High Speed Booth Encoded Multiplier By Minimising The Computation Time

S.Arul Mozhi
Assistant Professor, Department of ECE

Aswin Kumar.V, Sundaram.C, Arulmani.S, Balamurugesan.N
Final year, Department of ECE
SNS College of Engineering, Coimbatore

Abstract- Two's complement multipliers are used in most of the applications. The computation time is important in two's complement multiplier. The computation time gets decreased by reducing the number of gates. The reduction can be achieved by Modified Booth Encoded multiplier Technique. Two's complement multipliers are used in wide range of applications like multimedia, 3D graphics, signal processing etc. In this project, one row of the partial product array can be reduced without increasing the delay. This MBE technique allows faster computation of the partial product array which is used in most of the multiplier designs.

I. INTRODUCTION

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets - high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.

In parallel multipliers number of partial products to be added is the main parameter that determines the performance of the multiplier. To reduce the number of partial products to be added, Modified Booth algorithm is one of the most popular algorithms. To achieve speed improvements Wallace Tree algorithm can be used to reduce the number of sequential adding stages

Booth Multipliers is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly. For the standard add-shift operation, each multiplier bit generates one multiple of the multiplicand to be added to the partial product. If the multiplier is very large, then a large number of multiplicands have to be added. In this case the delay of multiplier is determined mainly by the number of additions to be performed. If there is a way to reduce the number of the additions, the performance will get better. Booth algorithm is a method that will reduce the number of multiplicand multipliers.VLSI refers to very large scale integration. It is basically a technology applied for minimization of circuits to achieve complexity, speed, performance and cost. VLSI implements the design of every complex integrated circuit in a single chip and improves all the above features. VLSI technology is applied in many innovative areas such as multimedia, 3D graphics and signal processing based system design.

VHDL stands for VHSIC (very high speed integrated circuits) Hardware Description language. It has become now one of industry's standard languages used to describe digital circuits. The other widely used hardware description language is verilog. Both are powerful languages that allow you to describe and simulate complex digital system. A third HDL language is ABEL (Advanced Boolean Equation Language) which was specifically designed for programmable Logic Devices (PLD). ABEL is less powerful then the other two languages and is less popular in industry. VHDL is used mainly for the development of application Specific Integrated Circuit (ASICs). A hardware description of the digital system is called as entity

II. LITERATURE SURVEY

Martin S. Schmookler (1991) describes about the AltiVec technology is an extension to the Power PC architecture which provides new computational and storage operations for handling vectors of various data lengths and data types. The first implementation using this technology is a low cost, low power processor based on the acclaimed Power PC 750 microprocessor. This describes the micro architecture and design of the vector arithmetic unit of this implementation [10].

D.S.Dawoud (1965) describes the performance of multiplication is crucial for multimedia applications such as 3D graphics and signal processing systems, which depend on the execution of large numbers of multiplications. Many algorithms are proposed to implement high speed parallel multipliers. These algorithms mainly focused on rapidly reducing the partial products rows down to final sums and carries used for the final accumulation. Fewer partial products rows means lowering the overall operation time. The second technique uses the conventional way of getting the 2's complement but a simple hardware is proposed to implement the "add 1" operation without any carry propagation. In addition to the speed improvement, our algorithms result in a true diamond-shape for the partial product tree, which is more efficient in terms of implementation. The simulation of our proposed techniques showed large improvement in speed and in power consumption when compared to conventional multiplication algorithms [15]. Kyeongsoon Cho (2010) describes the pipeline architecture of high-speed modified booth multipliers. The multiplier circuits are based on the modified booth algorithm and the pipeline technique which are the most widely used to accelerate the multiplication speed. In order to implement the optimally pipelined multipliers, many kinds of experiments have been conducted. The speed of the multipliers is greatly improved by properly deciding the number of pipeline stages and the positions for the pipeline registers to be inserted. We described the proposed modified Booth multiplier circuits in Verilog HDL and synthesized the gate-level circuits using 0.13um standard cell library. The resultant multiplier circuits show better performance than others. Since the proposed multipliers operate at GHz ranges, they can be used in the systems requiring very high performance [12] .

Reto Zimmermann (2003) describes the latest approach to data path synthesis from RTL, data paths are extracted into largest possible sum-of-product (SOP) blocks, thus making extensive use of carry-save intermediate results and reducing the number of expensive carry propagations to a minimum. The sum-of-product blocks are then implemented by constraint- and technology-driven generation of partial products, carry-save adder tree and carry-propagate adder. A smart generation feature selects the best among alternative implementation variants Special data path library cells are used where available and beneficial. All these measures translate into better performing circuits for simple and complex data paths in cell-based design [6] .

