

Optimizing the Modified Booth Recoder for Fused Add-Multiply Operator

Manju Mallayagari¹, G. Sahithi Reddy²

^{1,2}(Assistant Professor, ECE Dept., Mallareddy Engineering College for Women, JNTU Hyderabad, India)

Abstract

This proposed method the design and simulation of Radix-8 Booth Encoding that can employ in DSP applications. As the Radix-8 Booth Encoder circuit produces $n/3$ the partial products in parallel manner. The proposed scheme effectively implements a newly recoding technique for modified booth recoding that ensures to implement the direct recoding of the multiplier in its Sum Modified Booth (SMB) form. Another vital advantage of S-MB algorithm that is well structured, simple and can be easily modified in order to apply either in signed or unsigned numbers, which consists of odd or even number of bits. Also Carry Save Adder (CSA) tree and the final Carry Look ahead (CLA) adder used to speed up the multiplier operation. Because of signed and unsigned multiplication operation is well performed by the same multiplier unit the required hardware and thus chip area reduces and this in turn reduces power dissipation and cost of a system. Finally Fused Add Multiply operator is optimized to increase the performance of complex arithmetic operation with three different recoding schemes S-MB1, S-MB2, S-MB3. The simulation is done through Verilog on xilinx13.3 platform which provide accurate in calculating the various parameters.

Keywords - Array multiplier, Fused Add Multiply, Braun array multiplier, CLA, CSA, Radix-8 Booth Encoding multiplier, Signed-unsigned.

I. INTRODUCTION

In general in many applications like specially DSP's kind of things we require arithmetic optimization were shown[1-2] in which designing of arithmetic components that combine operations which share data leads to significant performance improvements. So based on deep observations that addition can often use of multiplication for the purpose of Multiply-Accumulator (MAC) and Multiply-Add (MAD) units that are introduced in [3] leading to more efficient implementations of DSP kind of algorithms as compared to the conventional ones, which use only primary resources [4]. The straightforward design of the AM unit, by first allocating an adder and then driving its output to the input of a multiplier, increases significantly both area and critical path delay of the circuit. Digital multiplication is essentially a series of additions, which can be performed with a reduced level of carry-propagation using a redundant representation. Indeed, the redundant carry-save representation has been used extensively for this purpose in

conventional multipliers, where the final sum, in its carry-save form, is converted to non-redundant binary using a carry-propagate adder. In particular, it is shown that a multiplier can be designed with one redundant input, which is suitable for applications such as high sample-rate IIR filters and the Givens rotation, with less area than a conventional multiplier.

As far as Portable multimedia and digital signal processing (DSP) systems, which may require enhanced processing ability, very low power consumption, and have short design cycle, had become increasingly popular over the past few years. Generally Multimedia as well as DSP applications are very highly multiplication intensive so that the performance and power consumption of these systems are dominated by multipliers. The straightforward design of the AM unit, by first allocating an adder and then driving its output to the input of a multiplier, increases significantly both area and critical path delay of the circuit.

The proposed system optimize the design of AM operators, by introducing fusion techniques which is based on the direct recoding of the sum of two numbers in its Modified Booth (MB) form. The direct recoding of the sum of two numbers in its MB form leads to a more efficient implementation of the fused Add-Multiply (FAM) unit compared to the conventional one. The Sum-Modified Booth (S-MB) recoding techniques are efficiently used to implement the direct recoding of the sum of two numbers in its MB form. As our proposed technique results in reductions in terms of hardware complexity, critical delay and also power consumption of FAM unit.

Fast carry propagate adders are important to high performance multiplier design in two ways.

- First, an efficient and fast adder is needed to make any "hard" multiples that are needed in partial product generation.
- Second, after the partial products have been summed in a redundant form, a carry propagate adder is needed to produce the final non redundant product.

II. LITERATURE SURVEY

A. Array Multiplier

As we say Array multiplier is very efficient layout of a combinational multiplier where multiplication of two binary number can be yielded with only one micro-operation by employing a

combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array.

