www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 5 May 2015, Page No. 11783-11787

High Speed Fused Add-Multiply Operator Design Based On SQRT-CSLA Adder

Vidya Baby,
PG scholar,
Department of Electronics and,
Communication Engineering,
Sree Sakthi Engineering College,
Coimbatore
mail:vidyanbaby90@gmail.com

***Abstract*— Digital Signal Processing (DSP) applications involve various complex operations. This paper deals with the high speed operation in the design of the Fused-Add multiply operator for increasing the Performance. This includes the techniques to implement the direct recoding of the sum of two numbers in the Modified Booth (MB) form and involves the high speed addition using the SQRT-CSLA adder. This paper makes use of all the three structured and efficient recoding technique by means of three different schemes available in FAM designs. In comparison to the existing schemes, the proposed technique yields considerable reductions in terms of critical delay, hardware complexity and power consumption of the FAM unit.**

***Keywords*— Add-Multiply Operation, SQRT-CSLA Adder, arithmetic circuits, Modified Booth recoding, VLSI design.**

I. INTRODUCTION

Digital Signal Processing (DSP) applications find extensive use in modern consumer electronics providing custom accelerators for the domains of multimedia, communications etc. Typical DSP applications involves a large number of arithmetic operations as their implementation is based on computationally intensive kernels, such as Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), Finite Impulse Response (FIR) filters and signals' convolution. The performance of such DSP systems is inherently affected by decisions on their design regarding the architecture and allocation of its arithmetic units. Recent studies shows that the design of arithmetic components combining operations which share data, can lead to the better performance improvement. Based on the results that an addition can often be subsequent to a multiplication, which is used in the Multiply-Accumulator (MAC) and Multiply-Add (MAD) units were leading to the more efficient implementations of DSP algorithms compared to the conventional ones, which use only primitive resources. Several architectures have been introduced to optimize the performance of the MAC operation in terms of area occupation, critical path delay or power consumption. The

MAC components increase the flexibility of DSP data path synthesis, as a large set of arithmetic operations can be efficiently mapped onto them. Apart from the MAC/MAD operations, many DSP applications are based on Add-Multiply (AM) operations. The simplest design of the AM unit, by first allocating an adder and then driving its output to the input of a multiplier, increases significantly both area and critical path delay of the circuit. Focusing an optimized design of AM operators, fusion techniques are employed based on the direct recoding of the sum of two numbers in its Modified Booth (MB) form. Thus, the carry-propagate (or carry-look-ahead) adder of the conventional AM design is avoided resulting in considerable gains of performance. There is signed-bit MB recoder which transforms redundant binary inputs to their MB recoding form. A special extension of the preprocessing step of the recoder is needed in order to handle operands in carry-save representation. There exists a two-stage recoder which converts a number in carry-save form to its MB representation. The first stage transforms the carry-save form of the input number into signed-digit form which is then recoded in the second stage so that it matches the form that the MB digits request. Recently, we have the technique for the design of high performance flexible coprocessor architectures targeting the computationally intensive DSP applications. .These optimized designs will results in improvements in both area and critical path. An another method is the recoding of a redundant input from its carry-save form to the corresponding borrow-save form keeping the critical path o multiplication operation fixed.

Even though 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, existing recoding schemes are based on complex operations in bit-level, which are implemented by dedicated circuits in gate-level. This paper focuses on the efficient design of FAM operators, focusing the optimization of the recoding scheme for direct shaping of the MB form of the sum of two numbers (Sum to MB – *S-MB*). We evaluated the performance of the projected S-MB technique by scrutiny its three completely different schemes with the existing schemes.

