IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 18, Issue 2, Ser. I (Mar. – Apr. 2023), PP 16-21
www.iosrjournals.org

Fused then Add based Multiplier

N.Samba Murty[1] Harika.V[2] , Thaslima[3], Sumanth.U[4]
[1](Electronics and Communication Engineering, Seshadri Rao Gudlavalleru Engineering College, India)
[2](Electronics and Communication Engineering, Seshadri Rao Gudlavalleru Engineering College , India)
(Electronics and Communication Engineering, Seshadri Rao Gudlavalleru Engineering College , India)

Abstract:
Background*: Nonlinear functions like the discrete wavelet transform (DWT) and discrete cosine transform (DCT) are frequently used in digital signal processing techniques. The speed of addition and multiplication operations has a significant impact on the speed and accuracy of these calculations. The critical path of the entire system may be impacted by multipliers, which typically have the longest delay among basic operational blocks. The modified radix-4 Booth's algorithm (MBA) is frequently used to achieve high-speed multiplication.*
These designs and algorithms can be used in application with the Xilinx 12.3i tool as well as the Verilog language. Booth recoding is a strategy for reducing the amount of incorrect products in multipliers. Different recodings may result in different effectiveness and functionality on the gate-level. Due to its lowered choosing area and well-balanced signal routes, that provide the smallest area and delay amounts in most technologies, the XOR-based near is usually preferred.

Key Word*: Booth recoding, partial products, XOR-based implementation, Fused add multiply operator, carry select adder, Xilinx 12.3i, Verilog.*

Date of Submission: 09-04-2023 Date of Acceptance: 23-04-2023

I. Introduction

Digital Signal Processing (DSP) has widespread applications in areas like multimedia and signal processing. Since DSP applications involve intensive arithmetic operations, the performance of DSP systems can vary depending on the allocation and architecture of arithmetic units. To improve performance, common data sharing techniques can be employed, such as the Divide-Add Fused operation and fused floating point operations.

Multiplication and addition operations are frequently used in DSP applications, often in succession. Dedicated units like Multiply-Accumulator (MAC) and Multiply-Add (MAD) units can improve performance by carrying out multiply-add or add-multiply operations. Add-Multiply (AM) operations are also commonly used, but their direct design can result in increased critical path delay and area. To address this, fusion techniques and Carry Look Ahead (CLA) adders can be used.

Recoding schemes like Modified Booth (MB) and Borrow-Save form can transform input numbers to improve performance, but they can be complex and implemented at the gate level. Efficient Sum to Modified Booth (S-MB) recoders using Radix-4 algorithms have been proposed for implementing AM units, and a modified Radix-8 MB Recoder design has been developed that is simpler, structured, and performs better.

These AM units can be used in Signal Processing applications such as Fast Fourier Transform (FFT), and their design and implementation can have a significant impact on overall system performance.

II. Implementation

Multiplication algorithms using parallel counters, such as the modified Booth algorithm, have been proposed as a way to achieve high-speed multiplication. These algorithms have been implemented in practical applications and are known to operate much faster than array multipliers for longer operands.

Modified Booth Algorithm: Booth multiplication is a technique that is commonly used in digital circuits to perform fast and efficient multiplication. By recoding the numbers that are being multiplied, it is possible to reduce the size and complexity of the multiplication circuit, resulting in smaller and faster circuits.

One popular technique is radix-4 Booth recoding, which allows for the reduction of partial products by half. Instead of shifting and adding for every column of the multiplier term and multiplying by 1 or 0, we only take every second column and multiply by ±1, ±2, or 0 to obtain the same results. This method provides significant advantages in terms of reduced circuit complexity and faster operation.

To implement radix-4 Booth recoding, we consider the bits in blocks of three, with each block overlapping the previous block by one bit. The grouping starts from the LSB, and the first block only uses two bits of the multiplier. By using this technique, we can reduce the number of partial products needed for multiplication, and therefore, reduce the overall circuit complexity.In summary, Booth multiplication is a powerful technique that allows for smaller and faster multiplication circuits. Radix-4 Booth recoding is an important variant of this technique that can reduce the number of partial products by half, resulting in significant advantages in terms of circuit complexity and speed.

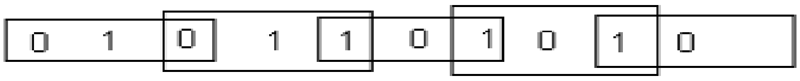


Figure1: Combination of bits from the multiplier term

For example of Booth multiplying two number”2AC9” and “006A”. The outline denotes that all of the numbers in that portion of the Booth multiplication are zero, letting this piece of the equations to be neglected. The power consumption arising in transient signals might be considerably reduced by saving which means computations.

