Design of The Optimized Fused-Add Multiply (Fam) Operator By Using Modified Booth

DOKKARA ANUSHA
Mtech.,
Avanthi inst. of Engg. & Tech.,
Makavarapalem, Narisipatnam, Visakhapatnam

P.ASHOK KUMAR
Assistant professor
Avanthi inst. of Engg. & Tech.,
Makavarapalem, Narisipatnam, Visakhapatnam

Abstract: **Complex arithmetic operations are widely used in Digital Signal Processing (DSP) applications. In this work, we focus on optimizing the design of the fused Add-Multiply (FAM) Operator for increasing performance. We investigate techniques to implement the direct recoding of the sum of two numbers in its Modified Booth (MB) form. We introduce a structured and efficient recoding technique and explore three different schemes by incorporating them in FAM designs. Comparing them with the FAM designs which use existing recoding schemes, the propose technique yields considerable reductions in terms of critical delay, hardware complexity of the FAM unit. The FAM Architecture is implemented by Verilog Hardware Description Language and it is synthesized by Xilinx ISE tool.**

In proposed, we focus on AM units which implement the operation. The conventional design of the AM operator requires that its inputs and are first driven to an adder and then the input and the sum are driven to a multiplier in order to get. The drawback of using an adder is that it inserts a significant delay in the critical path of the AM. As there are carry signals to be propagated inside the adder, the critical path depends on the bit-width of the inputs. In order to decrease this delay, a SPST adder can be used which, however, the increases the area occupation and the power dissipation. 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 by direct recoding of the sum to its MB representation. The fused Add-Multiply (FAM) component contains only one adder at the end (final adder of the parallel multiplier). As a result, significant area savings are observed and the critical path delay of the recoding process is reduced and decoupled from the bit-width of its inputs. In this work, we present a new technique for direct recoding of two numbers in the MB representation of their sum.

Keywords: **Fused Add-Multiply (FAM); AM operator; Verilog HDL; Xilinx ISE**

I. INTRODUCTION

Fast multipliers are essential parts of digital signal processing systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors today, especially since the media processing took off. In the past multiplication was generally implemented via a sequence of addition, Subtraction, and shift operations. Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product. In most computers, the operand usually contains the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional representation. It is possible to decompose multipliers into two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them.

The basic multiplication principle is twofold, i.e. evaluation of partial products and accumulation of the shifted partial products. It is performed by the successive Addition's of the columns of the shifted partial product matrix. The 'multiplier' is successfully shifted and gates the appropriate bit of the 'multiplicand'. The delayed, gated instance of the multiplicand must all be in the same column of the shifted partial product matrix. They are then added to form the product bit for the particular form. Multiplication is therefore a multi operand operation. To extend the multiplication to both signed and unsigned numbers, a convenient number system would be the representation of numbers in two's complement format.

II. HIGH SPEED BOOTH MULTIPIER

Procedure And Working Principle Of Block Diagram

In the majority of digital signal processing (DSP) applications the critical operations usually involve many multiplications and/or accumulations. For real-time signal processing, a high speed and high throughput Multiplier-Accumulator (MAC) is always a key to achieve a high performance digital signal processing system and versatile Multimedia functional units.

Architecture Of Modified 16-Bit Sqrt Csla

Description

This architecture is similar to regular 64-bit SQRT CSLA, the only change is that, we replace RCA with Cin=1 among the two available RCAs in a group with a BEC. This BEC has a feature that it can perform the similar operation as that of the replaced RCA with Cin=1. Fig. 6 shows the Modified block diagram of 64-bit SQRT CSLA. The number of bits required for BEC logic is 1 bit more than the RCA bits. The modified block diagram is also divided into various groups of variable sizes of bits with each group having the ripple carry adders, BEC and corresponding mux. Group 0 contain one RCA only which is having input of lower significant bit and carry in bit and produces result of sum [1:0] and carry out which is acting as mux selection line for the next group, similarly the procedure continues for higher groups but they includes BEC logic instead of RCA with Cin=1.Based on the consideration of delay values, the arrival time of selection input C1 of 6:3 mux is earlier than the sum of RCA and BEC. For remaining groups the selection input arrival is later than the RCA and BEC.

Thus, the sum1 and c1 (output from mux) are depending on mux and results computed by RCA and BEC respectively. The sum2 depends on c1 and mux.

