ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design of Advanced MAC Using Improved Radix-4 Architecture

Shaik Akthar Basha, M.Tech,
ECE department,
Guntur Engineering College, Guntur,
Andhra Pradesh, India.

Sri S.Suresh babu,
Asst. Prof. ECE dept,
Guntur Engineering College, Guntur,
Andhra Pradesh, India.

ABSTRACT-

In order to perform the fast operations, unwanted area and power concern to intermediate storage of individual multiplication products, multiply and accumulate is the promising alternative for the different DSP applications. As we know the multiplication is more power consumption by its architecture, in order to improve the feature of the multiplier different methods adopted, in that radix booth algorithm is more promising alternative related to the area, power and speed consideration.Radix-4 booth algorithm can be well suited for the small bit length applications like 8,16 bits. More over if advanced methods like CSA multi-operand addition improves further more advancement related to the significant specifications.

Keywords: ***DSP Applications, multiplication, radix and multiplier-and-accumulator (MAC).***

I. INTRODUCTION

With the present rapid betterments in multimedia and communication systems, real-time signal processing applications such as audio signal processing, video/image processing, or large-capacity data processing are increasingly being necessitated. The multiplier and multiplier-and-accumulator (MAC) are the crucial elements of the digital signal processing like filtering, convolution, and inner products. More digital signal processing methods use nonlinear functions like DCT or DWT. Because they are fundamentally accomplished by repetitious application of multiplication and addition, the speed of the multiplication and addition arithmetic's decides the functioning speed and performance of the total calculation. Because the multiplier provides the longest delay among the different operational blocks present in digital system, the critical path is ascertained by the multiplier; normally for high-speed multiplication radix-4 modified Booth's algorithm (MBA) is promising approach. However, this cannot entirely solve the problem due to the long critical path occurred in multiplication.

In general multiplier basically constitute with the basic elements such as partial product generation and partial product addition. For the partial product addition generally we use array of the full adders, walles trees or different adder trees also considered. Form the all possible alternatives CSA tree is the most promising solution for the improved partial product addition.
Multi-operand addition is essential in lot of design like Multiplication, SAD, MAC, filters and others. In order to achieve greater executions they rely on the, redundant adders.

This makes the reduction of addition time by reducing the propagation of carry chains. Carry-save (CS) and signed-digit (SD) are more familiar carry chain representations. In CS method the groups of full adders are used without carry propagation between them. This reduces the one row per level i.e it acts as 3:2 counter /compressor. More over compressor based multi-operand addition also carried out by using the compressors like 4:2 and 5:2 etc.

Even so the better performance of CPA's is extensively and these redundant adders are refused for FPGA implementation with reference to the area consideration and also speed also not that much satisfied.

This paper includes the Introduction and booth algorithm description as the section II, proposed method as the section III followed by the results, conclusion and references.

II. MODIFIED BOOTH ALGORITHM

Modified booth algorithm is one of the promising algorithms for the design of advanced multipliers which can reduce the partial product count. There by not only the reduce of area, power of partial product generator and also the increases the addition features There are different modified booth algorithms which selected based on the requirement and also concern with the bit length for multiplication. First generation of booth algorithm can be modified with certain

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

changes related to the advancement in fast processing accomplished by the modified booth algorithm.

Radix-2 is the basic booths algorithm for multiplication with this the generation of the partial products is different compared with the traditional multiplication algorithm but it does not reduce any partial products. In general smaller bit lengths we use the radix-4 booth algorithms. In radix -4 booth algorithm it reduces the partial products by two i.e. N/2. The generations of partial products are in different pattern there by the area and power of the partial products reduces and considerable speed improvement also noticed.

III.PROPOSED METHOD

In our proposed method we exploited a novel multiply and accumulate (MAC) unit which can be suited for the different application. MAC unit internally consists of multiplier as the fundamental unit. For multiplication we use the radix-4 unsigned booth algorithm because in general we don't deals with the negative number multiplication method because in out multiplication MSB is not concern with the sign of that particular numbers. There by we can reduce the area and bit count that related to the signed multiplication.

In traditional methods before the invention of the parallel MAC unit the operation performed by the repeated multiplications and that resultant multiplications can be stored in the temporary memories. The addition performed after completion of the all multiplications. Based on the bit length the intermediated memories also increase there by the area and power increases, and more over the number of additions also increases and addition circuits also increases.

