European Journal of Scientific Research
ISSN 1450-216X Vol.26 No.3 (2009), pp.362-368

http://www.eurojournals.com/ejsr.htm

# A New Multiplication Algorithm Using High-Speed Counters

**P. Assady**
*Islamic Azad University Varameen branch, Iran*
E-mail: assady2@gmail.com

**Abstract**

In this paper, a new high-speed multiplication algorithm has been presented. A multiplier has three important steps, which include partial product generation step, partial product reduction step, and final addition step. In partial product generation step, a new Booth algorithm has been presented. In partial product reduction step, a new tree structure has been designed and in final addition step, a new hybrid adder using 4-bit blocks has been proposed. A new partial product reduction algorithm using counter architecture is designed. A novel full-adder has been proposed. Simulations have been done with Spice and C codes. The proposed multiplier has 13 percent reduction in transistor count, 11 percent improvement in power consumption and 10 percent delay modification in compare with other multiplication algorithms.

**Keywords:** Adder, Booth Algorithm, CMOS, Multiplier

## 1. Introduction

Multipliers have an important effect in designing arithmetic processors. In recent years, high-speed multipliers have important role for microprocessors architecture and much investment has been dedicated to design new structures [1,2,3]. Many algorithms for designing high-speed multipliers have been modified and developed [4,5]. Some modified Booth algorithms have been presented. New array structures have been proposed. Number of hybrid adders was developed. Some CMOS techniques such as pass transistor logic and differential cascade voltage switch logic have been designed [6,7]. While the Booth method decreases the inner products, new CMOS techniques achieve high-speed blocks. For implementing these algorithms, full custom design algorithm was preferred to standard ones. These, however, have some problems with flexibility of design and much time to implement. In a try to solve these problems, modeling in hardware description languages and synthesis are researched.

This paper describes a low-power and high-speed multiplier suitable for DSP applications [8,9]. So, a high speed Booth algorithm, a full-adder, and 32-bit hybrid adder based on a carry select adder as components is designed. The high-speed Booth algorithm, speed-up the modified Booth algorithm and decreases the number of gates near 20% in compare with to the CMOS one [5,6,8]. Further, a new partial reduction algorithm using full-adders are presented to reduce time. Finally, ten carry select sum components are used to design a 32-bit hybrid adder.

## 2. Modified Booth Algorithm

To handle the implementing of signed numbers, a new algorithm with Booth recoding method are presented, assume X is the n-bit multiplicand, Y is the m-bit multiplier and P is the product of X and Y, then

$$X = -x_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i$$

$$Y = -y_{m-1}2^{m-1} + \sum_{i=0}^{m-2} y_i 2^i$$

$$P = \left( x_{n-1}y_{m-1}2^{m+n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{m-2} x_i y_j 2^{i+j} \right) - \left( \sum_{i=0}^{m-2} x_{n-1}y_i 2^{n-1+i} + \sum_{i=0}^{n-2} x_i y_{m-1} 2^{m-1+i} \right)$$

With changing expressions above

$$P = \left( x_{n-1}y_{m-1}2^{m+n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{m-2} x_i y_j 2^{i+j} \right) + ([E1] + [E2])$$

Where expressions for E1 and E2 can be showed as

$$E1 = \sum_{i=0}^{m-2} x_{n-1}y_i 2^{n-1+i} = -1 \cdot 2^{m+n-1} + 1 \cdot 2^{m+n-2} + \sum_{i=0}^{m-2} \overline{x_{n-1}y_i} 2^{n-1+i} + 1$$

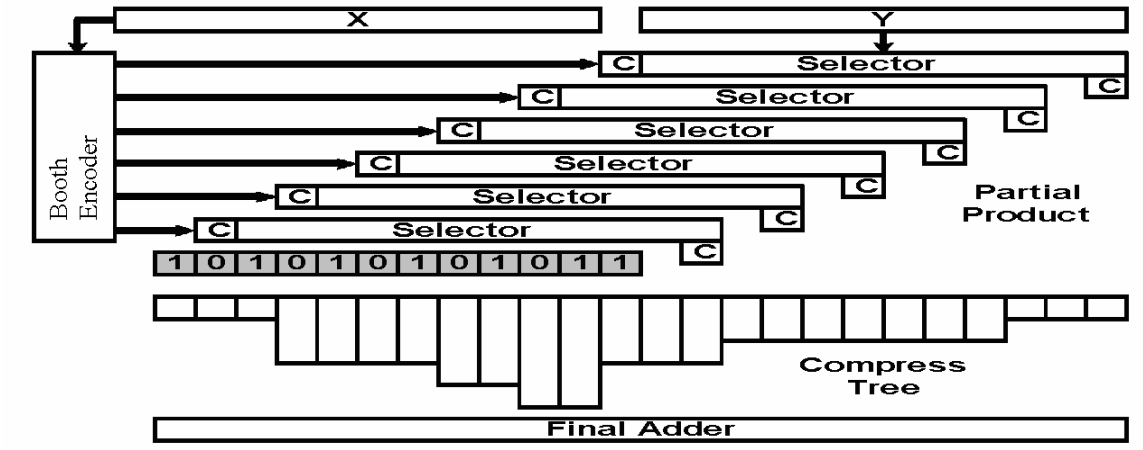
$$E2 = \sum_{i=0}^{n-2} x_i y_{m-1} 2^{m-1+i} = -1 \cdot 2^{m+n-1} + 1 \cdot 2^{m+n-2} + \sum_{i=0}^{n-2} \overline{x_i y_{m-1}} 2^{m-1+i} + 1$$