By having flexible structure, this multiplier is depends on the standard add and shift operations very easily. Moreover each partial product is generated by considering multiplicand and one bit of multiplier each time. Similarly impending addition is carried out by efficient high-speed carry-save algorithm and also the final product is yielded employing any of fast adders – consequently number of partial products depends upon the number of multiplier bits used.

Conclusion: Array Multiplier provides more power consumption as well as optimum number of requirement of components, at the other hand provides much delay. As the area is increased requirement of gates is also more, so due to this constraint array multiplier is less economical and hardware complexity is high.

B. Wallace Tree Multiplier

Wallace tree performs faster multiplication of two numbers that employs a three step method in order to complete this task where the bit products are can be formed and its matrix is reduced to a two row matrix in which its sum of the row equals the sum of bit products then the two resulting rows are summed up with efficient fast adder to obtain a final product. A fast adder is at the end produces the final result.

Conclusion: In Wallace tree circuit layout is complex even though speed of operation of it is high due to its irregular structure. Generally it is not preferable in low power applications because of excess wiring usage that result in extra circuitry and increase in power consumption in the multiplier.

C. Braun Multiplier

Braun multiplier is more efficient due to its regular structure. Due to its simple nature of parallel multiplier that is commonly termed as carry save array multiplier. So it is multiplier that restricted to perform multiplication of two unsigned numbers. Generally speaking about its structure that consists of array of AND gates and adders arranged in iterative structure that does not require any logic registers. So therefore termed as the non-additive multiplier since it does not add an additional operand to result of multiplication.

Conclusion: the major shortcomings of the Braun's multiplier is that the number of components required increases quadratically with that of number of bits which will make the multiplier to be inefficient. It cannot stop the switching activity even if the bit Coefficient is zero that ultimately results in unnecessary Power dissipation.

D. Booth Multiplier

The Booth recoding multiplier is one such multiplier; it scans the three bits at a time to reduce the number of partial products [7]. As these three bits in them the two bits from the current pair; and a third bit is taken from the high order bit of an adjacent lower order pair. Now after testing each triplet of bits, they are converted by Booth logic unit into a set of five control signals where it can be used by the adder cells in the array that control the operations performed by the adder cells. Also Booth recording reduces the numbers of adders and on the other hand delay required generating the partial sums by observing three bits at a time. In order to speed up the multiplication Booth encoding performs many steps of the multiplication simultaneously. Therefore we conclude Booth's algorithm takes merit of the consideration that an adder subtracted is nearly as fast and small as a simple adder.

Conclusion: The major drawbacks of booth multiplier are number of add subtract and shift operation becomes variable and becomes inconvenient for designing parallel multipliers. Also algorithm becomes inefficient when there are some isolated 1's that results in more power consumption due to large number of adders. Thus summing the partially redundant partial products requires as more hardware as representing them in the fully redundant form.

III. PROPOSED SYSTEM

Our proposed system optimize the design of AM operators, by introducing the fusion techniques that were based on the direct recoding of the sum of two numbers in its Modified Booth (MB) form. The direct recoding of the sum of two numbers in its MB form leads to a more efficient implementation of the fused Add-Multiply (FAM) unit compared to the conventional one. The Sum-Modified Booth (S-MB) recoding techniques are efficiently used to implement the direct recoding of the sum of two numbers in its MB form. Also our proposed technique obtains considerable reductions in terms of critical delay, hardware complexity and power consumption of the FAM unit.

FAM design with sum-modified booth (S-MB) recoding technique reduces the number of partial products and increasing speed of calculation. The FAM technique which decreases the critical path delay and reduces area and power consumption. The proposed S-MB algorithm is so structured in the manner that shows, simple and thus easily modified in order to be applied either in signed (in 2's complement representation) or unsigned numbers, which comprise of odd or even number of bits.