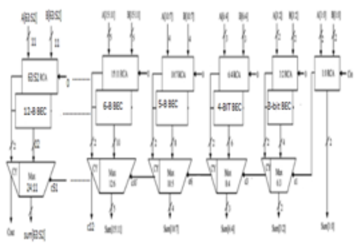


Fig.1CSLA using BEC

III. BASIC ADDER BLOCK

The adder block using a Ripple carry adder, BEC and Mux is explained in this section. In this we calculate and explain the delay & area using the theoretical approach and show how the delay and area effect the total implementation. The AND, OR, and Inverter (AOI) implementation of an XOR gate is shown in Fig. 1. The delay and area evaluation methodology considers all gates to be made up of AND, OR, and Inverter, each having delay equal to 1 unit and area equal to 1 unit. We then add up the number of gates in the longest path of a logic block that contributes to the maximum delay. The area evaluation is done by counting the total number of AOI gates required for each logic block. Based on this approach, the blocks of 2:1 mux, Half Adder (HA), and FA are evaluated and listed in Table I.

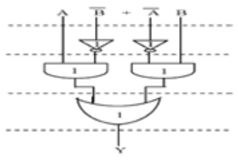


Fig.2: Delay and Area Evaluation of XOR.

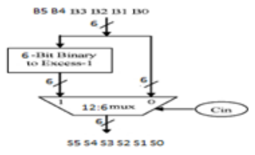


Fig.3: 6-bit BEC with 12:6 mux.

Fig. 3shows the basic 6-bit addition operation which includes 6-bit data, a 6-bit BEC logic and 12:6 mux. The addition operation is performed for Cin=0 and for Cin=1.For Cin=0 the addition is performed using ripple carry adder and for Cin=1 the operation is performed using 6-bit BEC (replacing the RCA for Cin=1).

IV. BINARY TO EXCESS-1 CONVERTER

The basic work is to use Binary to Excess-1 Converter (BEC) in the regular CSLA to achieve lower area and increased speed of operation. This logic is replaced in RCA with Cin=1. This logic can be implemented for different bits which are used in the modified design. The main advantage of this BEC logic comes from the fact that it uses lesser number of logic gates than the n-bit Full Adder (FA) structure. As stated above the main idea of this work is to use BEC instead of the RCA with Cin=1 in order to reduce the area and increase the speed of operation in the regular CSLA to obtain modified CSLA. To replace the n-bit RCA, n+ 1 bit BEC logic is required. The structure and the function table of a 6-bit BEC are shown in Figure.3 and Table .2, respectively

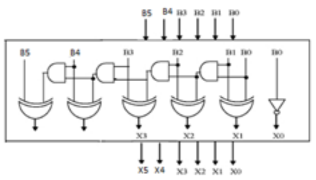


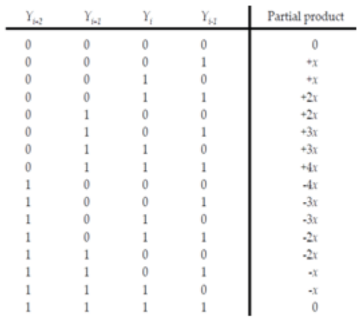
Fig.4:bit BEC Structure

High Speed Boothence

Design

Fast multipliers are essential parts of digital signal processing systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors today, especially since the media processing took off. In the past multiplication was generally implemented via a sequence of addition, subtraction, and shift operations.

Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product. In most computers, the operand usually contains the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional representation. It is possible to decompose multipliers into two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them.



Y_{i+2}	Y_{i+1}	Y_i	Y_{i-1}	Partial product
0	0	0	0	0
0	0	0	1	+x
0	0	1	0	+x
0	0	1	1	+2x
0	1	0	0	+2x
0	1	0	1	+3x
0	1	1	0	+3x
0	1	1	1	+4x
1	0	0	0	-4x
1	0	0	1	-3x
1	0	1	0	-3x
1	0	1	1	-2x
1	1	0	0	-2x
1	1	0	1	-x
1	1	1	0	-x
1	1	1	1	0

Table High-Speed Booth Encoded Parallel Multiplier

Modified Booth (MB) is a prevalent form used in multiplication. It is a redundant signed-digit radix-4 en-coding technique. Its main advantage is that it reduces by half the number of partial products in multiplication comparing to any other radix-2 representation.

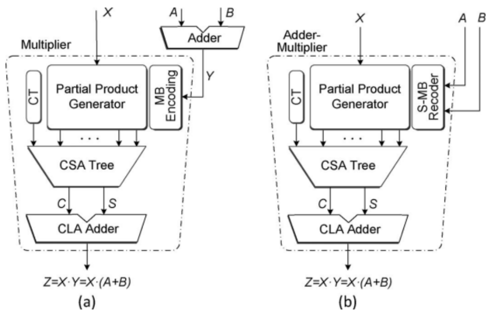


Fig. 5. AM operator based on the (a) conventional design and (b) fused design with direct recoding of the sum of A and B in its MB representation. The mul-tiplier is a basic parallel multiplier based on the MB algorithm. The terms CT, CSA Tree and CSLA Adder are referred to the Correction Term, the Carry-Sellect Adder Tree and the final Carry-Look-Ahead Adder of the multiplier.