Allowing for the Booth encoding and sum up the necessary bits in each row of partial product, the number necessary to be summed in inner products can be calculated as follows

$$signs = \sum_{i=0}^{\frac{m}{2}-1} ((-1)2^n)4^i = 2^n(-1)\left(\frac{2^m - 1}{3}\right)$$

Figure 1 shows presented method with Booth algorithm structure. The large numbers of bits in this part of the inner product bits are substituted by only a few bits. Also, the area is decreased.

**Figure 1:** The structure of inner product generation using the novel algorithm with Booth technique

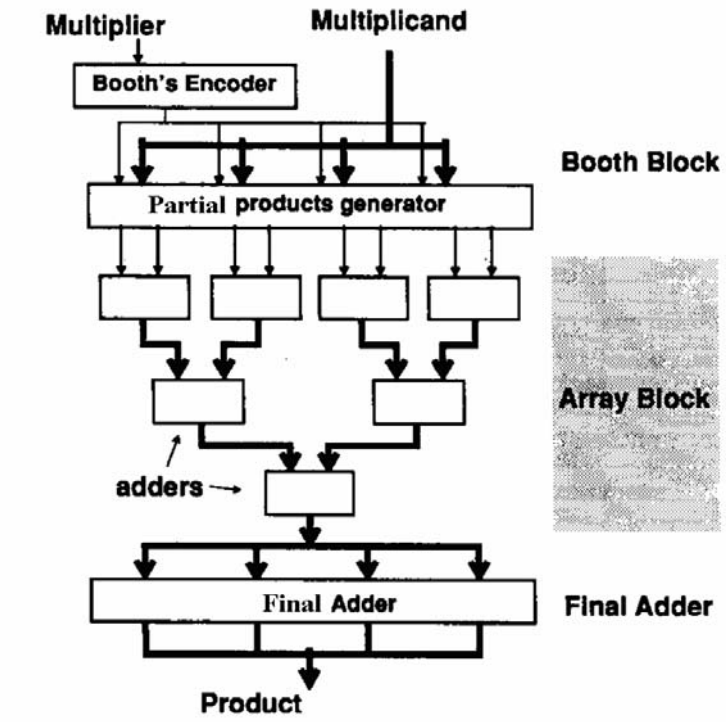


## 3. Block Diagram of the Multiplier

The multiplier architecture is showed in Figure 2. The multiplier uses three components: the Booth algorithm, the array structure, and the final hybrid adder [10,11]. The Booth algorithm produces inner products using new structure. The array structure sums inner products from the Booth algorithm, and produces two 32-b operands for final addition. In the array structure, the Wallace binary algorithm method was used to decrease time needed. The hybrid adder is a 32-b carry select adder and produces the result.

In presented multiplier, to obtain speed both the Booth structure and the array algorithm are used. Furthermore, in order to better structure, a new adder, has been designed as an array component instead of conventional full adders which are commonly used for multiplier structures. Transferring to 17*17-bit multiplier has been done. Simulations show appropriate modification for multiplication algorithm in compared with other multipliers. We used SPICE and C codes for simulations.