Paul F. Stelling (1998) present new design and analysis techniques for the synthesis of parallel multiplier circuits that have smaller predicted delay than the best current multipliers. In Oklobdzija, etal suggested a new approach, the Three-Dimensional Method (TDM), for Partial Product Reduction Tree (PPRT) design that produces multipliers that outperform the current best designs. The goal of TDM is to produce a minimum delay PPRT using full adders. This is done by carefully modeling the relationship of the output delays to the input delays in an adder and, then, interconnecting the adders in a globally optimal way. Oklobdzija et al. suggested a good heuristic for finding the optimal PPRT, but no proofs about the performance of this heuristic were given. We provide a formal characterization of optimal PPRT circuits and prove a number of properties about them. For the problem of summing a set of input bits within the minimum delay, we present an algorithm that produces a minimum delay circuit in time linear in the size of the inputs [14].

Jung-Yup Kang (2002) proposes an innovative algorithm to the two's complement of a binary number. This method works in logarithmic time (O (log)) instead of the worst case linear time ($O(N)$) where a carry has to ripple all the way from LSB to MSB. The proposed method also allows for more regularly structured logic units which can be easily modularized and can be naturally extended to any word size. Our synthesis results show that our method achieves up to 2.8£ of performance improvement and up to 7.27£ of power savings compared to the conventional method [12].

Stuart N.Wooters (2010) presents a fast energy-efficient level converter capable of conv erting an input signal from sub threshold voltages up to the nominal supply voltage. Measured results from a 130-nm test chip show robust conversion from 188 mV to 1.2 V with no intermediate supplies required. A combination of circuit methods makes the converter robust to the large variations in the current characteristics of sub threshold circuits. To support dynamic voltage scaling, the level converter can up convert an input at any voltage within this range to 1.2 V [4].

Pramod M transistor level CMOS implementation of XOR3 and DFF is done in 130 nm CMOS process. The layout for the cells are designed and extracted for parasitic. The extracted net list is simulated for power, delay and input capacitance and the simulation results are used in standard library cell for synthesis of ALU core from the SPARC micro-processor. The ALU can operate at frequency of 65 MHz considering area of XOR3 and DFF as 83.15 and 82.95 respectively [8]

III. HIGH SPEED BOOTH ENCODED MULTIPLIER

Two's complement multipliers of moderate bit-width (less than 32 bits) are also being used massively in FPGAs. This is used in the design of high performance short or moderate bit-width two's complement multipliers. The basic algorithm for multiplication is based on the well-known paper and pencil approach and passes through three main phases partial product (PP) generation, PP reduction, and final addition. During PP generation, a set of rows is generated where each one is the result of the product of one bit of the multiplier by the multiplicand. The PP reduction is the process of adding all PP rows by using a compression tree. The aim is to produce a PP array with a maximum height of [n/2] rows that is then reduced by the compressor tree stage.

Modified Booth Encoding (MBE) is a technique that has been introduced to reduce the number of PP rows, still keeping the generation process of each row both simple and fast enough. One of the most commonly used schemes is radix-4 MBE, for a number of reasons, the most important being that it allows for the reduction of the size of the partial product array by almost half, and it is very simple to generate the multiples of the multiplicand. More specifically, the classic two's complement $n \times n$ bit multiplier using the radix-4 MBE scheme, generates a PP array with a maximum height of $[n/2] + 1$ rows, each row before the last one being one of the following possible values: all zeros, $\pm X, \pm 2X$.

3.1 Modified Booth Multiplier Technique

This technique allows faster computation of the partial product array which is used in most of the multiplier designs. In general, a radix-$B = 2^b$ MBE leads to a reduction of the number of rows to about $[n/b]$ while, on the other hand, it introduces the need to generate all the multiples of the multiplicand X, at least from $-B/2 \times X$ to $B/2 \times X$. As mentioned above, radix-4 MBE is particularly of interest since, for radix-4, it is easy to create the multiples of the multiplicand 0, $\pm X$, $\pm 2X$. In particular, $\pm 2X$ can be simply obtained by single left shifting of the corresponding terms $\pm X$. It is clear that the MBE can be extended to higher radices but the advantage of getting a higher reduction.

