

OPTIMIZATION OF POWER IN FUSED ADD MULTIPLY OPERATOR USING MODIFIED BOOTH RECORDER

Dr.B..Gopi¹, G.Kohila²

¹Professor and HOD of ECE, Sona College of Technology, Tamilnadu, India

² PG Scholar, Department of VLSI DESIGN, Sona College of Technology, Tamilnadu, India

Abstract— Addition and multiplication is a crucial arithmetic function for most digital systems. It usually impacts the overall performance of digital systems heavily. In the existing system conventional method of ADD-MULTIPLY (AM) is operator performed separately. The drawback of conventional method is that it requires two adders for AM operation and also inserts a significant delay in the critical path of the AM. So it requires more power, area and hardware complexity. In the proposed method, instead of those adders the HYBRID ADDER MULTIPLIER is designed to achieve high performance and to improve the accuracy, reduction in power consumption and critical delay area of the FAM unit. Therefore the recoding technique were used to implement the direct recoding of the sum of two numbers in its Modified Booth (MB) form and it focuses on the efficient design of FAM operators and also targeting the optimization of the power FAM.

Keywords— Add-Multiply operation, arithmetic circuits, Modified Booth recoding, carry save adder, Carry Look-ahead adder, Hybrid adder VLSI design.

1. INTRODUCTION

Nowadays the hybrid adder multiplier is used in several commercial general purposes. There are many pairs of concurrent addition and multiplication instructions in programs, which can be usually executed in parallel. In the past a majority of researches focused on the reduction of the overall latency with respect to the conventional FAM unit and at the same time the accuracy is increased over the traditional implementation. In many digital signal processing (DSP) and multimedia applications, multiplication and addition are the Most commonly used operations. Therefore, multiply-add fused unit plays an important role in improving performance by combining multiplication and addition operation into a single unit in the modern embedded processors. The research proposes a hybrid add-multiply (HAM) unit that integrates a multiplier and an adder into a single unit. That is, they built a “bridge” circuits between multiplier and adder for combining the results from multiplier and adder as the final output result of HAM.

II. LITERATURE SURVEY

Recent research activities, in this paper focus on AM (ADD-MULTIPLY) operator. The field of arithmetic optimization [1], [2] have shown that the design of Floating point arithmetic components combining which share the data, and to increase the performance. Based on this observation Floating point arithmetic the error can accumulate and greatly affect the computation time and area. And it provides an inaccuracy result in DSP application. Several architectures have been proposed for increase the performance of efficient MAC operation in terms of area and power [3]. But in large set of arithmetic operation [4]–[5]. MAC components increases the flexibility. So the throughput is high. MAC/MAD operation does not depend upon the Add - Multiply operator. Many of DSP application based on the Add-Multiply operator (e.g., FFT algorithm [6]). The ability of distributed arithmetic is to reduce a multiply operation into a series of shifts and additions yields great potential for implementing various DSP systems at a significantly reduced area. Different recoding exists resulting in different gate level implementation and its performances are good. In this XOR-based implementation gives lowest area and delay numbers in most technologies due to the small selector size and the well-balanced signal paths. The addition operation is associative and can accept operands in redundant representation, which allows implementing a sum of multiple products and add - or a sum-of-products (SOP) [8] booth recoding cannot be performed. In [7] author introduced two stage recoder in MB form. First stages were assigned the input bit and second stages of recoding were used for matching the MB digits. Recently [7] these techniques were used in high performances of coprocessor architecture for improve the efficiency. For an conventional method AM unit, multiplication are performed separately. Requires that its inputs and are first driven to an adder and then the input and the sum are driven to a multiplier and produce output. The drawback in adder is an delay in the critical path of the AM [8-9].

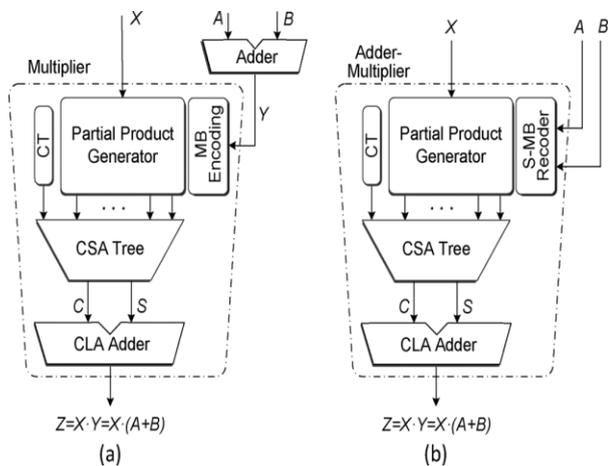


Fig-1: AM operator based on the (a) conventional design and (b) fused design with direct recoding of sum and its MB representation