III. EXISTING METHOD

An AM operator means the Add-Multiply operator in which both the addition and multiplication will be performed by a single execution cycle. Many of the DSP applications are based on this AM operation. The straightforward design of the AM unit is by first allocating an adder and then driving its output to the input of a multiplier, increases both the area and critical path delay of the circuit. The fused AM operator is represented by the function Z = X . (A + B). The existing method requires that its input A and B are driven first to an S-MB where the sum bits will be converted to their corresponding Modified Booth form that will makes it easier for the multiplication to take place. This Y input and the third input X are driven to the multiplier in which Modified Booth Multiplication will takes place so as to get Z (as shown in figure a). The drawback of using this particular scheme is that since it uses the normal Carry-Look-Ahead (CLA) adder, in which it needs to generate the carry generate and carry propagate function, even though it is not used directly in the system , which may cause the increased delay in our output. As there are carry signals to be fed in to the adder, the critical path depends on the input bit-width. In order to minimize this delay, we need to minimize the carry propagation stages and hence go for the proposed Square-root Carry Select (SQRT-CSLA) Adder.

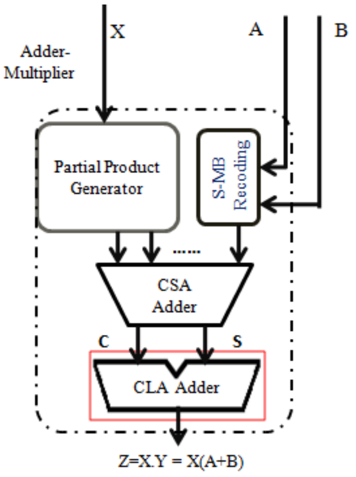

Fig.1. FAM operator based on the S-MB design.

IV. PROPOSED METHOD

A. Overview

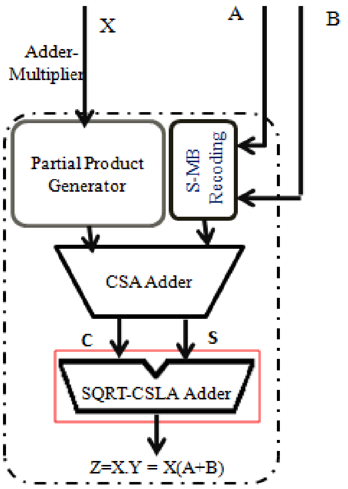

Fig.2.High speed FAM operator design.

This paper focus on the high speed design of AM units which implement the operation $Z = X.(A+B)$. Here in the proposed system it does all the functions except the Carry Select addition. For the existing it uses the normal CLA adder tree for the addition of the partial products generated after the multiplication process, their it requires the generation of the extra signal like the carry-generate and carry-propagate function even though it is not used directly in our system, and this may cause the increase in the delay. And hence in this proposed system instead of the normal adder, this paper proposes a special kind of adder, say Square-root Carry-Look-Ahead Adder (SQRT-CSLA).It is called as the SQRT adder, because for an n-bit input no length it requires √n no of SQRT-CSLA adder stages.

As the name implies this particular adder is based upon a carry select kind of action. As similar to the normal Carry Select Adder, it selects one out of the two Carry signals generated separately for the C_{in} as 0 or 1. The detailed block diagram is shown below.

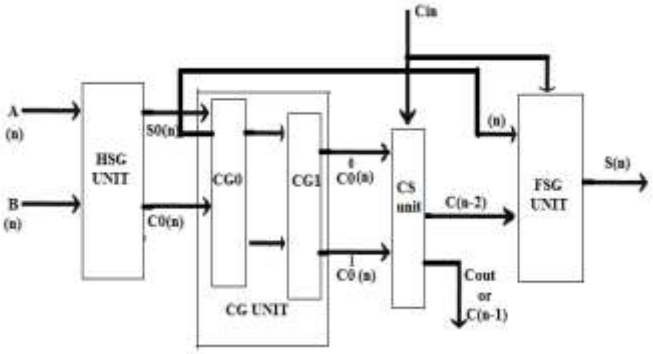

Fig.3. Block Diagram of SQRT-CSLA Adder

It takes 2 inputs say A and B which will be given to the half sum generation unit. It is nothing but the half adder circuit. The generated carry as well as the sum signals will be forwarded to the Carry-Generate unit where it generate separate carry signals for the 0 and 1 signal separately in accordance with the C_{in} signal. This generated carry signals will be forwarded to the Carry-Select unit where it makes a selection among the two different carry signals, in accordance with the same select line say $C_{in..}$ This selected carry signal will be forwarded to the final sum generation unit along with the half sum so as to get the final result. The detailed explanation about each and every component is given below.