V. IMPLEMENTATION OF MULTI OPERAND BOOTH MULTIPLER

Circuit Design Features

One of the most advanced types of MAC for general-purpose digital signal processing has been proposed by Elguibaly. It is an architecture in which accumulation has been combined with the carry save adder (CSA) tree that compresses partial products. In the architecture proposed in, the critical path was reduced by eliminating the adder for accumulation and decreasing the number of input bits in the final adder.

While it has a better performance because of the reduced critical path compared to the previous VMFU architectures, there is a need to improve the output rate due to the use of the final adder results for accumulation. The architecture to merge the adder block to the accumulator register in the VMFU operator was proposed to provide the possibility of using two separate N/2-bit adders instead of one-bit adder to accumulate the MAC results. Recently, Zicari proposed an architecture that took a merging technique to fully utilize the 4–2 compressor .It also took this compressor as the basic building blocks for the multiplication circuit.

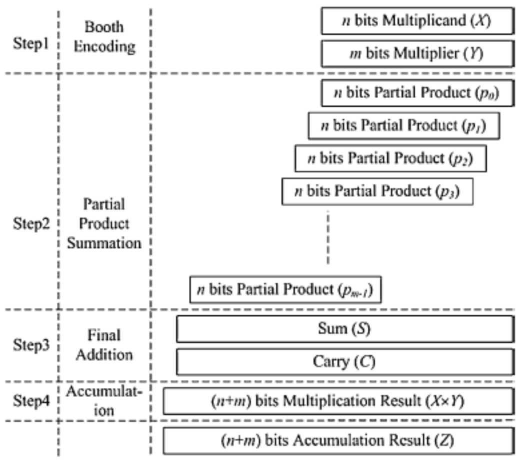

Step1
Booth Encoding
n bits Multiplicand (X)
m bits Multiplier (Y)
Step2
Partial Product Summation
n bits Partial Product (p0)
n bits Partial Product (p1)
n bits Partial Product (p2)
n bits Partial Product (p3)
n bits Partial Product (pm-1)
Step3
Final Addition
Sum (S)
Carry (C)
Step4
Accumulat-ion
(n+m) bits Multiplication Result (X×Y)
(n+m) bits Accumulation Result (Z)

Figure.6 Circuit Design Flow

Spanning carry look ahead adder

Another carry-tree adder known as the spanning tree carry-lookahead (CLA) adder is like the sparse Kogge-Stone adder, this design terminates with a 4- bit RCA. As the FPGA uses a fast carry-chain for the RCA, it is interesting to compare the performance of this adder with the sparse Kogge-Stone and regular Kogge-Stone adders.

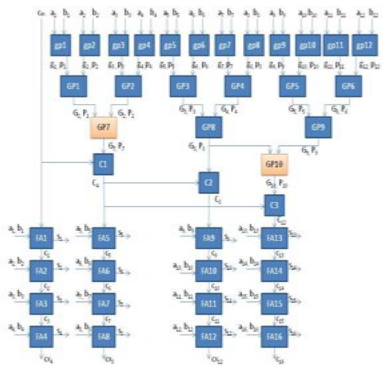


Fig 7: Spanning carry look ahead adder

Modified Booth Encoder

In order to achieve high-speed multiplication, multiplication algorithms using parallel counters, such as the modified Booth algorithm has been proposed, and some multipliers based on the algorithms have been implemented for practical use. This type of multiplier operates much faster than an array multiplier for longer operands because its computation time is proportional to the logarithm of the word length of operands.

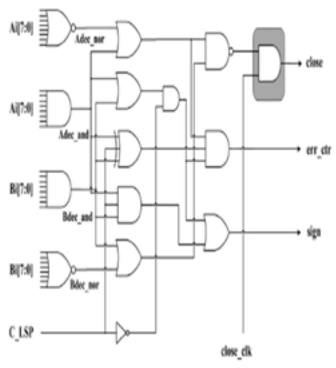


Figure.8 Modified Booth Encoder

Booth multiplication is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied. It is possible to reduce the number of partial products by half, by using the technique of radix-4 Booth recoding. The basic idea is that, instead of shifting and adding for every column of the multiplier term and multiplying by 1 or 0, only takes every second column, and multiply by ±1, ±2, or 0, to obtain the same results.