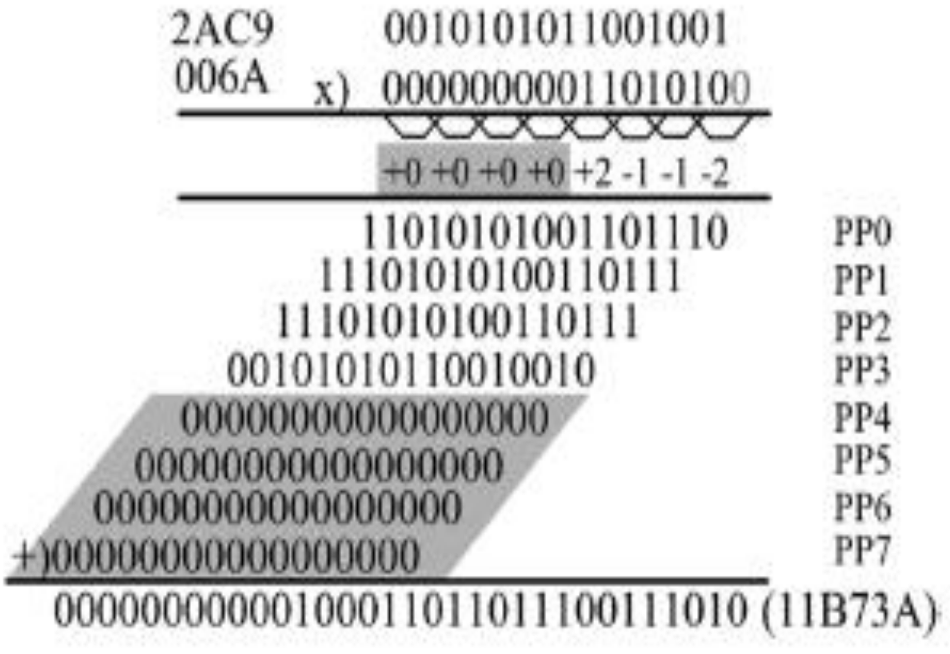


Figure2: Image of multiplication using modified Booth encoding

Here in Modified Booth Algorithm technique we have Radix-16,Radix-8,Radix-4 techniques which is used to reduce partial products.

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

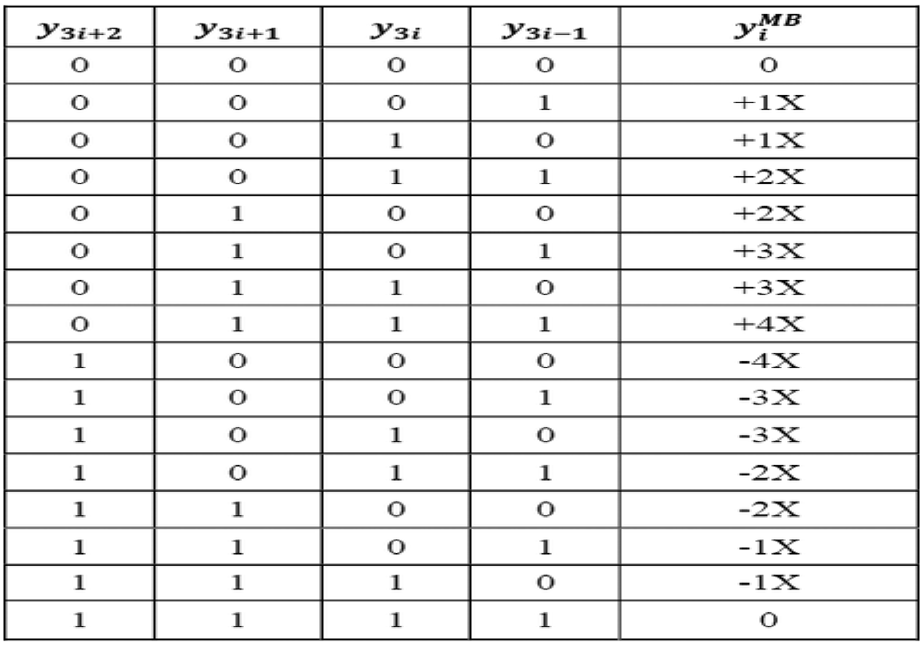
Table1:Operation on the Multiplicand

Partial Product Generation:

Partial products are an essential component of multiplication algorithms, and they are generated based on encoding schemes such as radix-8 encoding. In this encoding scheme, the encoding of a number X is got by lifting X to the left by one position to obtain the value of 2X. Similarly, the value of 4X is by ever changing X to the left by two positions, and the value of -X is acquired by giving the 2's complement of X.

