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

Design of the Radix-10 Based BCD Multiplier Based on Multi-Operand Addition

Nagarjuna Amara
M.Tech Student,
Department of ECE,
Guntur Engineering College,
Guntur, AndhraPradesh, India.

K.Syam Babu, M.Tech, Ph.D
Asso.Prof,
Department of ECE,
Guntur Engineering College,
Guntur, AndhraPradesh, India.

Abstract:

There are Different algorithms that are related to the generation of the fast calculations and also the different approaches for the accurate result generation. Likewise BCD representation, fixed point representation and etc are familiar in present designing. For designing the fast and accurse output generation they relay on the methods like booth algorithms and advanced addition methods were introduces for the increasing the addition speed and reducing the die foot print.

In this paper we are designing the novel BCD multiplication mechanism for the applications like commercial and user defined applications by using the advanced booth encoding called RADIX-10 booth multiplier as part of multiplier along with the fast partial product redundant addition called compressor tree. This reduces the area and power overhead conditions and suited ideal for the any application. Is designed using the verilog hdl and simulated using the model-sim and synthesized by using the Xilinx

I INTRODUCTION:

There are different applications that are required BCD arithmetic is the part of it because it has the feature of easy understanding and calculations. But the major drawbacks that influence the advantage of this format are the backward in speed and area. There are different researches are going on this area Even in the present design methods the multiplication itself the critical concern there wise the many methods are introduces for the improvement in basic VLSI consideration. In traditional multiplication internally is the different staged compositor, namely partial product generation, multi-operand addition and final sum generation. In previous attempts different addition strategies and In previous attempts different addition strategies and methods were introduces for the multiplier designing. After introduction of booth algorithms the multiplier designing can changed drastically.In general, a multiplier uses Booth's algorithm and array of full adders (FAs), or Wallace tree alternatively to array of FA's., i.e., in general multiplier mainly compose of the three parts: Booth encoder, compression tree for the partial product addition such as Wallace tree, and final sum adder. Because Wallace tree is the parallel addition to make faster processing, it is also related to the operation time proportionally. It uses the fact that counting the number of 1's among the inputs sickens the number of outputs into. In real execution, many counters are used to compress the number of outputs in each stage. An efficient method for the increase of speed by limiting the partial product count, in general multiplication internal consists stages of additions of partial products. To reduce the internal calculation of addition levels for the partial products,

It can be efficient where the addition methods like walles tree MBA algorithm has been applied for the increase in the speed to generation and addition of the partial products. Many parallel architecture are introduces the speed with MBA algorithm, many pipelined multiplication architectures works parallel have been researched. Among them, the design methods based on the Baugh–Wooley algorithm (BWA) have been developed and they have been applied to various digital filtering calculations. No any other improved design in reference to the BCD multiplier. Normal traditional based methods are introduced for the approaches to the design. This paper organizes as follows. In Section I deal with the introduction, followed by the session II that deals with booths algorithm, Session III covers novel multi-operand addition, session IV deals with proposed technique, Results and analysis of our architecture with the proposed techniques is covered in Section V. And Section VI concludes the paper.

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

II BOTH ALGORITHMS:

Different researches are introduced in the design of multiplier concern to the partial product addition only, but no other methods are concentrated on the partial product generation.When proper method for the partial product count reduction can makes efficient multiplier design because it not only reduces the count of the partial product count but also the increases the addition features. That makes the booth algorithms is promising and alternative approach for the efficient multiplier design.

There are different modified booth mechanisms are using based on the requirement of user and bit lengths we are introducing the radix-10 modified booth recording for the partial product generation .

III NOVEL MULTI-OPERAND ADDITION:

In the previous approach, speculated carry resources are only used in the design of a single 4:2 compressor, but the whole compressor tree structure is not designed using these resources. In order to minimize the usage of carry resources the new tree structure that can be similar to that of classic linear array of CSAs.

In this case only the intermediate carry only connected to the next and the respected carry and sum words are forwarded to the lower level stages. Here we proposed the different methods for CS compressor trees efficient mapping in FPGA. In advance the analysis of area and power also conducted for general case.