As noted that [10] focus on FAM design. They introduce a structured and efficient recoding technique and explore three different schemes in FAM design. But it requires two adders, CSA and CLA. By using two adders it requires more area and power. In order to decrease the area and power. We are going to HYBRID ADDER. By implementing the HYBRID ADDER, 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 and fused add multiply, existing recoding schemes are based on complex manipulations in bit-level, which are implemented in gate-level circuits. This work is efficient design on FAM operators using HYBRID ADDER, targeting the optimization of the recoding scheme for direct shaping of the MB form of the sum of two numbers (Sum to MB -S-MB)

III SYSTEM IMPLEMENTATION

3.1 Carry Save Adder

A carry-save adder is used to compute the sum of three or more n -bit numbers in binary. It differs from other digital adders in that it outputs two numbers of the same dimensions as the inputs, one which is a sequence of partial sum bits and another which is a sequence of carry bits.

3.2 Motivation

Consider the sum:

$$\begin{array}{r} 12345678 \\ + 87654322 \\ \hline = 100000000. \end{array}$$

Using basic arithmetic, we calculate right to left, "8+2=0, carry 1", "7+2+1=0, carry 1", "6+3+1=0, carry 1", and so on to the end of the sum. Although we know the last digit of the result at once, we cannot know the first digit until we have gone through every digit in the calculation, passing the carry from each digit to the one on its left. Thus adding two n -digit numbers has to take a time proportional to n , even if the machinery we are using would otherwise be capable of performing many calculations simultaneously.

using bits (binary digits), this means that even if we have n one-bit adders at our disposal, we still have to allow a time proportional to n to allow a possible carry to propagate from one end of the number to the other.

3.3 CARRY LOOK -AHEAD ADDER

A carry look-ahead adder can reduce the delay. In principle the delay can be reduced so that it is proportional to $\log n$, but for large numbers this is no longer the case, because even when carry look-ahead is implemented, the distances that signals have to travel on the chip increase in proportion to n , and propagation delays increase at the same rate. A carry-look ahead adder (CLA) is a type of adder used in digital logic. 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.

Carry look ahead depends on two things:

1. Calculating, for each digit position, whether that position is going to propagate a carry if one comes in from the right.
2. Combining these calculated values to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right.

3.3 Operation

Carry look ahead logic uses the concepts of *generating* and *propagating* carries. Although in the context of a carry look ahead adder, it is most natural to think of generating and propagating in the context of binary addition, the concepts can be used more generally than this. In the descriptions below, the word *digit* can be replaced by *bit* when referring to binary addition. The addition of two 1-digit inputs A and B is said to *generate* if the addition will always carry, regardless of whether there is an input carry (equivalently, regardless of whether any less significant digits in the sum carry). For example, in the decimal addition $52 + 67$, the addition of the tens digits 5 and 6 *generates* because the result carries to the hundreds digit regardless of whether the ones digit carries (in the

example, the ones digit does not carry (2+7=9)). In the case of binary addition, $A + B$ generates if and only if both A and B are 1. If we write $G(A, B)$ to represent the binary predicate that is true if and only if $A + B$ generates, we have:

$$G(A, B) = A \cdot B$$

The addition of two 1-digit inputs A and B is said to *propagate* if the addition will carry whenever there is an input carry (equivalently, when the next less significant digit in the sum carries). For example, in the decimal addition $37 + 62$, the addition of the tens digits 3 and 6 *propagate* because the result would carry to the hundreds digit if the ones were to carry (which in this example, it does not). Note that propagate and generate are defined with respect to a single digit of addition and do not depend on any other digits in the sum.

In the case of binary addition, $A + B$ propagates if and only if at least one of A or B is 1. If we write $P(A, B)$ to represent the binary predicate that is true if and only if $A + B$ propagates, we have:

$$P(A, B) = A + B$$

Sometimes a slightly different definition of *propagate* is used. By this definition $A + B$ is said to propagate if the addition will carry whenever there is an input carry, but will not carry if there is no input carry.