B. Half Sum Generation (HSG) Unit

The HSG unit is nothing but a half adder where the sum is obtained by making an XOR operation and the carry signals are obtained from the AND operation between the input signals.

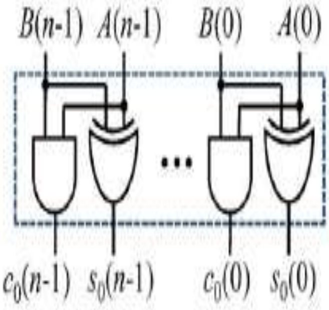

Fig.4. Block Diagram of Half-Sum generation (HSG) unit

C. Carry Generate (CG) unit

For the generation of the carry signal we have our Carry Generate (CG) unit. Based on the nature of the select line ie, C_{in} it will create separate carry signals for the 0 and 1 bit. The block diagram is shown below.

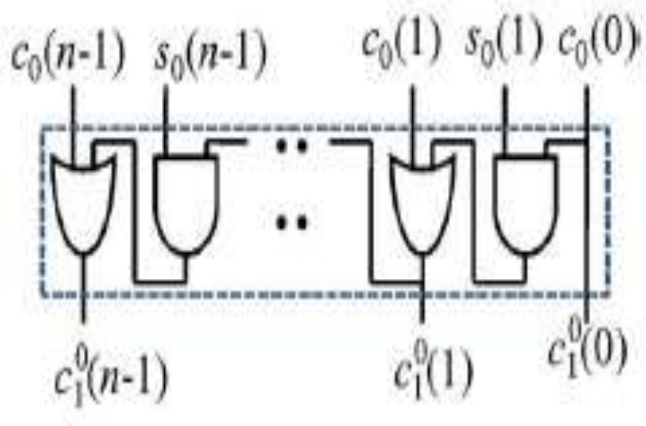

Fig.5. Block Diagram of carry generate function for ‘0’ bit

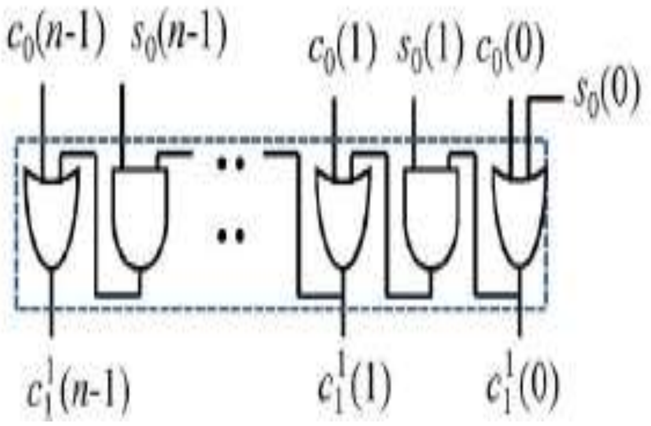

Fig.6. Block Diagram of carry generate function for ‘1’ bit

An extra OR gate is needed for the generation of the carry signal if we are giving ‘1’ as the select input of the CG unit for the addition of that extra ‘1’ with the input signal.

D. Carry Select (CS) Unit

The carry select unit functions similar to a multiplexer, where it makes a selection among two of the inputs say, CG(0) or CG(1) depending on the select line $C_{in.}$ For the ease of implementation this C_{in} is given with the value ‘0’ for the initialization and for the subsequent stages it will takes the C_{out} values of the preceeding stages as the select lines for the next stages of inputs, and the structure is shown below.

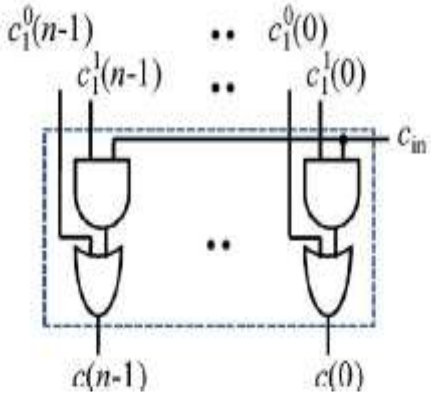

Fig.7. Block Diagram of carry select unit

E. Full Sum Generation (FSG) Unit

It is meant for the generation of the final sum and carry signal by giving the already generated sum signal from the HSG unit and the carry signal got from the CS unit. It will add all the intermediate results obtained so as to get the final sum and carry signal.