Refer to the area, the generic compressor tree implementation based on N bit width, CSAs requires Nop _ 2 of these elements (because each stage eliminates one input signal). Therefore, considering that a CSA could be implemented using the same number of resources as a binary CPA (as shown below), with over proposed linear 4:2 compressor array, and the hardware cost vise also the binary CPA tree have approximately the same.

But in whole compressor tree structure design the resources have not been considered. Here we are proposing the compressor tree similar to that of linear CSAs array.

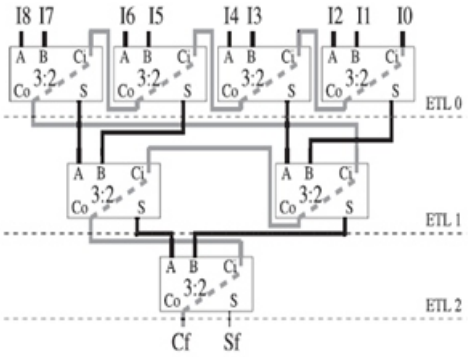


Fig.1. Time model of the proposed CS 9:2 compressor tree.

More over in given the sum-word and carry-word of respected adder the carry word is forwarded to next and sum word is connected to the next lower stagesBy fig 2, based on the liner structure the 9:2 counters is designed, where every line is individually of N bit width and the respected carry's are shifted. Where as in CSA, we need to separate the primary inputs and the carry input, the dotted line represents the fast carry resources. Where as in first CSA Ci can be connected to bring in an input operand, in every

CSA the previous carry output connected to the Ci. As primary The two inputs of each CSA are adds all input operands .when the all input operands are introduced in array .previously generate and sum word are added .by their as we minimizes the regular signals and carry-chains propagation As reckoning the area of the N bit width CSA that takes Nop-2 of logical elements. (reduces only one input the per stage) .on consideration the CSA implemented using the same resources, out proposed method the compressor and binary tree that they are utilizing the same amount of hardware cost
In the analysis of delay, for the reference point out design tree has Nop-2 levels. This is greater than the others critical path by this consideration the in critical path connation the carry sign al is discarded and hence out tree looks as the hypothetical tree,

IV PROPOSED METHOD:

In out proposed architecture design for designing the BCD coded multiplier we need to take BCD coded inputs and we need to design the intermediate multiplier with less power and area not only for the ASIC design and for the FPGA's also at the end again we need convert onto the BCD result.

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

For the proposed design method internally consists of basic three stages.
1)Stage one consists the partial product generation by using the radix-10 modified booth algorithm.
2)Stage two includes the multi-operand addition by using the compressor tree
3)Third stage again convert the temp sum in to the BCD output

a) Partial product generation :

The partial product generation by using the advanced radix method it internally consists of booth encoder and booth selector, booth encoder takes input group and produces the partial redundant bits that will give to the booth selector for the generation of the single partial product bit.It produces the partial redundant values +5 to -5 and the corresponding operations table is shown in table I.

b) Multi-operand addition and final addition:

The partial products generated by the partial product generator are added by the multi-operand addition and final addition. Multi-operand addition can be carried out by the compressor tree with 4:2 and 3:2 counters. Final addition can be carried out by the parallel pre-fix product addition. Parallel prefix adder is the advanced with less power and improved speed.

Table I: Partial redundant representation

Redundant value	operation
0	Result will be zero
1	Same as x
2	X is left shift by one
3	X is left shift by one is added with x
4	X is left shift by two
5	X is left shift by two is added with x
-1	X is complimented
-2	X is left shift by one and complimented
-3	X is left shift by one is added with x and complimented
-4	X is left shift by two and complimented
-5	X is left shift by two is added with x and complimented

c) Final conversion to BCD:

The simple method of converting the resultant X and Y multiplier is by adding the '6' to the resultant sum. Generally the BCD number is excess-6 than the binary value. We need to add the '6' to the resultant sum. But for the design we use the BCD carry save adder.These stages are inter connected with each otherpartial product generator followd by the partial product addtion it internally conisists the multi-operand addtion and final addition that will converted by the finalmstage in to the BCD format.Block diagram of the proposed method is shown in the below fig