The advantage of this method is the having of the number of partial products. To Booth recode the multiplier term and consider the bits in blocks of three, such that each block overlaps the previous block by one bit. Grouping starts from the LSB, and the first block only uses two bits of the multiplier. Shows the grouping of bits from the multiplier term for use in modified booth encoding.

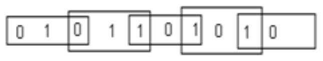


Figure 9 Grouping of bits from the multiplier term

Each block is decoded to generate the correct partial product. The encoding of the multiplier Y, using the modified booth algorithm, generates the following five signed digits, -2, -1, 0, +1, +2. Each encoded digit in the multiplier performs a certain operation on the multiplicand, X, as illustrated in Table 2

Block	Re - coded digit	Operation on X
000	0	0 X
001	+1	+1 X
010	+1	+1 X
011	+2	+2 X
100	-2	-2 X
101	-1	-1 X
110	-1	-1 X
111	0	0 X

Table : Modified Booth Encoder

For the partial product generation and adopt Radix-4 Modified Booth algorithm to reduce the number of partial products for roughly one half. For multiplication of 2’s complement numbers, the two-bit encoding using this algorithm scans a triplet of bits. When the multiplier B is divided into groups of two bits, the algorithm is applied to this group of divided bits.

Computing example of Booth multiplying two numbers ”2AC9” and “006A”. The shadow denotes that the numbers in this part of Booth multiplication are all zero so that this part of the computations can be neglected. Saving those computations can significantly reduce the power consumption caused by the transient signals. According to the analysis of the multiplication .It propose the SPST-equipped modified-Booth encoder, which is controlled by a detection unit. The detection unit has one of the two operands as its input to decide whether the Booth encoder calculates redundant computations. As shown in figure 9. The latches can, respectively, freeze the inputs of MUX-4 to MUX-7 or only those of MUX-6 to MUX-7 when the PP4 to PP7 or the PP6 to PP7 are zero; to reduce the transition power dissipation. Figure 10, shows the booth partial product generation circuit. It includes AND/OR/EX-OR logic.

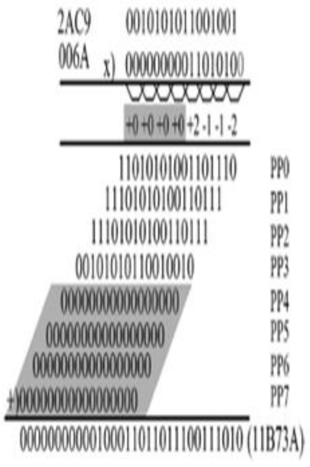


Figure.10 Illustration of multiplication using modified Booth encoding

The PP generator generates five candidates of the partial products, i.e., {-2A,-A, 0, A, 2A}. These are then selected according to the Booth encoding results of the operand B. When the operand besides the Booth encoded one has a small absolute value, there are opportunities to reduce the spurious power dissipated in the compression tree.

Partial Product Generator

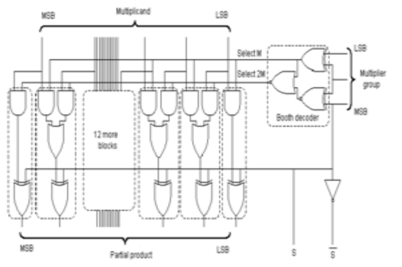


Figure.11: Booth Partial Producselector Logc

The multiplication first step generates from A and X a set of bits whose weights sum is the product P. For unsigned multiplication, P most significant bit weight is positive, while in 2's complement it is negative.

The partial product is generated by doing AND between 'a' and 'b' which are a 4 bit vectors and take four bit multiplier and 4-bit multiplicand get sixteen partial products in which the first partial product is stored in 'q'. Similarly, the second, third and fourth partial products are stored in 4-bit vector n, x,y.

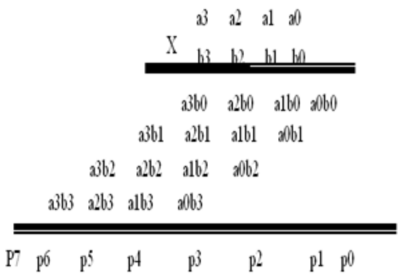


Figure.12: Booth Partial Product Generation

The multiplication second step reduces the partial products from the preceding step into two numbers while preserving the weighted sum. The sough after product P is the sum of those two numbers. The two numbers will be added during the third step The "Wallace trees" synthesis follows the Dadda's algorithm, which assures of the minimum counter number. If on top of that impose to reduce as late as (or as soon as) possible then the solution is unique. The two binary number to be added during the third step may also be seen a one number in CSA notation (2 bits per digit).

