ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 6, Issue 1, January 2017

Design of Add-Multiply operator usingModified Booth Recoder

K. Venkata Prasad, Dr.M.N. Giri Prasad

Abstract: ***Digital Signal Processing (DSP) applications carry out a large number of complex arithmetic operations. In this, we concentrate on improving the architecture of the fused add multiply (FAM) operator in an efficient way to increase its performance.We explore the methods to implement direct recoding the addition of two numbers in modified booth form. We propose a technique of efficient recoding. For this, we analyze three different techniques by associating them in FAM designs. Comparing with FAM designs i.e., existing recoding schemes, the proposed method gives substantial reductions in specifications of delay, complexity and utilization of power in FAM unit.***

Index Terms: Add-Multiply operation, FAM design, Modified Booth recoding, VLSI design.

I. Introduction

Digital Signal Processing (DSP) applications accomplish arithmetic operations based onkernels,those are like Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT). The pursuance of these DSP designs intrinsically affects the design and the allocation in arithmetic units.

Based on recent activities, multiplication is followed by an addition, the Multiply-Add (MAD) and Multiply-Accumulator (MAC) units which give [1] the efficient application of DSP methods.Many architectures are evolved to enhance the MAC operation in word of delay, area and power [1]-[3]. Apart from the MAC/MAD units, Add-Multiply unit is mostly used in DSP implementations (e.g. FFT [3]). The common AM unit consists of adder, whose output is connected to input of multiplier, which gives increase of area and delay of circuit. To enhance the AM operators we use fused techniques that are based on direct recoding of sum of two numbers in Modified Booth (MB) form. The carry propagate adder of AM design, which is removed gives the gain in performance.In [10], author proposes two-stage recoder, where itconvert a number in carry-save form to MB representation. First stage the carry-save form of input into signed-digit form and it recoded in second stage. Recently, the procedure of [10] is used for design of high performance coprocessor architectures directing theDSP applications [14]. In [11], itenhanced design of [8] which givesthe improvements in area and critical path. In [17], the authorpresents the recoding of a redundant input from carry-save form to the similar borrow-save by keeping the criticalpath of multiplication operation fixed.

The proposed technique gives the efficient implementation of FAM operators. Here we implement a recoding method, it decreases area, power utilization and path delay. The proposed S-MB algorithm is simple and easy to modify, which is applied either in signed or unsigned numbers which consists of either even or odd number of bits. We analyze three different methods of proposed S-MB scheme by using conventional and signed full adders (FAs) and half adders (HAs).

The proposed recoding scheme gives enhanced solutions for FAM design by making the targeted operator to be timing functional (no timing violations) for a larger range of frequencies. Also, under the same timing constraints, the proposed designs deliver developments in area occupation and power consumption, thus outperforming the existing *S-MB* recoding solutions.

II. Motivation and fused AM Implementation

A. Motivation

In this, we concentrate on AM designs which implements the operation Z=X.(A+B). Conventional designof AM operator (Fig. 1(a)) of inputs A, B and X. The addition of A and B gives the result, this output will be given to input of X which gives the resultant value Z. The major drawback is adder circuit, which increases critical path delay in AM unit. Inside the adder the carry signals will be propagated. This path delay depends on number of bits in input. To overcome this drawback in path delays we use Carry-Look-Ahead(CLA) adder, however which increases area and power consumption. To enhance AM operator, it

combinesthe both adder and MB encoder in single block shown in Fig. 1(b), which is done by recoding of Y=A+B directly in to MB form. Fused Add-Multiply(FAM) is having one adder in the parallel multiplication process which is present at the end of the process. It results in decreaseof area and path delay.