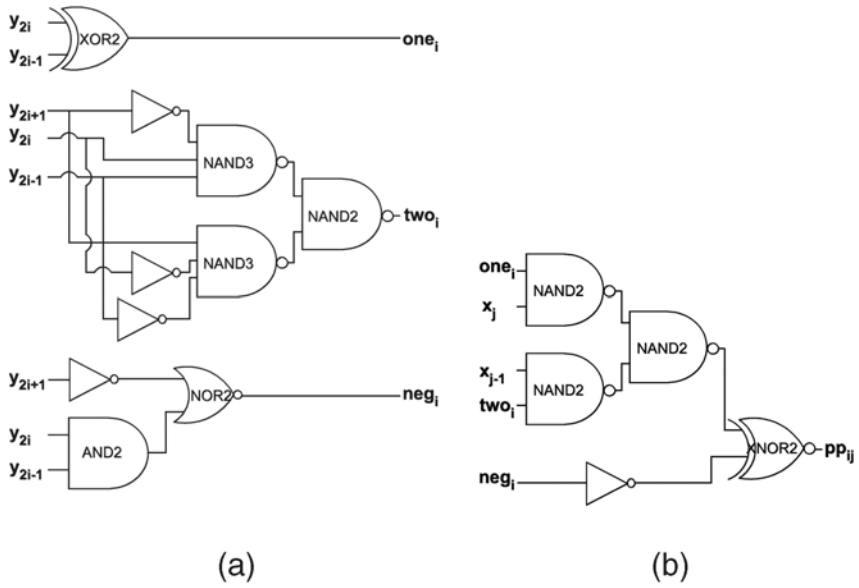

Fig.1. Gate-level diagram for partial product generation using MBE
(a) MBE signals generation. (b) Partial product

MBE Focus on radix-4 MBE, although the proposed method can be easily extended to any radix-B MBE. From an operational point of view, it is well known that the radix-4 MBE scheme consists of scanning the multiplier operand with a three-bit window and a stride of two bits (radix-4). A possible implementation of the radix-4 MBE and of the corresponding partial product generation is shown in Fig b, which comes from a small adaptation. For each partial product row Fig a, produces the one, two, and negation signals.

These signals are then exploited by the logic in Fig b, along with the appropriate bits of the multiplicand, in order to generate the whole partial product array. As introduced previously, the use of radix-4 MBE allows for the (theoretical) reduction of the PP rows to [n/2], with the possibility for each row to host a multiple of $y_i \times X$, with y_i $\{0, \pm 1, \pm 2\}$. While it is straight forward to generate the positive terms 0, X, and 2X at least through a left shift of X,

some attention is required to generate the terms -X and -2X which, as observed in Table 3.1, can arise from three configurations of the y_{2i+1}, y_{2i}, and y_{2i-1} bits Thus, a number of strategies for preventing sign extension have been developed. Thus, although for a n × n multiplier, only [n/2] partial products are generated, the maximum height of the partial product array is [n/2] +1.

When 4-to-2 compressors are used, which is a widely used option because of the high regularity of the resultant circuit layout for n power of two, the reduction of the extra row may require an additional delay of two XOR2 gates. However, the reduction still requires additional hardware, roughly a row of n half adders. This issue is of special interest when n is a power of two, which is by far a very common case, and the multiplier's critical path has to fit within the clock period of a high performance processor. For n = 16, the maximum column height of the partial product array is nine, with an equivalent delay for the reduction of six XOR2 gates. For a maximum height of the partial product array of 8, the delay of the reduction tree would be reduced by one XOR2 gate Alternatively, with a maximum height of eight, it would be possible to use 4 to 2 adders, with a delay of the reduction tree of six XOR2 gates, but with a very regular layout .

When 4-to-2 compressors are used, which is a widely used option because of the high regularity of the resultant circuit layout for n power of two, the reduction of the extra row may require an additional delay of two XOR2 gates Alternatively, with a maximum height of eight, it would be possible to use 4 to 2 adders, with a delay of the reduction tree of six XOR2 gates, but with a very regular layout.

