2nd International Conference on Emerging Trends in Engineering, Technology, Science and Management
Lakshmi Narain College of Technology, Bhopal
20th May 2017, www.conferenceworld.in
(ICETETSM-17)
ISBN: 978-93-86171-45-0

A REVIEW: AREA AND DELAY EFFICIENT PRE-ENCODED MULTIPLIERS BASED ON NON-REDUNDANT RADIX-4 ENCODING

Saumya Sharma[1], Bharti Gupta[2]

[12]*Department of Electronics and Communication Engineering, LNCT Bhopal (M.P), (India)*

ABSTRACT

In this paper,we introduce an architecture of pre-encoded multiplier. 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. 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: Modified Booth encoding, Pre-Encoded multipliers, VLSI implementation.

I . INTRODUCTION

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. 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.

ISBN: 978-93-86171-45-0

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

Figure1.Basic binary multiplication algorithms.

(ICETETSM-17)
ISBN: 978-93-86171-45-0

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. Booth's Multiplication Algorithm Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. The algorithm was invented by Andrew Donald Booth in 1950 while doing research on crystallography at Birkbeck College in Bloomsbury, London. Booth used desk calculators that were faster at shifting than adding and created the algorithm to increase their speed.

B) A Typical Implementation

Booth's algorithm can be implemented by repeatedly adding (with ordinary unsigned binary addition) one of two predetermined values A and S to a product P, then performing a rightward arithmetic shift on P. Let m and r be the multiplicand and multiplier, respectively; and let x and y represent the number of bits in m and r.

1) Determine the values of A and S, and the initial value of P. All of these numbers should have a length equal to (x + y + 1).

a) A: Fill the most significant (leftmost) bits with the value of m. Fill the remaining (y + 1) bits with zeros.

b) S: Fill the most significant bits with the value of (−m) in two's complement notation. Fill the remaining (y + 1) bits with zeros.

c) P: Fill the most significant x bits with zeros. To the right of this, append the value of r. Fill the least significant (rightmost) bit with a zero.

2) Determine the two least significant (rightmost) bits of P.

a) If they are 01, find the value of P + A. Ignore any overflow.

b) If they are 10, find the value of P + S. Ignore any overflow.

c) If they are 00, do nothing. Use P directly in the next step.

d) If they are 11, do nothing. Use P directly in the next step.

3) Arithmetically shift the value obtained in the 2nd step by a single place to the right. Let P now equal this new value.

4) Repeat steps 2 and 3 until they have been done y times.

5) Drop the least significant (rightmost) bit from P. This is the product of m and r.

The representations of the multiplicand and product are not specified; typically, these are both also in two's complement representation, like the multiplier, but any number system that supports addition and subtraction will work as well. As stated here, the order of the steps is not determined. Typically, it proceeds from LSB to MSB, starting at i = 0; the multiplication by 2i is then typically replaced by incremental shifting of the P accumulator to the right between steps; low bits can be shifted out, and subsequent additions and subtractions can then be done just on the highest N bits.

ISBN: 978-93-86171-45-0

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

[A] 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.

ISBN: 978-93-86171-45-0

[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.

[C] 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.

[D] 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 isanalysed using cadence tool.

[E] 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

(ICETETSM-17)
ISBN: 978-93-86171-45-0

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.

[F]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.

[G] 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

ISBN: 978-93-86171-45-0

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 be concluded 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 Basedon 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.