The value of -2X is taken by changing-X to the left by one position, and the value of -4X is acheived by removing -X to the left by two positions. To obtain the values of 3X and -3X, we add pre-computed values to X and -X, respectively. Specifically, we add the value of 2X to X to obtain 3X, and the value of -2X to -X to obtain -3X.

These operations can be implemented using a 16:1 multiplexer. The radix-8 modified booth encoding table can be used to determine the values to be selected by the multiplexer for a given input. The table specifies the values of X, 2X, 4X, -X, -2X, -4X, 3X, and -3X for all possible 3-bit inputs.



y_{3i+2}	y_{3i+1}	y_{3i}	y_{3i-1}	y_i^{MB}
0	0	0	0	0
0	0	0	1	+1X
0	0	1	0	+1X
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	-1X
1	1	1	0	-1X
1	1	1	1	0

Table2: Radix-8 Booth Algorithm

COMPRESSOR

The operation of a compressor is based on the principle of carry lookahead. The carry lookahead technique is used to determine the carry bits that are generated when adding two numbers. By predicting the carry bits in advance, we can speed up the addition process and reduce the number of logic gates required for the operation.In a m:n compressor, the input bits are divided into groups of size m. Each group is added using the carry lookahead technique to produce n output bits. The output bits from each group are then combined by means of adder final result.The efficiency of a compressor is determined by its delay and area requirements. The delay is the time required for the circuit to produce a valid output after the input has been applied.

Fused Add Then Based Multiplier:

The output of this operation is then rounded to the desired precision.One of the primary benefits of using FMA-based multipliers is improved accuracy. Because the multiplication and addition operations are achieved together in a one step, the exactness of the calculation is improved, and the risk of rounding errors is reduced. This is especially important for complex calculations that require high precision.Another benefit of FMA-based multipliers is improved performance. By performing both multiplication and addition in a single instruction, the number of instructions executed is reduced, resulting in faster computation times. In a traditional multiplier, the multiplication and addition operations are performed separately. The result of the multiplication is stored in a register and then added to another value in a subsequent step. However, in an FMA-based multiplier, which reduces the number of series required to execute the operation.The FMA operation can be used for a variety of arithmetic operations, including floating-point multiplication and addition, and is widely used in scientific simulations, financial modeling, and other applications that require high precision and fast calculations.

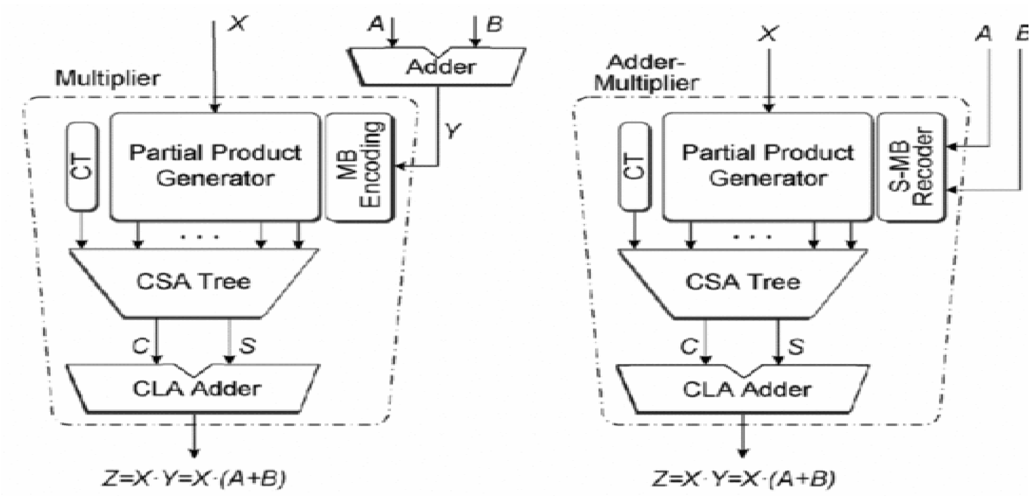


Figure4: (a) conventional design and (b) fused design using direct sum to modified booth recoding

III. Result

Table no 3 Shows Comparison of Fused then Add based Multiplier. The table consists of Existing,Radix-16,Radix-8,Radix-4 with Number of Slice Registers having 0,0,40,40 , Number of Slice LUTs having 399,,192,177,107, Number of Occupied Slices as 204,99,72,55 ,Number of Bonded IOBs as 46,46,64,64, Delay(ns) in 26.752,15.306,2.560,2.560 and Power(w) as 0.529,0.529,0.530,0.529.