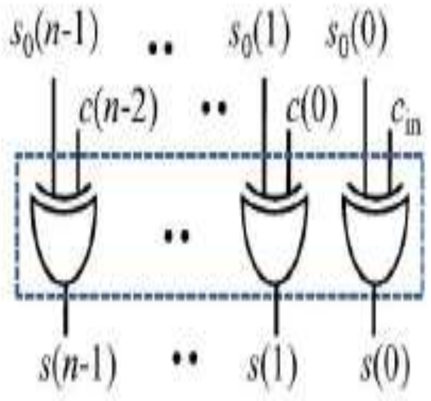
$s_0(n\text{-}1)$ .. $s_0(1)$ $s_0(0)$
$c(n\text{-}2)$ .. $c(0)$ c_{in}
..
$s(n\text{-}1)$ .. $s(1)$ $s(0)$

Fig. 8. Block Diagram of the Full Sum Generation(FSG) Unit

V. RESULTS AND DISCUSSION

A) Simulation Environment

We evaluate the effectiveness of our architecture through architectural simulation. Model sim is used for simulation and Xilinx is used for evaluating the time, area and power for the existing and proposed architecture.

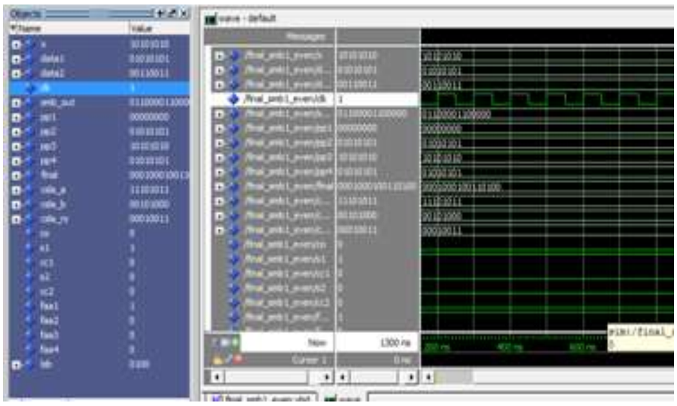

Fig 9: Simulation result of S-MB1 based scheme

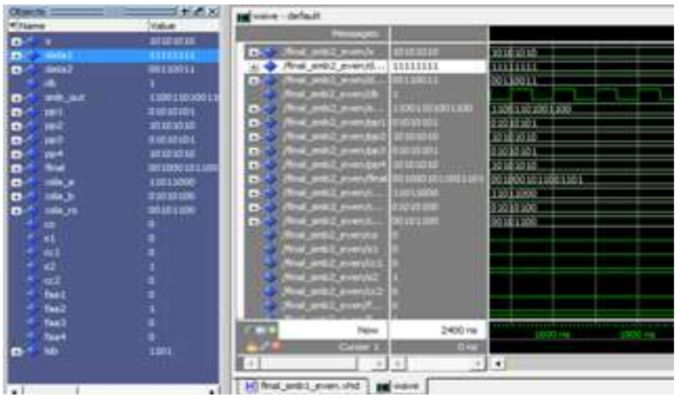

Fig 10: Simulation result of S-MB2 based scheme

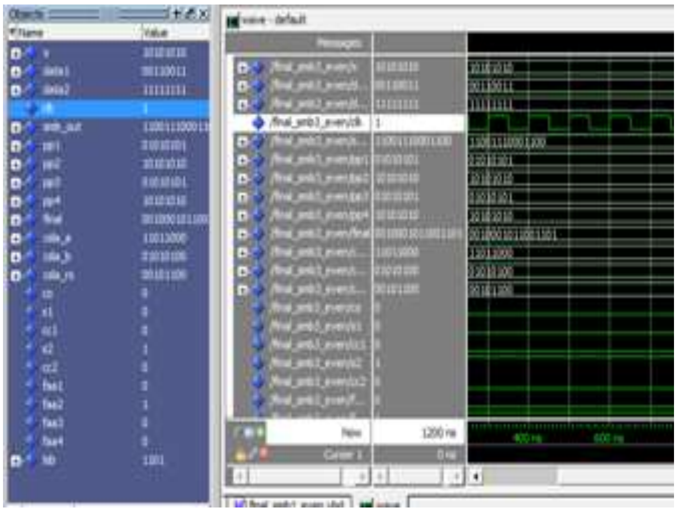

Fig 11: Simulation result of S-MB3 based scheme

VI. PERFORMANCE EVALUATION

A comparison study among the three different schemes in terms of the area complexity and critical delay among the three different schemes are given below.

Table VII
PERFORMANCE COMPARISON OF THE PROPOSED RECODING SCHEMES WITH EXISTING SCHEME

		Power Consumption (mW)	**Time Delay (nS)**	**Area used (total gate count)**
Existing method	**S-MB1**	**139**	**11.014**	**1378**
	S-MB2	**152**	**10.854**	**1354**
	S-MB3	**146**	**10.854**	**1354**
Proposed method	**S-MB1**	**133**	**5.092**	**1374**
	S-MB2	**145**	**5.081**	**1368**
	S-MB3	**143**	**5.081**	**1368**

VI. CONCLUSION

For optimizing the design of the fused-add-multiply three new different schemes are proposed, for the direct recoding of the sum of two numbers in its MB form. When compared to the existing recoding schemes, the proposed schemes deliver considerable improvements in both area, delay and power consumption because of the presence of the SQRT-CSLA Adder used in it. Regarding the over all performance of all the three schemes, S-MB2 and S-MB3 least delay and it uses the least amount of the gate count or area used. The S-MB1 based scheme will suits for the low power applications where the power consumption is very less.

