists of four steps, as shown in Fig. 1, which shows the operational steps explicitly.



(a) Proposed multi-modulus partial product addition for radix- Booth Encoding.
(b) Implementation of parallel-prefix adder components.

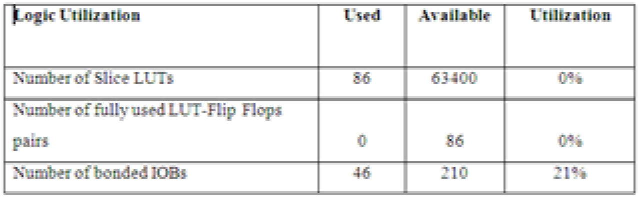
RESULTS

Simulation Results:



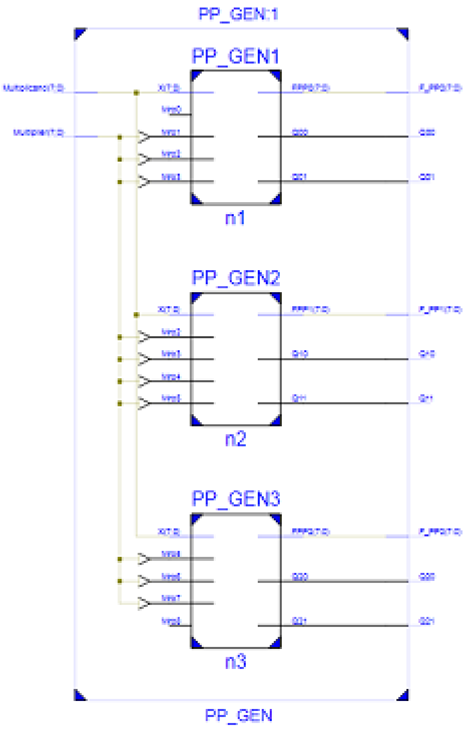
Name	Value
tb_FINAL_RESULT[7:0]	01101110
[7]	0
[6]	1
[5]	1
[4]	0
[3]	1
[2]	1
[1]	1
[0]	0
tb_CLK	1
tb_RST	0
tb_MULTIPLICAND[7:0]	00001010
tb_MULTIPLIER[7:0]	00001011

Area Report:



Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	86	63400	0%
Number of fully used LUT-Flip Flops pairs	0	86	0%
Number of bonded IOBs	46	210	21%

RTL Schematic:



CONCLUSION

In this project, A RADIX 4 Multi modulus Booth multiplier circuit is used for MAC architecture. Compared to other circuits, the Booth multiplier has the highest operational speed and less hardware count. The basic building blocks for the MAC unit are identified and each of the blocks is analyzed for its performance. Power and delay is calculated for the blocks. 1-bit MAC unit is designed with enable to reduce the total power consumption based on block enable technique. Using this block, the N-bit MAC unit is constructed and the total power consumption is calculated for the MAC unit. The power reduction techniques adopted in this work. The MAC unit designed in this work can be used in filter realizations for High speed DSP applications.

REFERENCES

[1]. Shu Lin and Daniel J. Costello. Booth Multiplier Algorithm. Prentice Hall,second edition, 2004.

[2] D. Bakalis and H. T. Vergos, “Area-efficient multi-moduli squarer for RNS ,” in Proc. 13th Euromicro Conf. Digit. Syst. Design, Lille, France, Sep. 2010, pp. 408-411.

[3] N. S. Szabo and R. I.Tanaka, Residue Arithmetic and its Application to Computer Technology. New York: McGraw-Hill, 1967.

[4] C. Efstathiou , K. Pekmestzi, and N. Axelos,“On the design of modulo multipliers ” in Proc. 14th Euromicro Conf. Digit. Syst. Design, Oulu, Finland, Aug. 2011, pp. 453–459.

[5]. W. W. Peterson and E. J. Weldon, Jr., Booth’s Algorithm for Multiplication, 2nd Ed. Cambridge, MA: M.I.T. Press, 1972.

[6] D. Adamidis and H. T. Vergos, “RNS multiplication/sum-of-squares units,” IET Comput. Digit. Tech., vol. 1, no. 1, pp. 38–48, Jan. 2007

[7] C. Efstathiou, H. T. Vergos, and D. Nikolas,“Modified Booth modulo multiplier” IEEE Trans. Comput., vol. 53, no. 3, pp. 370–374, Mar. 2004.

[8] R. Muralidharan and C. H. Chang, “A simple radix-4 Booth encoded modulo multiplier,” in Proc. IEEE Int. Symp. Circuits Syst., Rio de Janeiro, Brazil, May 2011, pp. 1163–1166.

[9] S. Menon and C. H. Chang, “A reconfigurable multi-modulus modulo Multiplier,” in Proc. IEEE Asia-Pacific Conf. Circuits Syst., Singapore, Dec. 2006, pp. 1168–1171.

[10] J. W. Chen, R. H. Yao, and W. J.Wu, “Efficient modulo multipliers,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 12, pp. 2149–2157, Dec. 2011.

[11] R. Muralidharan and C. H. Chang, “Area-power efficient modulo and modulo multipliers for based RNS,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 10, pp. 2263–2274, Oct. 2012.