A Review: Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Encoding

Saumya Sharma[1], Bharti Gupta[2]

[1]*Mtech. Research Scholar,* [1]*Research Guide*

Department of Electronics and Communication, LNCT, Bhopal

Abstract -With the increasing popularity of electronic applications, modular multiplication is becoming one of the important procedures in many computer applications. Because of its computational intensity, implementation in dedicated hardware is required for high-performance systems. Various techniques for speeding up modular multiplication have been reported in literature the basic operation of the modular arithmetic is modular multiplication, which is utilized in many of the computer algorithms and its applications. Thus there is a vital need to develop an efficient algorithms and highspeed hardware to carry out this multiplication.Many algorithms have been proposed for implementing efficient modular multiplication.The radix-4 modular multiplier can be used to implement fast computer applications, e.g RSA cryptosystem and to reduce the number of iterations and pipelining. In this research different various methodologies and algorithm has discussed.The performance of these algorithms is primarily determined by the efficient implementation of the modular multiplication and exponentiation.Discussed a Booth's Radix-2 multiplier and estimated its delay, area and power. A comparison analysis of Radix-2 and Radix-4 algorithm as it seems more suitable for the design by using different adder architectures like RCA and CLA.

Keywords- Multiplying circuits, Modified Booth encoding, Pre-Encoded multipliers, VLSI implementation, FPGA, LUTs.

I. INTRODUCTION

An algebraic field is, by definition, a set of elements that is closed under the ordinary arithmetical operations of addition, subtraction, multiplication, and division. The set of rational numbers is a field, whereas the integers are not a field, because the integers are not closed under the operation of division as the result of dividing one integer by another is not necessarily an integer. It is also possible to construct other fields by means of extending smaller fields.

Day by day IC technology is getting more complex in terms of design and its performance analysis. A faster design with lower power consumption and smaller area is implicit to the modern electronic designs. Unceasing advancement in microelectronics design technology makes improved use of energy, encrypt data successfully, communicate information much more steadfastly, etc. Particularly, many of these technologies address low-power consumption to meet the requirements of various portable applications. In these application systems, a multiplier is a fundamental arithmetic unit and widely used in circuits, for which the multiplication process should be optimized properly. Multipliers generally have extended latency, huge area and consume substantial amount of power. Hence low-power multiplier design has become an important part in VLSI system design. Everyday new approaches are being developed to design low-power multipliers at technological, physical, circuit and logic levels. Since the multiplier is generally the slowest element in a system, the system’s performance is determined by performance of the multiplier. Also multipliers are the most area consuming entity in a design. Therefore, optimizing speed and area of a multiplier is a major design issue nowadays. However, area and speed are usually conflicting constraints so that improving speed results in larger areas and vice-versa. Also area and power consumption of a circuit are linearly correlated. So a compromise has to be done in speed of the circuit for a greater improvement in reduction of area and power.

A higher representation radix effectively indicates to fewer digits. Thus, a single-digit multiplication algorithm necessitates fewer cycles as we start moving to much higher radices, which automatically leads to a lesser number of partial products. Several algorithms have been developed for this purpose like Booth’s Algorithm, Wallace Tree method etc. For the summation process several adder architectures are available viz. Ripple Carry Addition, Carry Look-ahead Addition, Carry Save Addition etc. But to reduce the power consumption the summation architecture of the multiplier should be carefully chosen.

Many algorithms have been proposed for implementing efficient modular multiplication. These algorithms can be classified into the following three categories:

1. Algorithms for general moduli: the classical algorithm, the Barrett algorithm and the Montgomery algorithm.

2. Algorithms for special moduli: modular reduction methods based on pseudo-Mersenne numbers and generalized Mersenne numbers.

3. Look-up table methods: Kawamura, Takabayashi and Shimbo's method; Hong, Oh and Yoon's method; and Lim, Hwang and Lee's method.

Look-up table methods are normally faster than the generalized ones, but require a large size of memory. The Barrett algorithm and the Montgomery algorithm requires small amount of pre-computation. The algorithms using pre-computation are only suitable when some parameters are fixed.

II. SYSTEM MODEL

Multiplying a variable by a set of known constant coefficients is a common operation in many digital signal processing (DSP) algorithms. Compared to other common operations in DSP algorithms, such as addition, subtraction, using delay elements, etc., multiplication is generally the most expensive. There is a trade-off between the amount of logic resources used (i.e. the amount of silicon in the integrated circuit) and how fast the computation can be done. Compared to most of the other operations, multiplication requires more time given the same amount of logic resources and it requires more logic resources under the constraint that each operation must be completed within the same amount of time.