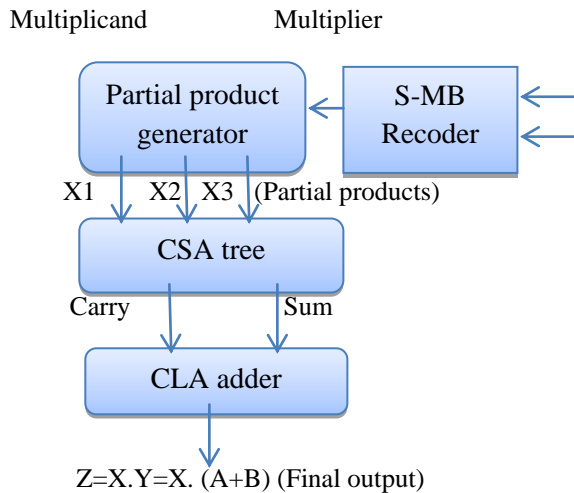


Fig.1 Block Diagram of Proposed System

An optimized design of the AM operator is based on the fusion of the adder and the MB encoding unit into a single data path block as depicted in (Fig. 1) by direct recoding of the sum $Y=A+B$ to its MB representation. The fused Add Multiply (FAM) component contains only one adder at the end (final adder of the parallel multiplier). So due this significant saving in area and critical path delay for the recoding process is greatly reduced. Thus we can say that FAM Design is a new technique for direct recoding of two numbers in the MB representation of their sum. In Fig 1 there are 3-partial products namely X1, X2, X3. These partial products are added by the carry save adder (CSA) and the final stage is carry look ahead (CLA) adder. CSA adder takes three inputs and produce sum and carry parallel. There is one CSA. Three partial products are added by the CSA tree and finally when there are only two outputs. Left out then finally CLA adder is used to produce final result. CSA Tree: The conventional carry select adder consists of n-bit adder for the lower half of the bits i.e. least significant bits (LSB's) and for the upper half i.e. most significant bits (MSB's) two n-bit adders. In MSB adder's one adder assumes carry input as one for performing addition and another assumes carry input as zero.

As carry out calculated from the previous stage i.e. least significant bit stage is used to select the actual calculated values of output carry and sum produced where its selection is done by using a multiplexer. Thus this method of dividing adder in two stages increases the area utilization but at same time addition operation gets fastens.

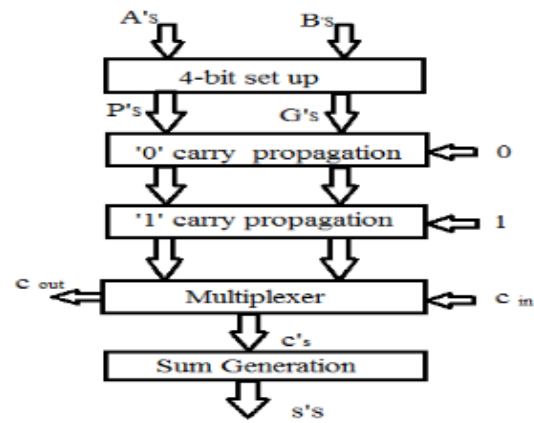


Fig.2 Block diagram of Carry Select Adders

Fig.3 shows block diagram of Carry Select Adders (CSLA) is one of the fastest adders used in many applications of data-processing processors to efficiently perform fast arithmetic functions.

CLA Adder: A carry-look-ahead adder improves speed by reducing the amount of time required to determine carry bits. The carry-look-ahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits. One of the widely used method employs the principle of carry-look-ahead solves this problem by calculating the carry signals in advance, based on the input signals.

It is based on the fact that a carry signal will be generated in two cases:

- When both bits A_i and B_i are 1, or
- When one of the two bits is 1 and the carry-in (carry of the previous stage) is 1.

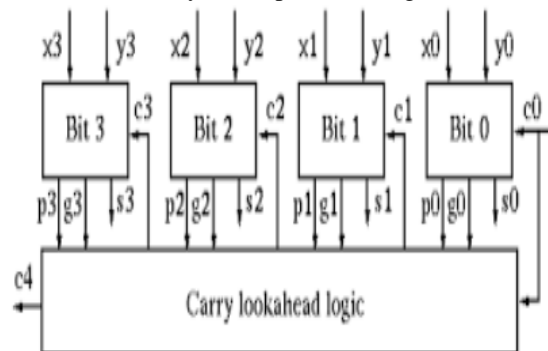


Fig.3 Block diagram of CLA

The sum to modified booth recoder is embedded with adder and encoder block .It is structured with half adders and full adders where the adder and encoding is done in single structure. Here fused block reduces the area of the FAM design. This S-MB Recoder block is implemented by S-MB Recoder technique.