Fortunately, due to the way generate and propagate bits are used by the carry lookahead logic, it doesn't matter which definition is used. In the case of binary addition, this definition is expressed by:

$$P'(A, B) = A \oplus B$$

For binary arithmetic, *or* is faster than *xor* and takes fewer transistors to implement. However, for a multiple-level carry lookahead adder, it is simpler to use $P'(A, B)$.

Given these concepts of generate and propagate, when will a digit of addition carry? It will carry precisely when either the addition generates *or* the next less significant bit carries and the addition propagates. Written in boolean algebra, with C_i the carry bit of digit i , and P_i and G_i the propagate and generate bits of digit i respectively,

$$C_{i+1} = G_i + (P_i \cdot C_i)$$

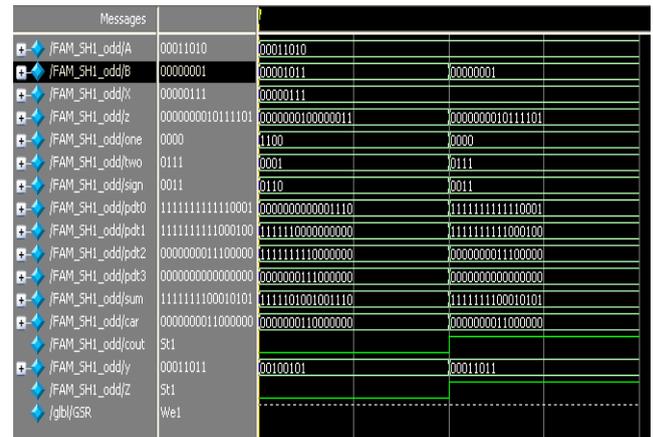
3.4 Fused Add- Multiply Operator

A *fused* multiply-add is a floating-point multiply-add operation performed in one step, with a single rounding. That is, where an unfused multiply-add would compute

the product $b \times c$, round it to N significant bits, add the result to a , and round back to N significant bits, a fused multiply-add would compute the entire sum $a + b \times c$ to its full precision before rounding the final result down to N significant bits. Fused multiply-add can usually be relied on to give more accurate results. as $((x \times x) - y \times y)$ using fused multiply-add, then the result may be negative even when $x = y$ due to the first multiplication discarding low significance bits. This could then lead to an error if, for instance, the square root of the result is then evaluated. When implemented inside a microprocessor, an FMA can actually be faster than a multiply operation followed by an add.

IV RESULTS AND DISCUSSION

We Comparing them with the FAM designs which use existing recoding schemes, the proposed technique system performance in reduction of critical delay in the system level, hardware complexity and power consumption are low in FAM unit. Therefore performance of the proposed recoding schemes is very efficient .It is implemented them using structural Verilog HDL for both cases of even and odd bit-width of the recode's input numbers.



Messages				
/FAM_SH1_odd/A	00011010	00011010		
/FAM_SH1_odd/B	00000001	00001011	00000001	
/FAM_SH1_odd/X	00000111	00000111		
/FAM_SH1_odd/z	0000000010111101	0000000100000011	0000000010111101	
/FAM_SH1_odd/one	0000	1100	0000	
/FAM_SH1_odd/two	0111	0001	0111	
/FAM_SH1_odd/sign	0011	0110	0011	
/FAM_SH1_odd/pdct0	111111111110001	0000000000001110	1111111111110001	
/FAM_SH1_odd/pdct1	1111111111000100	1111110000000000	111111111111000100	
/FAM_SH1_odd/pdct2	0000000011100000	1111111111000000	0000000011100000	
/FAM_SH1_odd/pdct3	0000000000000000	0000000011100000	0000000000000000	
/FAM_SH1_odd/sum	1111111100010101	1111101001001110	1111111100010101	
/FAM_SH1_odd/car	0000000011000000	0000000011000000	0000000011000000	
/FAM_SH1_odd/cout	St1			
/FAM_SH1_odd/y	00011011	00010010	00011011	
/FAM_SH1_odd/Z	St1			
/gbl/GSR	We1			

Fig-2: Simulation of odd for s-mb1. A & B is the input can be recoded in a sum-modified booth algorithm. The output of A & B can be multiply with X partial product can be produced and this partial product can produced the output of Z.