A general multiplier is needed if one performs multiplication between two arbitrary variables. However, when multiplying by a known constant, we can exploit the properties of binary multiplication in order to obtain a less expensive logic circuit that is functionally equivalent to simply asserting the constant on one input of a general multiplier. In many cases, using a cheaper implementation for only multiplication still results in significant savings when considering the entire logic circuit because multiplication is relatively expensive. Furthermore, multiplication could be the dominant operation, depending on the application.

A. Basic binary multiplier

The operation of multiplication is rather simple in digital electronics.

```
          10 x 8 = 80                               -6 x 4 = -24

          1 0 1 0                                    1 0 1 0
          1 0 0 0                                    0 1 0 0
-----------------------------------------------------------------
        0 0 0 0                                    0 0 0 0
      0 0 0 0                                    0 0 0 0
    0 0 0 0                                1 1 1 0 1 0
  1 0 1 0                                  0 0 0 0 0
-----------------------------------------------------------------
  1 0 1 0 0 0 0                            1 1 1 0 1 0 0 0
```

Figure 2.1 Basic binary multiplication algorithms.

It has its origin from the classical algorithm for the product of two binary numbers. This algorithm uses addition and shift left operations to calculate the product of two numbers. Two examples are presented below.

The left example shows the multiplication procedure of two unsigned binary digits while the one on the right is for signed multiplication.. The first digit is called Multiplicand and the second Multiplier. The only difference between signed and unsigned multiplication is that we have to extend the sign bit in the case of signed one, as depicted in the given right example in PP row 3. Based upon the above procedure,

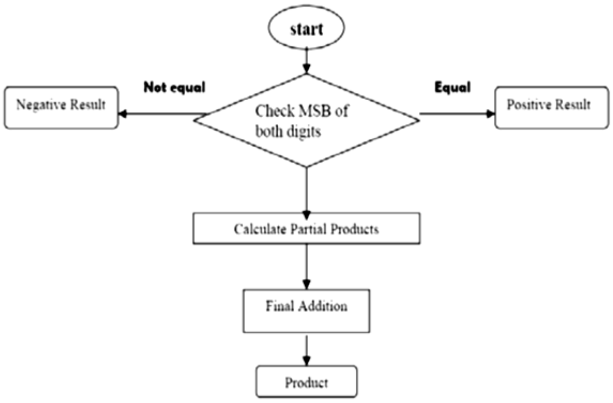

Figure 2.2 Signed multiplication algorithms.

B. Booth Encoding

Booth encoding is a method used for the reduction of the number of partial products proposed by A.D. Booth in 1950. A binary number X consisting of m bits represented in 2's complement format can be described.

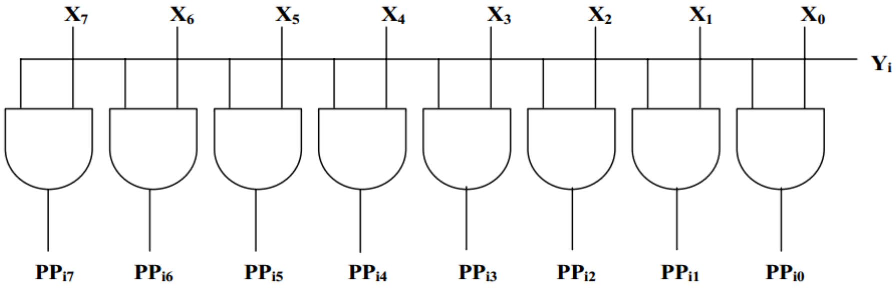

Figure 2.3 Partial product generation logic.

$$X = -2^m X_m + 2^{m-1} X_{m-1} + 2^{m-2} X_{m-2.} \dots \dots \dots \dots \dots \dots \dots Eq.1$$

Rewriting Eq.1 using $2^a = 2^{a+1} - 2^a$ $leds\ to$

$$X = -2^m (X_{m-1} - X_m) + 2^{m-1} (X_{m-2} - X_{m-1}) +_. 2^{m-2} (X_{m-3} - X_{m-2}) +_. \dots \dots \dots \dots \dots \dots \dots Eq.1$$

Considering the first 3 bits of X, we can determine whether to add Y, 2Y or 0 to partial product. The grouping of X bits is shown in Figure 4.