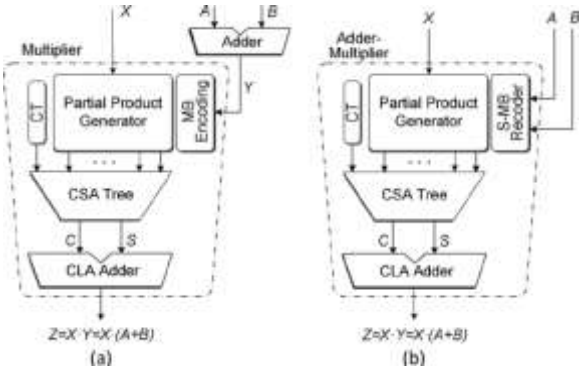

Fig.1. AM unit based on (a) conventional design and (b) fused design

B. Review of the Modified Booth Form

In MB encoding we use Signed-digit radix-4 encoding technique. Advantage of using this, it can reduce the number of partial products by half in multiplication by comparing with the radix-2 encoding technique.

Let us take product of two numbers in 2's complement form X and Y of having n=2k bits.

The multiplicand Y is written in MB form as:

$$Y = < y_{n-1}y_{n-2} \ldots y_1y_2 >_{2's} = -y_{2k-1} \cdot 2^{2k-1} + \sum_{i=0}^{2k-2} y_i \cdot 2^i$$

$$= < Y_{k-1}^{MB} Y_{k-2}^{MB} \ldots Y_1^{MB} Y_0^{MB} >_{MB} = \sum_{j=0}^{k-1} Y_j^{MB} \cdot 2^{2j}$$

$$Y_j^{MB} = -2y_{2j+1} + y_{2j} + y_{2j+1}$$

$$y_j^{MB} \in \{-2, -1, 0, +1, +2\}, \qquad 0 \le j \le k-1$$

Consider $y_{-1} = 0$

Table 1 represents the summarization of MB encoding scheme. In this the outputs are s, one and two. The signal s shows the sign bit it shows positive (s=0) or negative (s=1). Signals one and two represents the absolute value if the value is one then signal one is 1, if the value is 2 then signal two is 1 otherwise the 0. By using these signals we can calculate MB bits by using this relation:

$$Y_j^{MB} = (-1)^{s_j} \cdot [one_j + 2 \cdot two_j]$$

Fig. 2(a) shows the Boolean equations on which the operation of MB encoding signals is based (Fig. 2(b)).

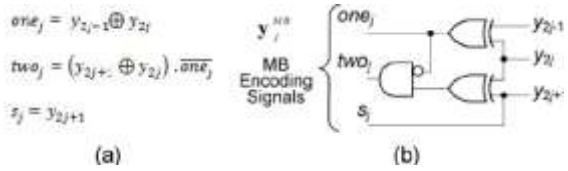

Fig. 2 (a) Boolean equations and (b) gate-level schematic of the MB encoding signals.

TABLE I

MODIFIED BOOTH ENCODING TABLE

Binary			y_j^{MB}	MB encoding			Input carry
y_{2j+1}	y_{2j}	y_{2j-1}		Sign=s_j	×1=one_j	×2=two_j	$c_{in,j}$
0	0	0	**0**	0	0	0	0
0	0	1	**+1**	0	1	0	0
0	1	0	**+1**	0	1	0	0
0	1	1	**+2**	0	0	1	0
1	0	0	**-2**	1	0	1	1
1	0	1	**-1**	1	1	0	1
1	1	0	**-1**	1	1	0	1
1	1	1	**0**	1	0	0	0

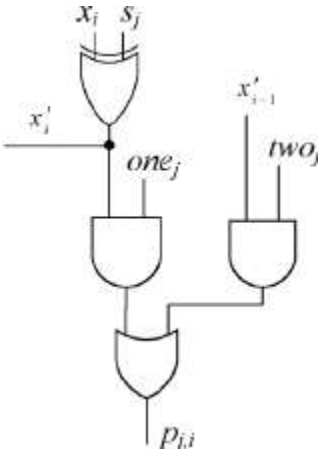

Fig.3. Generation of i-thbit $p_{j,i}$ of the partial product PP_j

C. FAM Implementation

Let us take the product of X and Y. The term Y = <$y_{n-1}\ y_{n-2}\ \cdot\ \cdot\ \cdot\ y_1\ y_2$>is recoded based on MB method and multiplied with X = < $x_{n-1}\ x_{n-2}\ \cdot\ \cdot\ \cdot\ x_1\ x_2$>. X and Y has n = 2k bits which are in 2's complement form.The generation of the k partial products is followed as:

$$PP_j = X\,.\,Y_j^{MB} = \overline{p_{j,n}}\,.2^n + \sum_{i=0}^{n-1} p_{j,i}\,.2^i$$

Fig. 3 shows the implementation for generation of the i-th bit $p_{j,I}$ of the partial product PP_j followed as:

$$p_{j,i} = ((x_i \oplus s_j) \wedge one_j) \vee ((x_{i-1} \oplus s_j) \wedge two_j)$$

Let us assume $x_{-1} = 0$ and $x_n = x_{n-1}$

After getting the partial products, they are added, properly weighted, through a Carry-Save Adder (CSA) with the Correction Term (CT) which is followed as:

$$Z = X\,.\,Y = CT + \sum_{j=0}^{k-1} PP_j\,.2^{2j}$$

$$CT = CT(low) + CT(high)$$

$$= \sum_{j=0}^{k-1} c_{in,j}\,.2^{2j} + 2^n\left(1 + \sum_{j=0}^{k-1} 2^{2j+1}\right)$$

$$c_{in,j} = \left(one_j \vee two_j\right) \wedge s_j$$

Finally, the carry-save output of the Wallace CSA tree is leaded to a fast Carry Look Ahead (CLA) adder to form the final result as shown in Fig. 1(b).

III. SUM TO MODIFIED BOOTH RECODING TECHNIQUE (S-MB)

A. *Defining Signed-Bit Full Adders and Half Adders for Structured Signed Arithmetic*

In S-MB recoding scheme, we recode sum of two consecutive bits of input A(a_{2j}, a_{2j+1}) with two consecutive bits of input B(b_{2j}, b_{2j+1}) to one MB digit Y_j^{MB}. To calculate this MB digit we can develop a signed Full Adders (FA) and Half Adders (HA), for this both input and output are in 2's complement form.

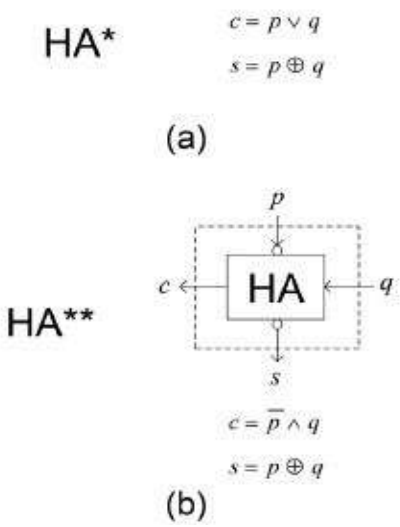

Fig. 4. Schematics for signed (a) HA* and (b) HA**

Here, half adders in signed form consisting of two types i.e., HA* and HA**are used. The truth tables for this HAs presents in Table II – IV and shown in Fig. 4(a). HA* of the output values are {0, +1, +2}. It is shown in Table III. HA** is shown in Fig. 4(b). HA** of the output values are {0, -1, -2}. It is shown in Table IV.

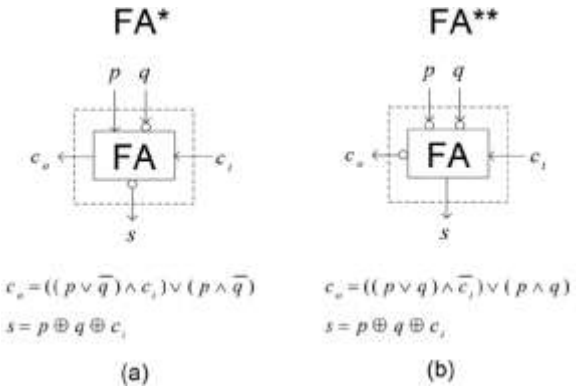

Fig. 5 Boolean equations and schematics for signed (a) FA* and (b) FA**

TABLE II
HA* BASIC OPERATION.