**Figure 2:** Structure of the multiplier



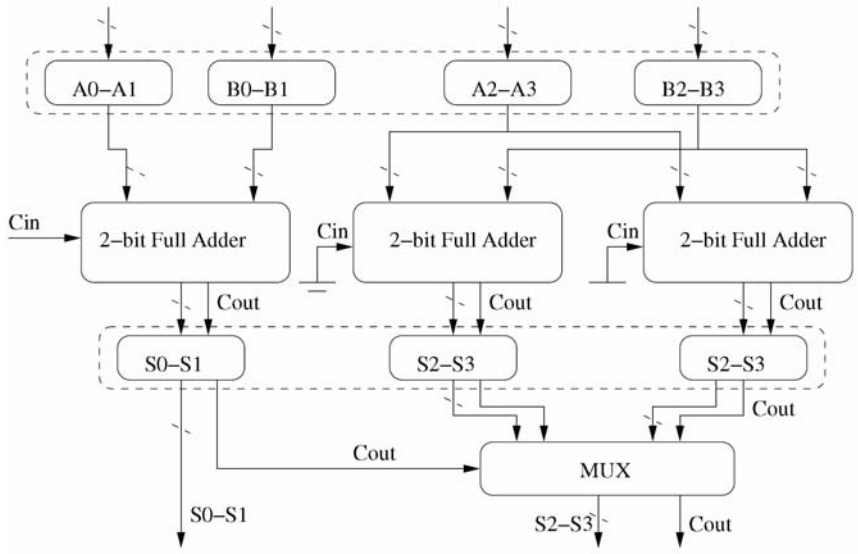

## 4. Carry Hybrid Adder Design

Designing of full adders have two important structures. Adder architecture is the first structure. Carry-ripple makes the worst path in adder structure. Many algorithms have been designed for modification the carry ripple path. Most used adder structures are chain carry adder, carry skip adder, carry hybrid adder and prefix adder. Each adder has its advantages and disadvantages. Usually, obtaining high-speed circuits need bigger and more complicated circuits. There are trade off between speed and complication in these algorithms. These adder structures have different electronic characteristics. Some are suitable for delay time and others are suitable for power consumption. In this design 4-bit adder structure has been developed, carry-hybrid adder is used as the base structure. This kind of adder is usually designed for high-speed structures, where power consumption is not important. 1-bit full adder architecture is the second structure. Different implementations for CMOS techniques are obtained for full adder architecture. Problems here are high-speed production carry output that can be achieved by the next step.

The structural chart of the carry-hybrid adder is presented in Figure 3. This algorithm has advantages of carry select adder, by doing all the calculations in one cycle and this can be done from the least bit carry in to the next step. Because, the delay of the multiplexer is low, extra logic can be used after multiplexer within the next step, letting additions in less time.

Because already the architecture of 4-bit carry-hybrid adder is implemented, the structure of 2-bit full adders is very important for performance and minimum area needs of the complete structure. The architecture of 2-bit adder can also be into two modules, implementing of 1-bit adder and design of carry-ripple.

**Figure 3:** Architecture of 4-bit carry hybrid adder




## 5. New Full Adder

Full adder has an important role in designing big adders such as the chain carry, carry prefix, and carry hybrid adders and other arithmetic operators [10,11,12]. Full adder has an important effect in performance of multiplier implementation. The full adder inputs are the two one bit digit to be added, A and B, and the carry input $C_i$ coming from the digits computations. The outputs are the outcome of the sum operation S and the carry out $C_0$, showed by

$$S = A \oplus B \oplus C_i = A\overline{BC_i} + \overline{A}B\overline{C_i} + \overline{AB}C_i + ABC_i$$

$$C_o = AB + (A + B)\,C_i$$

From which $C_o = C_i$ if $A \neq B$, and in this form, the full adder is in the spread form. The formerly adder structures are designed by chaining full adders, each of full adders has its carry related to the carry input of the next one, as shown in Figure 4. The resulting delay calculates from the carry propagation (i.e., with all full adders in the spread form) from all full adders, as shown in Figure 4 by the path in the line. So, the worst delay is as like as to the product of the carry input to carry output delay $\tau_{carry}$ of the full adder, and it is enough to know the differences in $\tau_{carry}$ to calculate the adder delay.

To calculate the delay of full adder an input signal is used and simulation with programming code is done. Because from Figure 4, each full adder is motivated by a unit, the circuit structure to compute $\tau_{carry}$ is reported in Figure 4. The full adder structure in Figure 5 is designed with the CMOS technique.