Messages					
/FAM/A	00000101	00000001	00000101		
/FAM/B	00001011	00001011			
/FAM/X	00001100	00001100			
/FAM/z	0000000011000000	0000000010010000	0000000011000000		
/FAM/one	0100	0110	0100		
/FAM/two	0000	0000			
/FAM/sign	0000	0010	0000		
/FAM/pd0	0000000000000000	0000000000000000			
/FAM/pd1	0000000000000000	1111111111001100	0000000000000000		
/FAM/pd2	0000000011000000	0000000011000000			
/FAM/pd3	0000000000000000	0000000000000000			
/FAM/sum	0000000011000000	1111111010001100	0000000011000000		
/FAM/car	0000000000000000	0000000010000000	0000000000000000		
/FAM/cout	S0				
/FAM/y	00010000	00001100	00010000		
/FAM/Z	S0				
/gH/CSR	We1				

Fig-3: simulation of FAM. A & B can be recoded in a sum-modified booth algorithm. The output of A & B can be multiply with X partial product can be produced and this partial product can produced the output of Z.

V CONCLUSION

This paper has proposed a HYBRID ADDER multiplier, fused Add-multiply unit which can sacrifice the accuracy of addition and multiplication operations for saving the power consumption. While performing addition and multiplication operation in the single-precision mode it was also capable of reducing the area ,critical path and hardware complexity. In future it can be implemented in FPGA using Verilog code. The proposed architectures show the best performance compared with the previous method of the FAM unit.

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] L.-H. Chen, O. T.-C. Chen, T.-Y. Wang, and Y.-C. Ma, "A multiplication-accumulation computation unit with optimized compressors and minimized switching activities," in *Proc. IEEE Int. Symp. Circuits and Syst.*, Kobe, Japan, 2005, vol. 6, pp. 6118-6121.

[4] O. Kwon, K. Nowka, and E. E. Swartzlander, "A 16-bit by 16-bit MAC design using fast 5:3 compressor cells," *J. VLSI Signal Process. Syst.*, vol. 31, no. 2, pp. 77-89, Jun. 2002.

[5] 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.

[6] W.-C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 864-874, Mar. 2003

[7] W.-C. Yeh, "Arithmetic Module Design and its Application to FFT," Ph.D. dissertation, Dept. Electron. Eng., National Chiao-Tung University, Chiao-Tung, 2001

[8] R. Zimmermann and D. Q. Tran, "Optimized synthesis of sum-of-products," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, Washington, DC, 2003, pp. 867-872

[9] M. Dumas and D. W. Matula, "A Booth multiplier accepting both a redundant or a non redundant input with no additional delay," in *Proc. IEEE Int. Conf. on Application-Specific Syst., Architectures, and Processors*, 2000, pp. 205-214

[10] C. N. Lyu and D. W. Matula, "Redundant binary Booth recoding," in *Proc. 12th Symp. Comput. Arithmetic*, 1995, pp. 50-57

[11] "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 Pekmestzi, VOL. 61, NO. 4, APRIL 2014

BIOGRAPHIES



Dr.B.Gopi completed his B.E In Electronics and communication Engineering In 1990. There after he joined in three different Industries Textile and Process industries. Due credits in Industries are Erection and commissioning of 1000KVA Transformer, Maintenance of Auto-coner winding Machine. He adds value to his experience by having done Energy auditing to the Textile Mill, to bring down the energy consumption of the same, called as UKG Report, around 2.7units per Kg of production. In the process industry He was in charge of production planning and control of the final product, Sponge Iron. 100 tons per day. He was performing his duty as senior control Engineer and in charge of DCS (Distributed Control System) .Currently he is Head of the Department of Electronics and Communication Engineering , Sona College of Technology, Salem. He has teaching experience of around Seventeen years. His areas of Interests Include Robotics, Embedded system,Nano Electronics. He received his M.E. Degree in applied Electronics and Doctorate in Information and Communication Engineering Covering the area Nano electronics Device. His total experience of Twenty Four Years (Industry -8 , Teaching !6) is fully dedicated for the growth of Young Engineers. He was awarded as Best Project Mentor BY RENASAS in the year 2009 and 2011. He also Received Appreciation award for mentoring the projects BY TEXAS INSTRUMENTS, i n the Year 2015.

He Holds Two patents and has applied for another three.So for He has published six papers in International Journals and four papers in International conferences.



G.Kohila is a PG scholar in VLSI design, at SonaCollege of Technology. She received the B.E degree in Electronics and Communication Engineering from Pavai College Of Technology. She has published papers on various International conferences and International journals. Her area of interests includes VLSI Design, Low power and Digital Electronics.