Inputs		Output Value[1]	outputs	
p(+)	q(+)		c(+)	s(-)
0	0	0	0	0
0	1	+1	1	1
1	0	+1	1	1
1	1	+2	1	0

TABLE III
HA* DUAL OPERATION.

Inputs		Output Value[2]	outputs	
p(-)	q(-)		c(-)	s(+)
0	0	0	0	0
0	1	-1	1	1
1	0	-1	1	1
1	1	-2	1	0

TABLE IV
HA** OPERATION

Inputs		Output Value[3]	outputs	
p(-)	q(+)		c(+)	s(-)
0	0	0	0	0
0	1	+1	1	1
1	0	-1	0	1
1	1	0	0	0

TABLE V
FA* OPERATION

Inputs			Output Value[1]	Outputs	
p(+)	q(-)	c_i(+)		c_0(+)	s(-)
0	0	0	0	0	0
0	0	1	+1	1	1
0	1	0	-1	0	1
0	1	1	0	0	0
1	0	0	+1	1	1
1	0	1	+2	1	0
1	1	0	0	0	0
1	1	1	+1	1	1

Signed full adders consisting of two different types is shown in Table V and VI and in Fig. 5. The schematics for both FA* and FA** are presented in Fig. 5(a) and (b) respectively. FA* of output values are {-1, 0, +1, +2} from the FA* output equation. FA** of output values are {-2, -1, 0, +1} from the FA** output equation.

TABLE VI
FA** OPERATION

Inputs			Output Value[2]	Outputs	
p(-)	q(-)	c_i(+)		c_0(-)	s(+)
0	0	0	0	0	0
0	0	1	+1	0	1
0	1	0	-1	1	1
0	1	1	0	0	0
1	0	0	-1	1	1
1	0	1	0	0	0
1	1	0	-2	1	0
1	1	1	-1	1	1

B. *Proposed S-MB Recoding Techniques*

In S-MB recoding scheme use both the conventional and signed FAs and HAs to design three new schemes. In these three schemes inputs A and B are in 2's complement form and bit length is either even (2k bits) or odd (2k+1 bits). Targeting to transform the sum of A and B (Y = A+B)in MB form.

1)*S-MB1 Recoding scheme:* The scheme of S-MB1for the both even and odd number of bit width can be shown in Fig. 6 (for even Fig. 6(a) and odd Fig. 6(b)). Consider $b_{-1}=0$ and $c_0=0$.

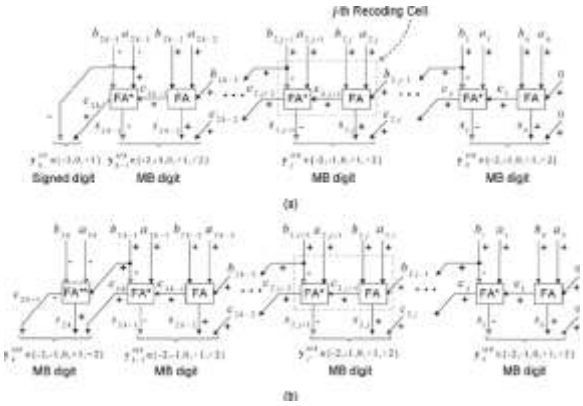

Fig. 6 S-MB1 recoding scheme for (a) even and (b) odd number of bits

$$Y = A + B = y_k \cdot 2^{2k} + \sum_{j=0}^{k-1} y_j^{MB} \cdot 2^{2j}$$

Where $Y_j^{MB} = -2s_{2j+1} + s_{2j} + c_{2j}$

$$c_{2j+1} = (a_{2j} \wedge b_{2j}) \vee \left(b_{2j-1} \wedge (a_{2j} \vee b_{2j})\right)$$