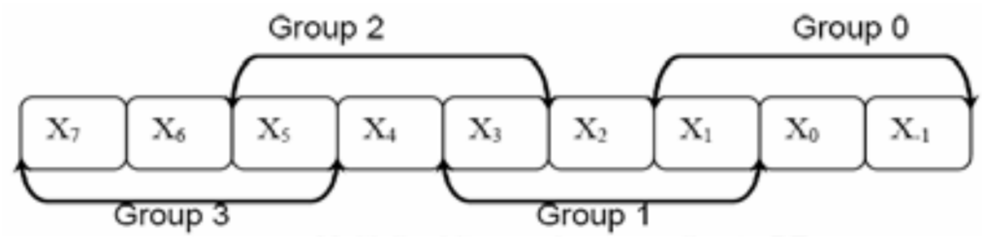

Figure 2.4 Multiplication bit grouping according to booth encoding.

Modified Booth Encoding (MBE)

Modified booth encoding was invented by O.L. Macsorley in 1961. MBE is anenhanced form of Booth encoding.

1.Zero must always be concatenated to the right of X, i.e. x-1 is considered to be 0.

2.M must always be even.

There are two unavoidable consequences when utilizing MBE as sign extension prevention and negative encoding. The combination of these two results in the formation of one additional partial product row, which requires more hardware and the system, also becomes slower [3]. This problem and its solution are discussed in section 6.

The advantage of using MBE is that the number of partial products is reduced to m/2. This, in turn, reduces the hardware burden and increases the speed of multiplier.

III. LITERATURE REVIEW

SR. No.	TITLE	AUTHOR	YEAR	APPROACH
1	Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Signed-Digit Encoding	K. Tsoumanis, N. Axelos, N. Moschopoulos, G. Zervakis and K. Pekmestzi,	2016	Digital signal processing applications based on off-line encoding of coefficients
2	Comparison of regular and tree based multiplier architectures with modified booth encoding for 4 bits on layout level using 45nm technology,	B. Dinesh, V. Venkateshwaran, P. Kavinmalar and M. Kathirvelu,	2014	A detailed analysis of all the serial-parallel and parallel architectures
3	Hybrid modified booth encoded algorithm-carry save adder fast multiplier,	N. G. NikDaud, F. R. Hashim, M. Mustapha and M. S. Badruddin,	2014	A new architecture of hybrid Modified Booth Encoded Algorithm (MBE) and Carry Save Adder (CSA)
4	An efficient fixed width multiplier for digital filter,	S. Nithya and M. N. V. Nithya,	2014	A high speed and low power FIR digital filter design using the fixed width booth multiplier
5	A New Redundant Binary Partial Product Generator for Fast 2n-Bit Multiplier Design,	C. Xiaoping, H. Wei, C. Xin and W. Shumin,	2014	A new radix-16 RB Booth Encoding (RBBE-4) to avoid the hard multiple of high-radix Booth encoding without incurring any ECW
6	High Speed Modified Booth Encoder Multiplier for Signed and Unsigned Numbers,	R. P. Rajput and M. N. S. Swamy,	2012	The design and implementation of signed-unsigned Modified Booth Encoding (SUMBE) multiplier
7	ROM-Based Logic (RBL) Design: A Low-Power 16 Bit Multiplier,	B. C. Paul, S. Fujita and M. Okajima,	2009	A ROM-based 16 times 16 multiplier for low-power applications

K. Tsoumanis, N. Axelos, N. Moschopoulos, G. Zervakis and K. Pekmestzi,[1] In this work, we introduce an architecture of pre-encoded multipliers for digital signal processing applications based on off-line encoding of coefficients. To this extend, the Non-Redundant radix-4 Signed-Digit (NR4SD) encoding technique, which uses the digit values $lbrace -1,0,+1,+2 rbrace$ or $lbrace -2,-1,0,+1 rbrace$, is proposed leading to a multiplier design with less complex partial products implementation. Extensive experimental analysis verifies that the proposed pre-encoded NR4SD multipliers, including the coefficients memory, are more area and power efficient than the conventional Modified Booth scheme.