In order to avoid the all the difficulties that occurred in the traditional methods can be overcome by the new parallel addition mechanism, which can be described in Fig.1.

The below architecture performed parallel addition. First multiplication performed and is stored in memory and when adding the second multiplication partial product first multiplied result also added, when performing third multiplication their partial product along with accumulated result also added, and the operation continuous until the last input pair.

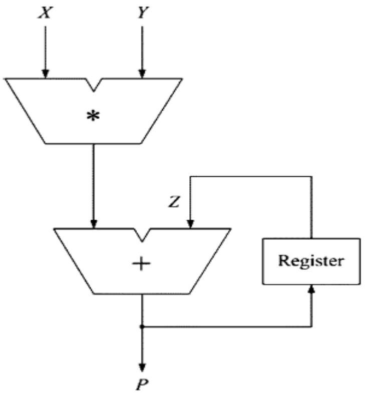

Figure1. Proposed MAC architecture.

The multiplication is segmented as the partial product generation and tree addition and the final addition, here as discussed above the radix-4 unsigned algorithm used for the partial product generation and compressor can be used for the tree addition and the advanced adder can be used for the final addition.

In this the booth recoding can be of three bits each starts for either LSB to MSB or MSB to LSB by the over lapping each group numbers that are multiplied. By this we can reduce the number of partial products by half, by radix-Booth recoding technique can be considered as the partial product generated.

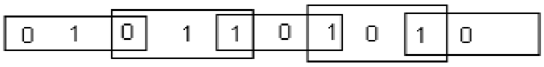

Figure2. Grouping of bits from the multiplier term

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 1.

Form the table we noticed that there are two possible operations 1x and 2X along with sign.

TABLE 1
Redundant value for group pairing

Block	Recoded digit	operation
000	0	0
001	1	1X
010	1	1X
011	2	2X
100	-2	-2X

101	-1	-1X
110	-1	-1X
111	0	0

TABLE 2
Operation of Redundant value

1X	same number (multiplicand)
2X	left shift by one position
-1X	Twos compliment of 1X
-2X	Twos compliment of 2X

.

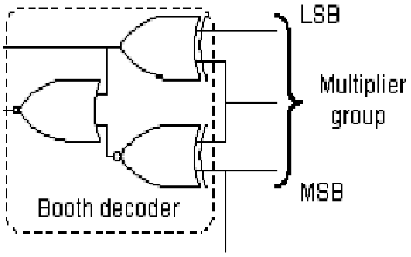

Figure3. Booth decoder

Partial product generation is designed with two modules like booth decoder and booth selector. Booth encoder takes the corresponding group and provides the redundent bit alog with sign, booth selector slects the required resultent as the partial product out

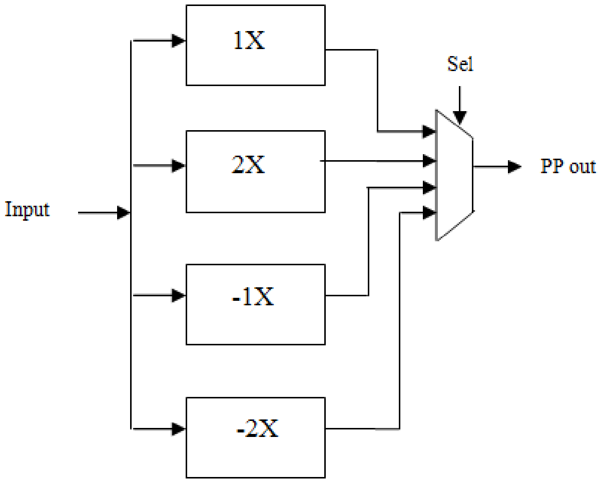

Figure4. Booth selector

Booth decoder and booth selector operational diagrams shown in figure 3&4

Generally the multiplier is taken for the groping as discussed above three bits overlapped grouping performed and individual grops given as input to the each booth decoders and that generates the rescpecteive secection for the booth selector. Booth selector selects the corresponding partial product as the output by performing the appropriate operation.

IV.RESULTS AND DISCUSSION

Our proposed method can be designed by using the verilog HDL and simulated by using the Model sim simulator and Synthesis can be done by Xilinx. We take a and b as input and it produces 4 partial products p0,p1,p2,p3, partial product compression can be performed using the CSA array and final summation can be done by Ripple-carry adder. That produces the output final sum.