Table no 3: Shows Comparison of Fused then Add based Multiplier.

	Existing	Radix-16	Radix-8	Radix-4
Number of Slice Registers	0	0	40	40
Number of Slice LUTs	394	192	177	107
Number of Occupied Slices	204	99	72	55
Number of bonded IOBs	46	46	64	64
Delay(ns)	26.752	15.306	2.56	2.56
Power(w)	0.529	0.529	0.53	0.529

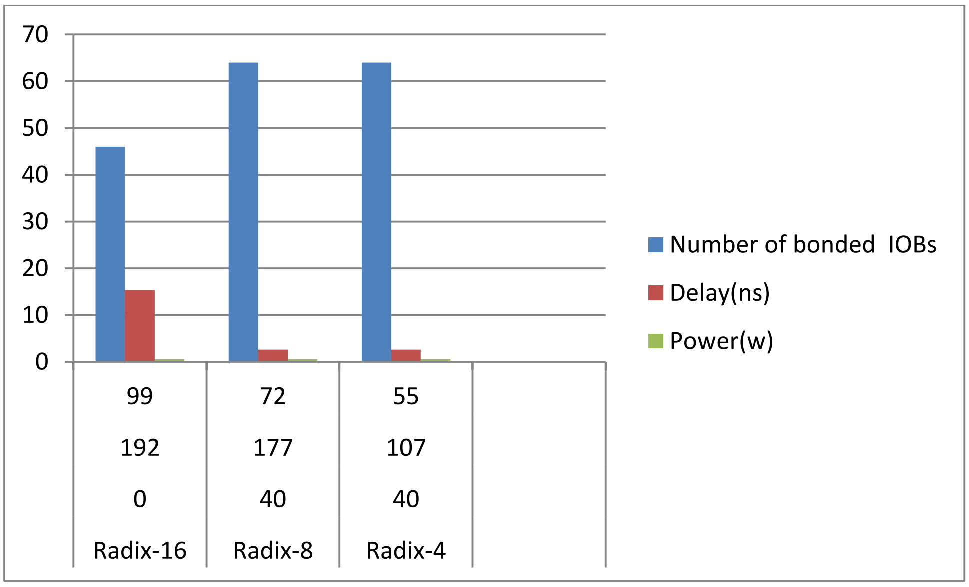
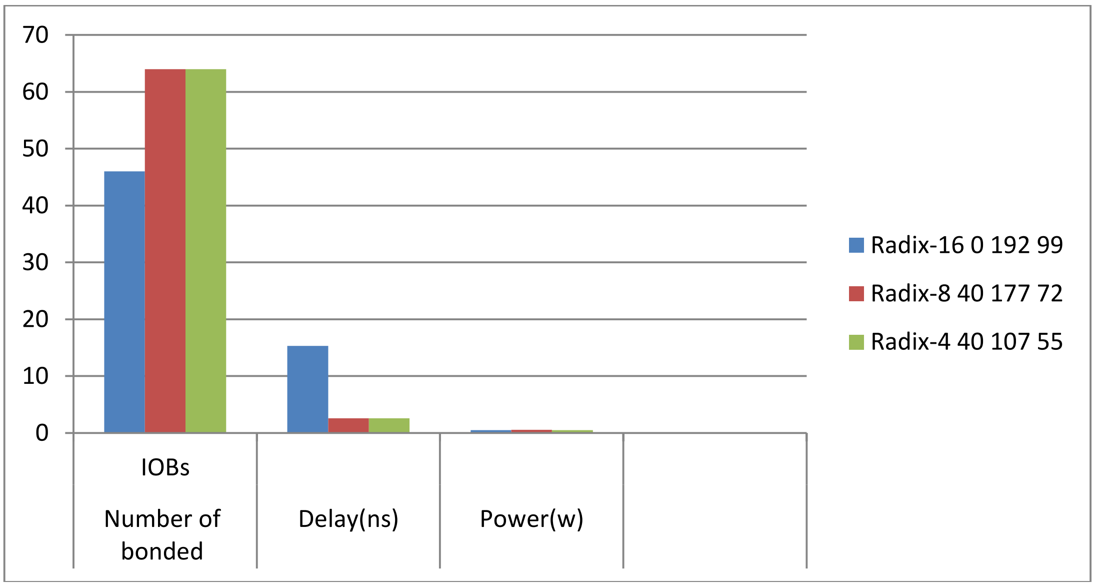


Table 4 shows the comparison of Radix-16,Radix-8,Radix-4 with Number of Slice Registers having 0,40,40 , Number of Slice LUTs having ,192,177,107, Number of Occupied Slices as 99,72,55 ,Number of Bonded IOBs as 46,64,64, Delay(ns) in 15.306,2.560,2.560 and Power(w) as 0.529,0.530,0.529.