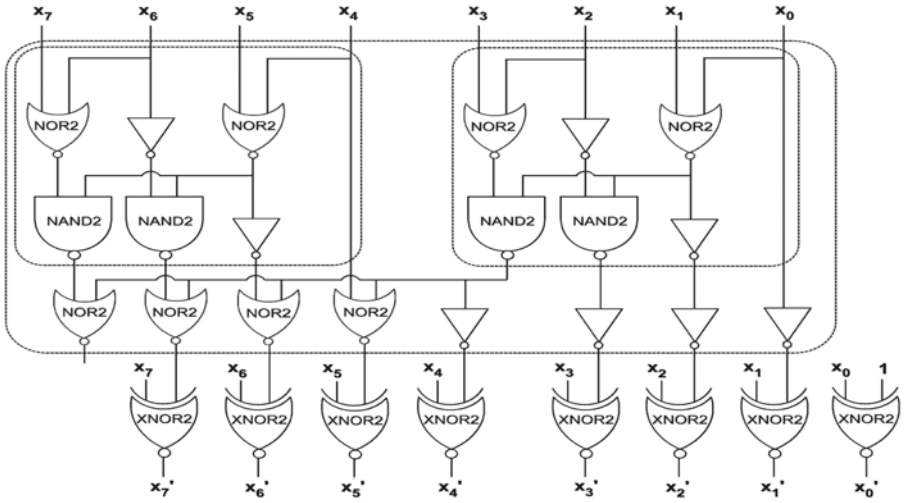

Fig.2.Two's Complement Computation (n = 8)

In case of n square multipliers is quite common, as the case of n that is a power of two. Thus, we start by focusing our attention on square multipliers, and then present the extension to the general case of m x n rectangular multipliers.The project approach is general and, for the sake of clarity, will be explained through the practical case of 8 x 8 multiplications. As briefly outlined in the previous sections, the main goal of our approach is to produce a partial product array with a maximum height of [n/ 2] rows, without introducing any additional delay. Let us consider, as the starting point, the form of the simplified array as reported for all the partial product rows except the first one. As depicted in the first temporarily considered as being split into two sub rows, the first one containing the partial product bits (from right to left) from pp00 to pp80 and the second one with two bits set at "one" in positions 9 and 8.

Then, the bit neg3 related to the fourth partial product row, is moved to become a part of the second sub row. we have observed that it requires a carry propagation only across the least-significant three positions, a fact that can also be seen by the implementation. It is worth observing that, in order not to have delay penalizations, it is necessary that the generation of the other rows is done in parallel with the generation of the first row cascaded by the computation of the bits qq_{90} qq_{90} qq_{80} qq_{70} qq_{60}.This MBE observe that the booth recoding for the first row is computed more easily than for the other rows, because the y_{-1} bit used by the MBE is always equal to zero. In order to have a preliminary

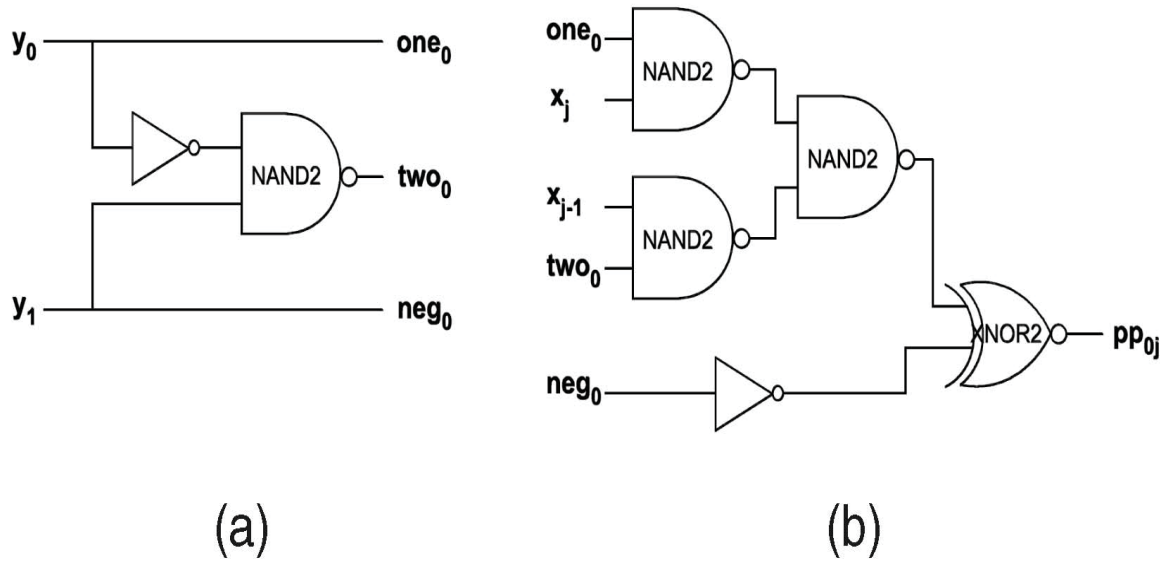

Fig.3.Gate-level diagram for partial product generation
(a) MBE signals generation. (b) Partial product generation

IV. CONCLUSION

The MBE method based on the addition of the last negation signal to the first row is first evaluated. The designed architecture is then compared with the other methods and implementation based on the computation of the two's complement of the last row using the designs for the 3-5 decoders, 4-1 multiplexers, and two's complement tree. In the analysis, the MBE reduces the number of gates used, then power and time consumed. The reduced number of gates of this work offers the great advantage in the reduction of area and also the total power. The simulated results shows that the reduction in area and power consumed by the multiplier. Therefore, MBE technique is low area, low power, simple and efficient for VLSI hardware implementation.