**Figure 4:** Structure to compute the carry propagation delay $\tau_{carry}$ of the adder

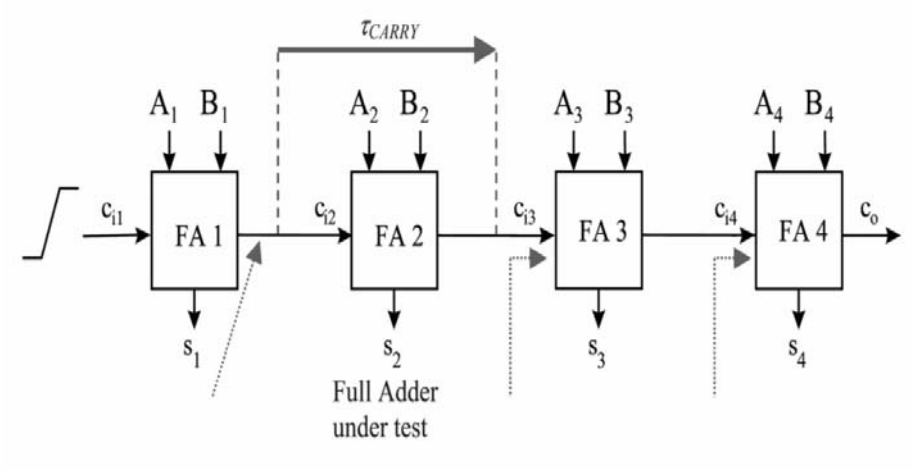
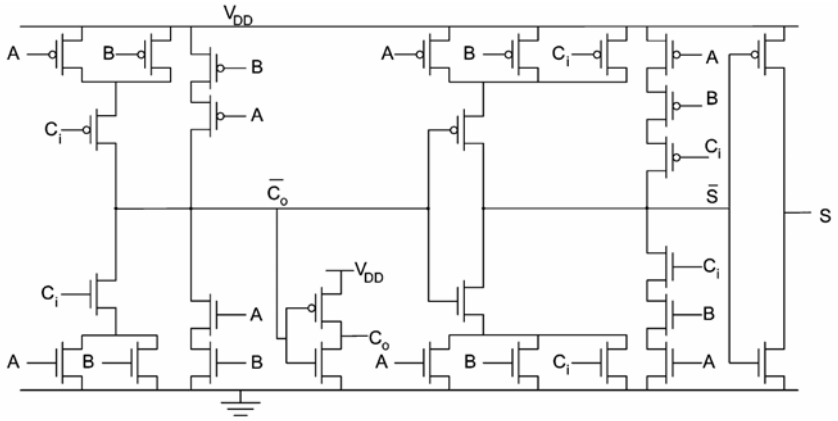



**Figure 5:** Adder structure

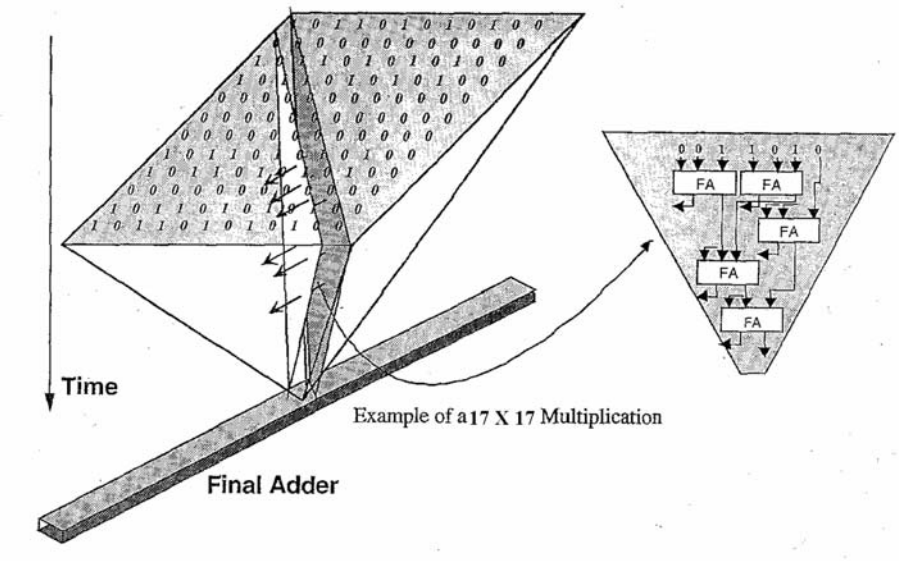



## 6. Partial Product Reduction Algorithm

A new partial product reduction algorithm using counter architecture and partial product reduction structure is designed. In this algorithm, the entire multiplier structure was synthesized entirely. The counter includes of a part where the partial product bit structure is showed in decreasing stages. This section in our design showed in Figure 6, presents the partial product counter of the multiplier structure. This section in this form uses the counters that are processed in the reduction procedure and, therefore, shows a counter architecture to the suitable bit place. It will be used as a counter part. It should be noticed that there are a number of input signals into the part and a number of output signals into the vertical part and a number of output signals obtaining from the part which are then being passed to the next counter part relating to the first higher order bit position.