S-MB Recoding Techniques: Both the conventional and signed HAs and FAs is used to design the three new alternative schemes of the S-MB recoding technique by radix-8 Booth encoding.

- 1) S-MB1 Recoding Scheme
- 2) S-MB2 Recoding Scheme
- 3) S-MB3 Recoding Scheme

Each of the three schemes can be easily applied in either signed (2's complement representation) or unsigned numbers which consist of odd or even number of bits. Consider that both inputs A and B are in 2's complement form and consist of 2k bits in case of even or 2k+1 bits in case of odd bit width. Targeting to transform the sum of A and B(Y=A+B) in its MB representation .The three S-MB recoding schemes are:

Radix-8 Booth Encoding: As we know that Booth's algorithm repeatedly add one of two predetermined values to a product P, and then performing rightward arithmetic shift on P.Radix-8 Booth encoding is most often used to avoid variable size partial product arrays before designing Radix-8 BE, the multiplier has to be converted into a Radix-8 number by dividing them into 4 digits respectively according to [5] Booth Encoder Table given afterwards. Prior to convert the multiplier, a zero is appended into the Least Significant Bit (LSB) of the multiplier.

Table (1): Radix-8 recoding

Quarter value	Signed extension
0000	0
0001	+1
0010	+1
0011	+2
0100	+2
0101	+3
0110	+3
0111	+4
1000	-4
1001	-3
1010	-3
1011	-2
1100	-2
1101	-1

1110	-1
1111	0

Partial Product Generator : Here the product obtained by multiplying the multiplicand with that of multiplier when multiplier possess more than one digit then the partial products are used as intermediate steps in calculating larger products .Partial product generator is designed to produce the product by multiplying the multiplicand M by 0, 1, -1,2,- 2,-3,-4, 3, 4.

- For product generator, multiply by zero means the multiplicand is multiplied by “0”.Multiply by “1” means the product still remains the same as the multiplicand value.
- Multiply by “-1” means that the product is the two’s complement form of the number. Multiply by “-2” is to shift left one bit the two’s complement of the multiplicand value and multiply by “2” means just shift left the multiplicand by one place.
- Multiply by “-4” is to shift left two bit the 2’s complement of the multiplicand value and gets multiply by “2” indicates just shift left the multiplicand by two places.

IV. SIMULATION RESULTS

The proposed Fused add unit is implemented in [6] the Verilog, simulated with Xilinx and modelsim. Testing is carried out using various inputs. It is a used to reduce the computation in the circuits. FAM unit with CLA adder for odd bit.

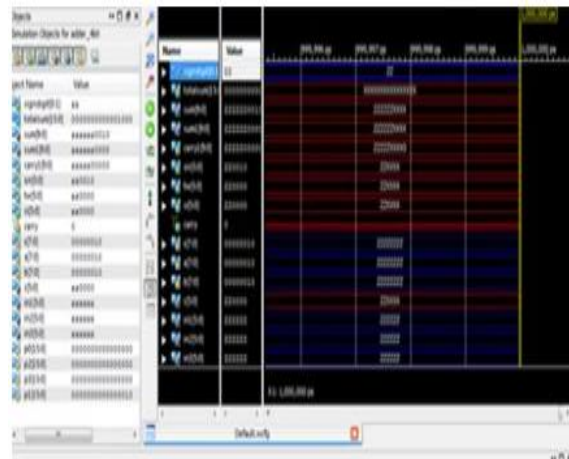


Fig 4. Output of S-MB1 Even

A two 8-bit input A and B is given to the FAM unit. Then produced output 8-bit from FAM unit is multiplied with the 8-Bit multiplicand X and partial products are generated. Then the generated 16-bit Z output is taken from CLA adder. Verilog code is written to generate the required hardware and to produce the partial product.