Table4: Comparison of Radix-16,Radix-8,Radix-4

	Radix-16	Radix-8	Radix-4
Number of Slice Registers	0	40	40
Number of Slice LUTs	192	177	107
Number of Occupied Slices	99	72	55
Number of bonded IOBs	46	64	64
Delay(ns)	15.306	2.56	2.56
Power(w)	0.529	0.53	0.529



IV. Discussion

The challenges of producing a high-performance multiplier for uses in digital signal processing, which often significantly rely on multiplication steps. The primary three stages of multiplication generate part products, lower partial products, and finding the final product. The hardest step, that additionally impacts the multiplier's total speed as it contributes to the unit's entire delay and power consumption, is the decrease of partial products.

Higher order compressors were built by researchers to shorten the processing unit's critical path delay and achieve outstanding results in the partial result reduction stage. However, the area and latency situations of the multiplier could grow if adders are included in the reduction process. Compressors are included within the multiplier to address this issue.

The authors propose the use of the Sum to MB (S-MB) algorithm to optimize the recoding scheme for direct shaping of the MB form of the sum of two numbers. Three different recoding techniques (S-MB 1, S-MB2, and S-MB3) are utilized, and it is found that S-MB2 is the most efficient technique in terms of hardware complexity, power dissipation, and processing time. The authors highlight that combining operations in the design of arithmetic components can lead to significant performance enhancements, and their proposed technique provides a novel approach to optimizing the partial products reduction stage in the FAM unit.

V. Conclusion

Comparison among all FAM designs, in terms of area and power consumption respectively, for even bit-width. For each case that were explored, focus on the lowest clock period where all FAM designs are synthesized

References

[1]. A.Amaricai,M.Vladutiu,andO.Boncalo,“Designissuesandimplementations for floating-point divide-add fused,” IEEE Trans. Circuits Syst.
[2]. E. E. Swartzlander and H. H. M. Saleh, “FFT implementation with fused floating-pointoperations,” IEEE Trans. Comput.
[3]. J.J.F.Cavanagh,DigitalComputerArithmetic.
[4]. S.Nikolaidis,E.Karaolis,andE.D.Kyriakis-Bitzaros,“Estimationof signal transition activity in FIR filters implemented by a MAC archi- tecture,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.
[5]. O.Kwon,K.Nowka,andE.E.Swartzlander,“A16-bitby16-bitMAC designusingfast5:3compressorcells,”J.VLSISignalProcess.
[6]. L.-H.Chen,O.T.-C.Chen,T.-Y.Wang,andY.-C.Ma,“Amultiplica-tion-accumulation computation unit with optimized compressors and minimizedswitchingactivities,”inProc.IEEEInt,Symp.Circuitsand Syst.

[7]. Y.-H. Seo and D.-W. Kim, "A new VLSI architecture of parallel multiplier–accumulatorbasedonRadix-2modifiedBoothalgorithm," IEEE Trans. Very Large Scale Integr. (VLSI) Syst.

[8]. A. Peymandoust and G. de Micheli, "Using symbolic algebra in algo- rithmic level DSP synthesis," inProc. Design Automation Conf.

[9]. W.-C. Yeh and C.-W. Jen, "High-speed and low-power split-radix FFT," IEEE Trans. Signal Process.

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

[11]. J. D. Bruguera and T. Lang, "Implementation of the FFT butterfly with redundant arithmetic," IEEE Trans. Circuits Syst.Il, Analog Digit. Signal Process.

[12]. W.-C. Yeh, "Arithmetic Module Design and its Application to FFT," Ph.D.dissertation,Dept.Electron.Eng.,NationalChiao.

[13]. R.Zimmermann and D.Q.Tran,"Optimized synthesis of sum-of-prod- ucts," in Proc. Asilomar Conf. Signals, Syst. Comput., Pacific Grove.

[14]. B. Parhami, Computer Arithmetic: Algorithms and Hardware De- signs

[15]. O. L. Macsorley, "High-speed arithmetic in binary computers," Proc. IRE.

[16]. A Low Power Radix-4 Booth Multiplier With Pre-Encoded Mechanism.

N.Samba Murty, et. al. "Fused then Add based Multiplier." *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)* 18(2), (2023): pp 16-21.