VI. RESULTS

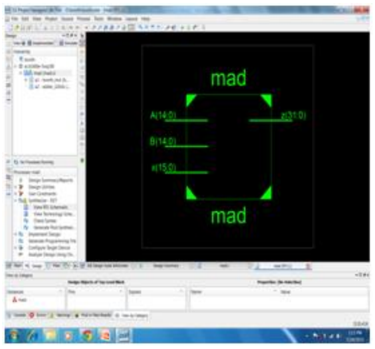


Figure 13: RTL schematic of modified booth recoder

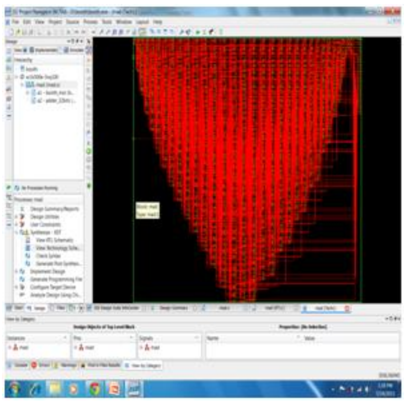


Figure 14: Technology schematic for modified booth multiplier

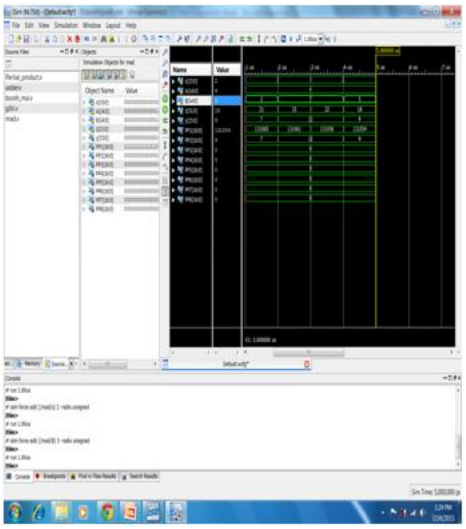


Figure 15: Behavioral simulation results modified booth multiplier

VII. CONCLUSION AND FUTURE SCOPE

Conclusion

This paper focuses on optimizing the design of the Fused-Add Multiply (FAM) operator. This work presents a functional unit which is designed with multiplier-accumulator (MAC), addition, subtraction and sum of absolute difference. Compared to other circuits, the Booth multiplier has the highest operational speed and less hardware count. The basic building blocks for the unit are identified and each of the blocks is analyzed for its performance.MAC unit is designed with enable to block. Using this block, the MAC unit is constructed and calculated for the MAC unit parameters.

We propose a structured technique for the direct recoding of the sum of two numbers to its MB form. We explore three alternative designs of the proposed *S-MB* recoder and compare them to the existing ones and. The proposed recoding schemes, when they are incorporated in FAM designs, yield considerable performance improvements in comparison with the most efficient recoding schemes found in literature.

The presented technique explores its applications in multimedia/DSP computations, where the theoretical analysis and the realization issues are fully discussed. In this project Xilinx-ISE tool is used for logical verification, synthesizing performing placing & routing operation for system verification.

VIII. FUTURE SCOPE

In future it can be extended to floating point numbers also with the supportive EDA tools. By using transistor level implementation for the carry save logic the design reduces the total area required compared to gate level designs. There is chance to improve the speed somewhat more by changing architecture.

IX. REFERENCES

[1] Soojin Kim and Kyeongsoon Cho "Design of High-speed Modified Booth Multipliers Operating at GHz Ranges" World Academy of Science, Engineering and Technology 61 2010.

[2] Magnus Sjalander and Per Larson-Edefors. "The Case for HPM-Based Baugh-Wooley Multipliers," Chalmers University of Technology,Sweden, March 2008.

[3] Z Haung and M D Ercegovac, "High performance Low Power left to right array multiplier design" IEEE rans.Computer, vol 64 no3, page 272-283 Mar 2006.

[4] Aswathy Sudhakar, and D. Gokila, "Run-Time configurable Pipelined Modified Baugh-Wooley Multipliers," Advances in Computational Sciences and Technology ISSN 0973-6107 Volume 3 Number 2 (2010) pp. 223–236.

[5] Myoung-Cheol Shin, Se-Hyeon Kang, and In-Cheol Park, "An Area-Efficient Iterative Modified-Booth Multiplier Based on Self-Timed Clocking," Industry, and Energy through the project System IC 2010, and by IC Design Education Center (IDEC).