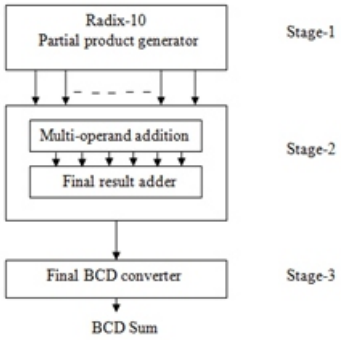
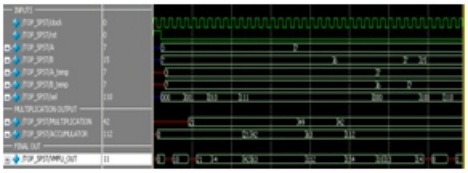


Fig.2: Block diagram of proposed multiplier

V RESULTS AND DISCUSSION:



The partial products are generated by using the Radix-10 modified booth algorithm and the [partial product addition can be carried out by using the advanced novel multi-operand mechanism final addition also carried last converted into BCD format

VI CONCLUSION:

BCD multiplier is designed with the advanced partial product generation using with Radix-10 booth recording mechanism and multi-operand addition can be carried out by the compressor tree and final conversion also can be carried out by the advanced BCD adder. By using the above mentioned method an efficient design is proposed with considerations of the area and power and speed.

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

In general when compare to that f the normal multiplier BCD multiplier takes more area and power but also important more delay. By our design the approximate reaches to the multiplier aspects can be obtained with this design .

REFERENCES:

[1]Aswal, M. G. Perumal, and G. N. S. Prasanna, "On basic financial decimal operations on binary machines," IEEE Trans. Comput., vol. 61, no. 8, pp. 1084–1096, Aug. 2012.

[2]M. F. Cowlishaw, E. M. Schwarz, R. M. Smith, and C. F. Webb, "A decimal floating-point specification," in Proc. 15th IEEE Symp. Comput. Arithmetic, Jun. 2001, pp. 147–154.

[3]M. F. Cowlishaw, "Decimal floating-point: Algorism for computers," in Proc. 16th IEEE Symp. Comput. Arithmetic, Jul. 2003, pp. 104–111

[4]S. Carlough and E. Schwarz, "Power6 decimal divide," in Proc. 18th IEEE Symp. Appl.-Specific Syst., Arch., Process., Jul. 2007, pp. 128–133.

[5]S. Carlough, S. Mueller, A. Collura, and M. Kroener, "The IBM zEnterprise-196 decimal floating point accelerator," in Proc. 20th IEEE Symp. Comput. Arithmetic, Jul. 2011, pp. 139 146.

[6]L. Dadda, "Multioperand parallel decimal adder: A mixed binary and BCD approach," IEEE Trans. Comput., vol. 56, no. 10, pp. 1320–1328, Oct. 2007.

[7]L. Dadda and A. Nannarelli, "A variant of a Radix-10 combinational multiplier," in Proc. IEEE Int. Symp. Circuits Syst., May 2008, pp. 3370–3373.

[8]L. Eisen, J. W. Ward, H.-W. Tast, N. Mading, J. Leenstra, S. M. Mueller, C. Jacobi, J. Preiss, E. M. Schwarz, and S. R. Carlough, "IBM POWER6 accelerators: VMX and DFU," IBM J. Res. Dev., vol. 51, no. 6, pp. 663–684, Nov. 2007.

[9]M. A. Erle and M. J. Schulte, "Decimal multiplication via carrysave addition," in Proc. IEEE Int. Conf Appl.-Specific Syst., Arch., Process., Jun. 2003, pp. 348–358.

[10]M. A. Erle, E. M. Schwarz, and M. J. Schulte, "Decimal multiplication with efficient partial product generation," in Proc. 17th IEEE Symp. Comput. Arithmetic, Jun. 2005, pp. 21–28..