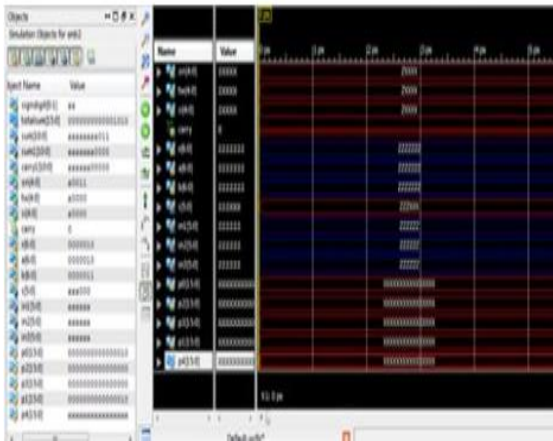


Fig 5. Output of S-MB1 odd

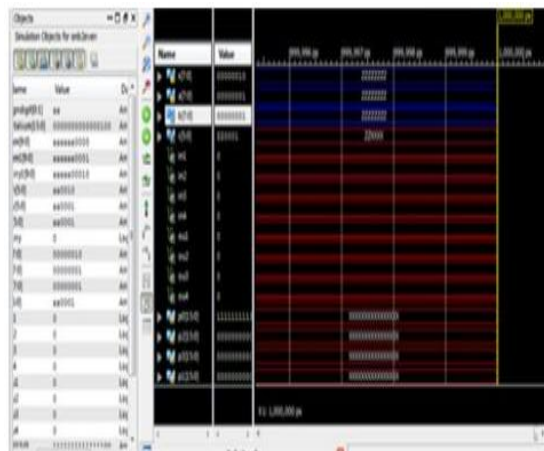


Fig 6. Output of S-MB2 even



Fig 7. Output of S-MB2 odd

V. CONCLUSION

In our paper, we proposed novel high speed architecture for multiplication of two 8 bit numbers, combining the advantages of Modified Booth algorithm and CSA tree to get faster arithmetic operations in case of multiplier. Moreover the simulation results of different S-MB methods are compared and the proposed multiplier is optimized according to speed, generating partial products and sums. Our scheme reduces reduction in critical delay, hardware complexity and power consumption of

FAM unit. Our multiplier can be used in high speed applications such as adaptive filter, phase locked loop and neural networks. As a future work, the multiplier’s performance could be tested with Filters, ALU and several other multipliers.

REFERENCES

- [1] A. Amaricai, M. Vladutiu, and O. Boncalo, “Design issues and implementations for floating-point divide-add fused,” IEEE Trans. Circuits Syst. II–Exp. Briefs, vol. 57, no. 4, pp. 295–299, Apr. 2010.
- [2] E. E. Swartzlander and H. H. M. Saleh, “FFT implementation with fused floating-point operations,” IEEE Trans. Comput., vol. 61, no. 2, pp. 284–288, Feb. 2012.
- [3] J. J. F. Cavanagh, Digital Computer Arithmetic. New York: McGraw Hill, 1984.
- [4] S. Nikolaidis, E. Karaolis, and E. D. Kyriakis-Bitzaros, “Estimation of signal transition activity in FIR filters implemented by a MAC architecture,” IEEE Trans. Comput.-Aided Des.Integr. Circuits Syst., vol.19, no. 1, pp. 164–169, Jan. 2000.
- [5] An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator Kostas Tsoumanis, Student Member, IEEE, Sotiris Xydis, Constantinos Efstathiou, Nikos Moschopoulos, and Kiamal Pekmestzia IEEE transactions on circuits and systems: regular papers, vol. 61, no. 4, April 2014.
- [6] E. E. Swartzlander and H. H. M. Saleh, “FFT implementation with fused floating-point operations,” IEEE Trans. Comput., vol. 61, no. 2, pp. 284–288, Feb. 2012.
- [7] Y.-H. Seo and D.-W. Kim, “A new VLSI architecture of parallel multiplier-accumulator based on Radix-2 modified Booth algorithm,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp.201–208, Feb. 2010.