B. Dinesh, V. Venkateshwaran, P. Kavinmalar and M. Kathirvelu,[2] Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, etc. A system's performance is generally determined by the performance of the multiplier as the multiplier is generally the slowest element in the system. The analysis of performance parameters of different multiplier logics is essential for design of a system intended for a specific function with constraints on Power, Area and Delay. The work presents a detailed analysis of all the serial-parallel and parallel architectures. The multipliers are designed for 4 bit multiplication using DSCH tool and the corresponding layouts are obtained using Microwind 3.5 tool using 45nm technology. From the analysis it is observed that the array multipliers provide a regular routing structure which will be optimum for FPGA based systems. Among the tree based multipliers Dadda multipliers have a slight advantage over Wallace tree multipliers in terms of performance. The Modified booth multiplier is comparatively inefficient for bits lesser than or equal to 4, due to the increased area involved for realization of the booth encoder and booth selector blocks. The analysis shows that for lower order bits Dadda reduction is the most efficient.

N. G. NikDaud, F. R. Hashim, M. Mustapha and M. S. Badruddin,[3] One of the effective ways to speed up multiplication are by reducing the number of partial products and accelerating the accumulation. In this work, a new architecture of hybrid Modified Booth Encoded Algorithm (MBE) and Carry Save Adder (CSA) is developed as fast multiplier architecture. Altera Quartus II platform is used to run the simulation. The architecture design is programmed into FPGA using Altera DE2 board to verify the synthesizability on physical hardware. This hybrid fast multiplier delivers good performance in term of higher speed as well as in term of less usage of logic elements.

S. Nithya and M. N. V. Nithya,[4]We implement a high speed and low power FIR digital filter design using the fixed width booth multiplier. To reduce the truncation error in fixed width multiplier Adaptive Conditional Probability Estimator is used (ACPE). To achieve higher speed, the modified Booth encoding has been used and also to speed up the addition the carry look ahead adder is used as a carry propagate adder. The multiplier circuit is designed using VERILOG and synthesized using Xilinx ISE9.2i simulator. The area, power and delay of the designed filter is analysed using cadence tool.

C. Xiaoping, H. Wei, C. Xin and W. Shumin,[5] The radix-4 Booth encoding or Modified Booth encoding (MBE) has been widely adopted in partial products generator to design high-speed redundant binary (RB) multipliers. Due to the existence of an error-correcting word (ECW) generated by MBE and RB encoding, the RB multiplier generates an additional RB partial product rows. An extra RB partial product accumulator (RBPPA) stage is needed for 2n-b RB MBE multiplier. The higher radix Booth algorithm than radix-4 can be adopted to reduce the number of partial products. However, the Booth encoding is not efficient because of the difficulty in generating hard multiples. The hard multiples problem in RB multiplier can be resolved by difference of two simple power-of-two multiples. This work presents a new radix-16 RB Booth Encoding (RBBE-4) to avoid the hard multiple of high-radix Booth encoding without incurring any ECW. The proposed method leads to make high-speed and low-power RB multipliers. The experimental results show that the proposed RBBE-4 multiplier achieves significant improvement in delay and power consumption compared with the RB MBE multiplier and the current reported best RBBE-4 multipliers.

R. P. Rajput and M. N. S. Swamy,[6] This work presents the design and implementation of signed-unsigned Modified Booth Encoding (SUMBE) multiplier. The present Modified Booth Encoding (MBE) multiplier and the Baugh-Wooley multiplier perform multiplication operation on signed numbers only. 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 work presents the design and implementation of SUMBE 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 SUMBE multiplier is obtained. The Carry Save Adderr (CSA) tree and the final Carry Look ahead (CLA) adder used to speed up the multiplier operation. Since signed and unsigned multiplication operation is performed by the same multiplier unit the required hardware and the chip area reduces and this in turn reduces power dissipation and cost of a system.

B. C. Paul, S. Fujita and M. Okajima,[7] We present a ROM-based 16 times 16 multiplier for low-power applications. The design uses sixteen 4 times 4 ROM-based multiplier blocks followed by carry-save adders and a final carry-select adder (all ROM-based) to obtain the 32 bit output. All ROM blocks are implemented using single transistor ROM cells and eliminating identical rows and columns for optimizing the power and performance. Measurement results in 0.18 mum CMOS process show a 40% reduction in power over the conventional carry-save array multiplier when operated at its maximum frequency. The ROM-based design also provides 44% less delay than the array multiplier with a minimal increase (7.7%) in power. This demonstrates the low-power operation of the ROM-based multiplier also at higher frequencies.