For simulation initial we take a=0011 and b-=0100 that produces the partial products as p0=111111100, p1=000000100, p2=000000000 and p3=000000000 and produce the multipled result as 0000000000001100 and for first final result also 0000000000001100.

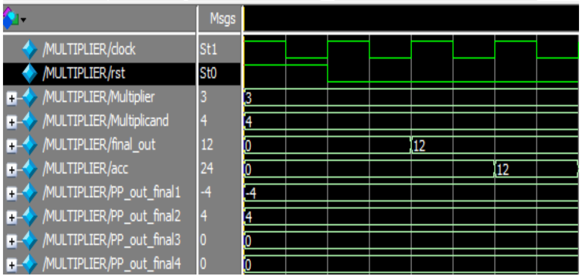
Msgs
/MULTIPLIER/clock St1
/MULTIPLIER/rst St0
/MULTIPLIER/Multiplier 3
/MULTIPLIER/Multiplicand 4
/MULTIPLIER/final_out 12
/MULTIPLIER/acc 24
/MULTIPLIER/PP_out_final1 -4
/MULTIPLIER/PP_out_final2 4
/MULTIPLIER/PP_out_final3 0
/MULTIPLIER/PP_out_final4 0
3
4
0
12
0
12
-4
4
0
0

Figure5. Initial input Simulation

For simulating second input combination we take a=1111110 and b-=111110 that produces the partial products as p0=110000100, p1=000000000, p2=000000000, p3=0011111101 and produce the multipled result as 1111010000100 and final result is 1111010010000

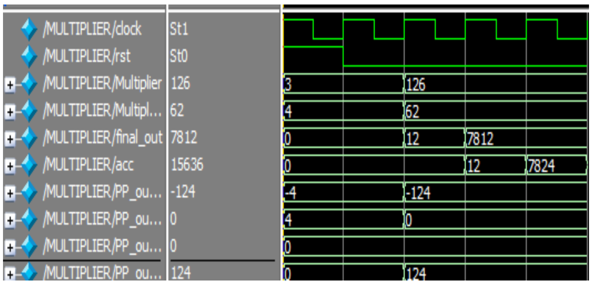

Figure6. Second input Simulation.

V.CONCLUSION

By our new architecture for fast multiply and accumulation calculation by using the advance multiplication method called radix-4., more changes like CSA array addition were adopter for the improvement to the advanced VLSI considerations. Our proposed design gives the considerable improvement towards the area and speed.

In future work, we plan to explore the use of advanced radix mechanisms for longer bits and advanced addition strategies for improved multiplication.

REFERENCES

[1] D. A Pucknell, K. Eshraghain, Basic VLSI Design, Prentice-Hall, ISBN 81-203-0986-3.

[2] C.R.Baugh ,B.A.Wooly, A two's complement parallel multiplication algorithm IEEE Transaction on computers, vol, c-22,no.12,December 1973.

[3] Israel Koren, Computer arithmatics algorithms A.K.Peters Ltd. ISBN 1568811608.

[4] A.D.Booth, A signed binary multiplication technique, Quarterly Journal of Mechanics and Applied mathematics, vol-IV,pt-2-1951.

[5] Rajendra katti, A Modified Booth Algorithm for High Radix Fixed-point Multiplication, IEEE Transaction on very large scale integration VLSI systems,vol-2,no.-4 December-1994.

[6] Wen-Chang Yeh, Chein-Wei Jen, High speed booth encoded parallel multiplier design, IEEE transaction on computers, vol-49, no-7.July-2000.

[7] Eduardo Costa, Sergio Bampi, Jos´e Monteiro, a New Architecture for 2's Complement Gray Encoded Array Multiplier, Proceedings of the 15th Symposium on Integrated Circuits and Systems Design (SBCCI'02).

[8] A.R. cooper, Parallel architecture modified Booth multiplier, IEEE proceedings.vol-135.pt-G.No.3June 1988.

[9] Sanjiv Kumar Mangal, Rahul M. Badghare, Raghavendra B. Deshmukh, R. M. Patrikar, FPGA Implementation of Low Power Parallel Multiplier, 20th International Conference on VLSI Design (VLSID'07).