In Figure 6, decreasing matrix of partial products by a tree of full adder using one counter part has been shown. On the other hand, each full adder generates a carry out which goes the part of next stage. Therefore, the worst path is not only a path through a given part, but is also a path through the parts. As previously mentioned, the counter decreases the path. The aim of the following method is to decrease both paths by building parts that are optimized for a minimal delay. An algorithm for designing counter parts by using critical paths is presented. In this paper, ideas from previous Booth algorithms, counters, final adder and partial product reduction techniques have been used and a new multiplier has been presented that has better electronic parameters in compare with other conventional multipliers.

**Figure 6:** Partial product reduction structure




## 7. Conclusions

The multiplier is an important part in arithmetic processors. Especially, the current mobile applications need ICs with high-speed operators and low power consumption. A new high-speed multiplier has been presented. In partial product generation step, a new Booth algorithm has been proposed. In partial product reduction step, a new tree structure has been modified and in final addition step a new hybrid adder using 4-bits blocks has been presented. Proposed Booth algorithm has important effect on multiplier performance. The modified algorithm and structure for multiplier implementation takes advantage of new adder circuits in order to make a universal counter that decreases the critical path of the multiplier. The reduction tree is used into vertical parts that are optimized entirely to generate counter parts. This minimization does not only include the signal path, but also includes the signals from the former counter parts. A novel full-adder has been presented which reduces critical path of partial product reduction step. This project has been implemented using C language and Spice simulation. For comparison, other multiplication algorithms have been implemented in 17*17-bit scale. Simulations have been done in 90-nm CMOS process. The proposed 17*17-bit multiplier has 11552 transistors, delay time of multiplication process is 3.2 ns and power consumption is 124 mWatt. The proposed multiplier has 13 percent reduction in transistor count, 11 percent improvement in power consumption and 10 percent delay modification in compare with other multiplication algorithms [8,9,10,11,12].

## References

[1] Sakuta, T., W. Lee and P.T. Balsara, "Delay balanced multipliers for low-power/low-voltage DSP core", in Proc. 1995 IEEE Symp. Low Power Electronics, pp. 36-40, Oct. 1995.

[2] Callaway, E. and E.E. Swartzlander, "Power-delay characteristics of CMOS multipliers", in Proc. 13th Int. Symp. Computer Arithmetic, pp. 26-30, 1997.

[3] Oliveri, M., "Design of synchronous and asynchronous variable-latency pipelined multipliers", IEEE Transactions on VLSI systems, vol. 9, no. 2, pp. 365-376, April 2001.

[4] Kim, K., P. Beerel, "A synchronous matrix-vector multiplier for discrete cosine transform", in international symposium on low power electronics and design, pp. 256-261, July 2000.

[5] Tang, T.Y., C.S. Choy, P.L. Siu and C.F. Chan, "Design of self-timed asynchronous Booth's multiplier", in Proc. Asia South Pacific Design Automation Conf., pp. 15-16, Jan 2000.

[6] Bui H.T., Y. Wang, and Y. Jiang, "Design and analysis of low-power 10-transistor full adders using XOR-XNOR gates", IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process, vol. 49, no. 1, pp. 25-30, Jan. 2002.

[7] Sunder S., F. Elguibaly and A. Antoniou, "Two's complement fast serial-parallel multiplier", Proc. IEE Circuits Devices Syst., vol. 142, pp. 41-44, 1995.

[8] Gok, M., M.J. Schulte and M.G. Arnold, "Integer multipliers with overflow detection", IEEE Trans. on Computers, vol. 55, no. 8, pp. 1062-1066, August 2006.

[9] Bewick, G. and M. Flynn, "Binary multiplication using partially redundant multiples", Technical report No. CSL-TR-92-528, Computer Systems Laboratory, Stanford Univ., June 1992.

[10] Fahmi, M.N., F. Elguibaly, E. Abdel-raheem, and A Tawfik, "Area-time efficient fixed-point multiplier-accumulators for inner-product computation", in Proc. IEEE Int. Conf. Microelectronics, Dhahran, Saudi Arabia, pp. 189-192, Dec, 1999.

[11] Kim, S., C.H. Ziesler, and M.C. thymiou, "Design, verification, and test of a true single-phase 8-bit adiabatic multiplier", in Proc. 19th Conf. Advances Research VLSI, Salt Lake City, UT, pp. 42-58, Mar. 2001.

[12] Wang, J., P. Yang and D. Sheng, "Design of a 3-V 300-MHz low-power 8-b by 8-b pipelined multiplier using pulse-triggered TSPC flip-flops", IEEE J. Solid-State Circuits, vol. 35, pp. 583-592, Apr. 2005.