IV. PROBLEM STATEMENT

One of the many solutions of realizing high speed multipliers is enhancing parallelism which helps in decreasing the number of subsequent calculation levels. The original version of Booth algorithm (Radix-2) had two particular drawbacks. They were:

- The number of add-subtract operations and shift operations become variable and causes inconvenience in designing parallel multipliers.
- The algorithm becomes inefficient when there are isolated 1’ s.

These problems are overwhelmed by using modified Radix4 Booth algorithm which scans strings of three bits using the algorithm given below:

1. Lengthen the sign bit 1 position if necessary to ensure that n is even.
2. Add a 0 to the right of the LSB of the multiplier.
3. Corresponding to the value of each vector, each Partial Product will be 0, +M, -M, +2M or -2M.

V. CONCLUSION

After going through all the literature survey and review of past work and after facing a lot of problems in previous base work, we are able determine the objectives of the research work that are to implement Booth’s Algorithm for the design of a binary multiplier using different architectures and power analysis at various levels.To analyze the area and the time delay which consumed by different adders and found out an appropriate relationship among the time and area complexity the adders which we have taken into consideration? After comparing all it can beconcluded that no of bits changes are best suited for Low Power Applications. Then the research turned focus toward the area and delay of Multipliers. Further work can be done on design a Radix-4 multiplier and estimated its delay, area.

REFERENCES

[1] K. Tsoumanis, N. Axelos, N. Moschopoulos, G. Zervakis and K. Pekmestzi, "Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Signed-Digit Encoding," in IEEE Transactions on Computers, vol. 65, no. 2, pp. 670-676, Feb. 1 2016.

[2] B. Dinesh, V. Venkateshwaran, P. Kavinmalar and M. Kathirvelu, "Comparison of regular and tree based multiplier architectures with modified booth encoding for 4 bits on layout level using 45nm technology," 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), Coimbatore, 2014, pp. 1-6.

[3] N. G. NikDaud, F. R. Hashim, M. Mustapha and M. S. Badruddin, "Hybrid modified booth encoded algorithm-carry save adder fast multiplier," The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M), Kuching, 2014, pp. 1-6.

[4] S. Nithya and M. N. V. Nithya, "An efficient fixed width multiplier for digital filter," 2014 IEEE 8th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, 2014, pp. 96-102.

[5] C. Xiaoping, H. Wei, C. Xin and W. Shumin, "A New Redundant Binary Partial Product Generator for Fast 2n-Bit Multiplier Design," 2014 IEEE 17th International Conference on Computational Science and Engineering, Chengdu, 2014, pp. 840-844.

[6] R. P. Rajput and M. N. S. Swamy, "High Speed Modified Booth Encoder Multiplier for Signed and Unsigned Numbers," 2012 UKSim 14th International Conference on Computer Modelling and Simulation, Cambridge, 2012, pp. 649-654.

[7] B. C. Paul, S. Fujita and M. Okajima, "ROM-Based Logic (RBL) Design: A Low-Power 16 Bit Multiplier," in IEEE Journal of Solid-State Circuits, vol. 44, no. 11, pp. 2935-2942, Nov. 2009.

[8] G. W. Reitwiesner, "Binary arithmetic," Advances in Computers, vol. 1, pp. 231-308, 1960.

[9] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. John Wiley & Sons, 2007.

[10] K. Yong-Eun, C. Kyung-Ju, J.-G. Chung, and X. Huang, "Csd- based programmable multiplier design for predetermined coefficient groups," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. 93, no. 1, pp. 324-326, 2010.

[11] O. Macsorley, "High-speed arithmetic in binary computers," Proc. IRE, vol. 49, no. 1, pp. 67-91, Jan. 1961.

[12] W.-C. Yeh and C.-W. Jen, "High-speed booth encoded parallel multiplier design," IEEE Trans. Comput., vol. 49, no. 7, pp. 692-701, Jul. 2000.

[13] Z. Huang, "High-level optimization techniques for low-power multiplier design," Ph.D. dissertation, Department of

Com¬puter Science, University of California, Los Angeles, CA, 2003.

[14] Z. Huang and M. Ercegovac, "High-performance low-power left-to-right array multiplier design," IEEE Trans. Comput., vol. 54, no. 3, pp. 272-283, Mar. 2005.

[15] Y.-E. Kim, K.-J. Cho, and J.-G. Chung, "Low power small area modified booth multiplier design for predetermined coeffi¬cients," IEICE Trans. Fundam. Electron. Commun. Comput. Sci., vol. E90-A, no. 3, pp. 694-697, Mar. 2007.