REFERENCES

[1] Baugh,C.R. and Wooley,B.A.(1973) 'A two's complement parallel array multiplication algorithm' IEEE Transactions on Computers,
[2] vol.C-22, pp.1045–1047.
[3] Booth,A.D.(1951) 'A signed binary multiplication technique' Q. J. Mech. Appl. Math., vol. 4,pp. 236–240.
[4] Bickerstaff.K., Schulte.M.J, and Swartzlander,E.EJr.(1995) 'Parallel Reduced Area Multipliers' Journal of VLSI Signal Processing, vol
[5] 9, pp.181–192.
[6] Chang,J.N, Satayanarayana,J.H. and Parhi,K.K.,(1998) 'Systematic design of high-speed and low-power digit-serial multipliers' IEEE
[7] M Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no. 12, pp. 1585–1596.
[8] Cho,K.J., Lee,K.C., Chung,J.G. and Parhi,K.K.(2004) 'Design of low error fixed-width modified Booth multiplier' IEEE Trans. Very
[9] Large Scale Integration. (VLSI) Syst., vol. 12, no. 5, pp. 522–531.
[10] Dadda,L. (1965) 'Some Schemes for Parallel Multipliers' Alta Frequenza, vol. 34, pp. 349-356.
[11] Elguibaly, F(2000) 'A fast parallel multiplier-accumulator using the modified Booth algorithm' IEEE Trans. Circuits Syst. II, Reg.
[12] Papers, vol. 47,no. 9, pp. 902–908.
[13] Ercegovac,M.D and Lang,T.(2003) Digital Arithmetic, Morgan Kaufmann Publishers.
[14] Gajski,D. (1997) Principles of Digital Design. Prentice-Hall.
[15] Hwang,K.(1979) Computer Arithmetic Principles, Architectures, and Design.
[16] Hashemian,R. and Chen,C.P.(1991) 'A New Parallel Technique for Design of Decrement/Increment and Two's Complement Circuits'
[17] Proc. 34th Midwest Symp. Circuits and Systems, vol. 2,pp. 887-890
[18] Huang,Z. and Ercegovac,M.D. (2005) 'High-Performance Low-Power Left-to-Right Array Multiplier Design' IEEE Trans.Computers,
[19] vol.54, no.3, pp.272-283.
[20] Hsu,S.K., Mathew,S.K., Krishnamurthy,R.K, and Borkar,S.Y,(2006) 'A 110GOPS/W 16-Bit Multiplier and Reconfigurable PLA Loop
[21] in 90-nm CMOS' IEEE J.Solid State Circuits, vol. 41, no. 1, pp. 256-264.
[22] Jen,C.W and Yeh.W.C.(2000) 'High-Speed Booth Encoded Parallel Multiplier Design' IEEE Trans. Computers, vol. 49, no.7, pp. 692-
[23] 701.
[24] Kang,J.Y and Gaudiot.J.L,(2004) 'A Fast and Well-Structured Multiplier' Proc. Euromicro Symp. Digital System Design, pp. 508-515.
[25] Kang.J.Y and Gaudiot.J.L,(2005) 'A Logarithmic Time Method for Two's Complementation' Proc. Int'l Conf. Computational Science,
[26] pp. 212-219.
[27] Kang J.Y and Gaudiot J.L(2006) 'A Simple High-Speed Multiplier Design' IEEE Trans. Computers, vol. 55, no. 10, pp. 1253-1258.
[28] Lamberti,F., Andrikos,N., and Montuschi.P,(2009) 'Speeding-Up Booth Encoded Multipliers by Reducing the Size of Partial Product

[29] Array' internal report, http://arith.polito.it/ir_mbe.pdf, pp. 1-14.
[30] MacSorley,O.L.(1961) 'High Speed Arithmetic in Binary Computers' Proc. IRE, vol. 49, pp. 67-91.
[31] Oklobdzija,V.G., Villeger,D. and Liu,S.S.(1996) 'A Method for Speed Optimized Partial Product Reduction and Generation of Fast
[32] Parallel Multipliers Using an Algorithmic Approach' IEEE Trans.Computers, vol.45, no.3, pp.294-306.