ACKNOWLEDGEMENT

I would like to thank my guide and all the reviewers for their constructive comments and active support to my work.

REFERENCES

[1] Kostas Tsoumanis, Sotiris Xydis, Constantinos Efstathiou, Nikos Moschopoulos, and Kiamal Pekmestzi, “An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator” ieee transactions on circuits and systems—i: regular papers, vol. 61, no. 4, april 2014.

[2] Basant Kumar Mohanty and Sujit Kumar Patel, “Area-Delay-Power efficient Carry-Select adder”, IEEE Trans. On circuits and Systems, vol. 61, no. 6,jan. 2014, pp. 418-422.

[3]Shen-Fu Hsiao, Jun-Hong Zhang Jian, and Ming-Chih Chen, “Low-cost FIR filter designs based on faithfully rounded truncated multiple constant multiplication/Accumulation” IEEE transactions on circuits and systems—ii: express briefs, vol. 60, no. 5, may 2013.

[4] 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.

[5] Y.H.Seo and D.W.Kim, “ A New VLSI architecture of parallel Multiplier based on Radix-2 modified booth algorithm,” IEEE Trans. Very Large Scale Inter. (VLSI) Syst., vol.18, no.2, pp.201-208, Feb.2010.

[6] S.Nikolaidis, E.Karaolis, and E.D.Kyriakis-Bitzaros, “Estimation of signal transition activity in FIR filters implemented by a MAC architecture,” IEEE Trans. Comput.- Aided Des. Integer. Circuits Syst., vol.19, no.1, pp.164-169, jan.2000.

[7] Sukhmeet Kaur1, Suman2 and Manpreet Signh Manna3, “Implementation of Modified Booth Algorithm (Radix 4) and its Comparison with Booth Algorithm (Radix-2)”, Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 3, Number 6 (2013), pp. 683-690.

[8] A.Amaricai, M.Vladutiu, and O.Boncalo, “Design issues and implementationsfor floating-point divide-add fused,” IEEE Trans. Circuits Syst. II–Exp. Briefs, vol. 57, no. 4, pp. 295–299, Apr. 2010.

[9] 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.

[10] J. J. F. Cavanagh,Digital Computer Arithmetic. NewYork: Mc Graw- Hill, 1984.