$$s_{2j} = a_{2j} \oplus b_{2j} \oplus b_{2j-1}$$

$$c_{2j+2} = (a_{2j+1} \wedge \bar{b}_{2j+1}) \vee \left(c_{2j+1} \wedge (a_{2j+1} \vee \bar{b}_{2j+1})\right)$$

$$s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}$$

The MSB ofS-MB1 is of two cases in first case, the bit width is even (Fig. 6(a)), for this the MSB is a signed digit and is followed as:

$$y_{k,even}^{SD} = -a_{2k-1} + c_{2k}$$

In second case, the bit width is odd (Fig. 6(b)), the MSB is formed based on $Y_{k,odd}^{SD}$ outputs of FA**. Path

delay of S-MB1 recoding is constant with respect to input bit width and is followed as:

$$T_{S-MB1} = T_{FA,carry} + T_{FA^*,sum}$$

2) *S-MB2 Recoding Scheme:* The second technique of the proposed, which is S-MB2, which is shown in Fig. 7 for both even (Fig. 7(a)) and odd (Fig. 7(b)) bit width in its input.Consider $c_{0,1} = 0$ and $c_{0,2} = 0$. The values s_{2j+1}, s_{2j}, and $c_{2j,2}$ are calculated according to S-MB1 recoding scheme.

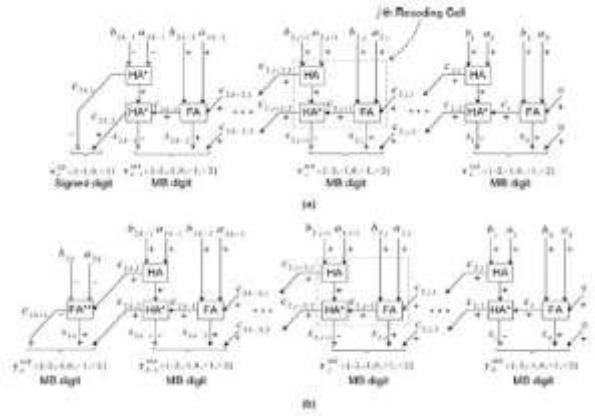

Fig. 7 S-MB2 recoding scheme for (a) even and (b) odd number of bits

$$c_{2j+2,2} = c_{2j+1} \vee (a_{2j+1} \oplus b_{2j+1})$$

$$s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}$$

The MSB of the S-MB2 is of two cases in first case, the bit width is even (Fig. 7(a)), for this the MSB is a signed digit and is followed as:

$$y^{SD}_{k,even} = -c_{2k,1} + c_{2k,2}$$

In second case, the bit width is odd (Fig. 6(b)), the MSB $Y^{SD}_{k,odd}$ is formed based on outputs of FA**.Thepath delay of S-MB2 recoding is constant with respect to the input bit width and is followed as:

$$T_{S-MB2} = T_{HA,carry} + T_{FA,carry} + T_{HA^*,sum}$$

3) *S-MB3 Recoding Scheme:* The third technique of the proposed, which is S-MB3, which is shown in Fig. 8 for both even (Fig. 8(a)) and odd (Fig. 8(b)) bit width in its input. Consider $c_{0,1} = 0$ and $c_{0,2} = 0$. We can use conventional FA for to produce carry and sum. $C_{2j,1}$ is calculated from the HA* operation and s_{2j+1} is produced from HA**. The remaining can be calculated as follows:

$$c_{2j+2,2} = c_{2j+1} \wedge \overline{(a_{2j+1} \oplus b_{2j+1})}$$

$$s_{2j+1} = a_{2j+1} \oplus b_{2j+1} \oplus c_{2j+1}$$

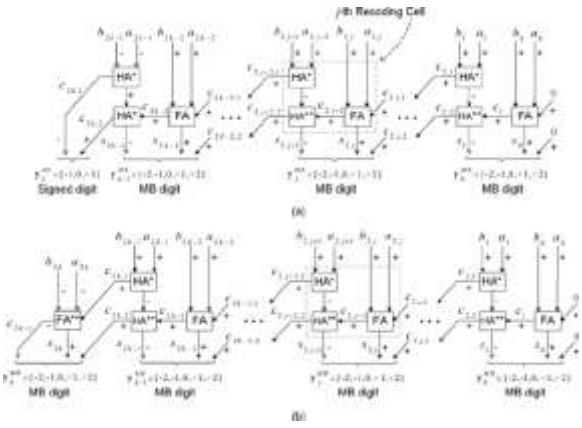

Fig. 8 S-MB3 recoding scheme for (a) even and (b) odd number of bits

The MSB bit of the S-MB3 is of two cases. Firstcase is input bit width is even, which is implemented by the HA* and HA** recoding cell, other case is the input bit width is odd. The MSD for even and odd bit width cases of A and B, are formed as S-MB2 scheme. The path delay of S-MB3recoding scheme is followed by:

$$T_{S-MB3} = T_{HA^*,carry} + T_{FA,carry} + T_{HA^{**},sum}$$

4) *Unsigned Input Numbers*: In this scheme if the input is in unsigned form, the most significant bits are positively signed. The Fig 9 – 11 shows for all the S-MB recoding schemes to both even and odd number of bit width of inputs.

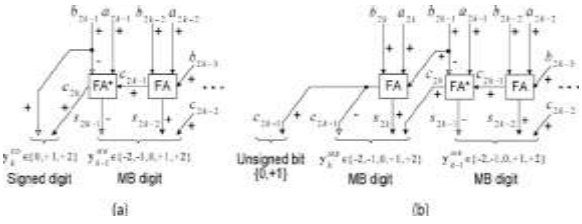

Fig. 9. Implementation of MSD of*S-MB*1 recoding scheme of unsigned input numbers for (a) even and (b) odd bit-width.

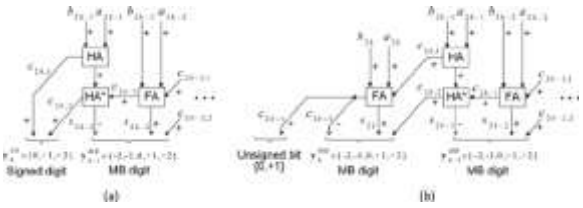

Fig. 10. Implementation of MSD of *S-MB*2 recoding scheme of unsigned input numbers for (a) even and (b) odd bit-width.

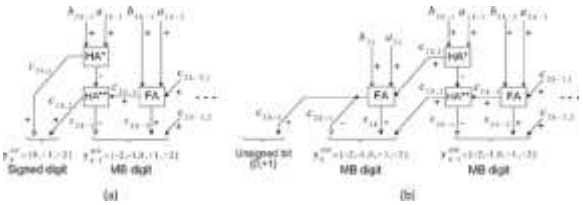

Fig. 11. Implementation of MSD of *S-MB*3 recoding scheme of unsigned input numbers for (a) even and (b) odd bit-width.

IV. PEFORMANCE EVALUATION

A. *Theoretical Analysis:*

Here, we will do theoretical analysis comparison with area, path delay of three recoding schemes and existing schemes [10], [11] and [17]. Our work will depends on gate model. Area and delay of various components in the gate model is shown in Table VII. More specifically, for comparisons the 2-input primitive gates (NAND, AND, NOR, OR) count as one gate equivalent for area and delay, where the 2-input XOR, XNOR gates count as two gate equivalents [11]. The area of a FA and a HA is 7 and 3 gate equivalents respectively. The delays of sum and carry outputs of a FA are 4 and 3 gate equivalents respectively, while those of a HA are 2 and 1.In Fig 12-14 represents the [10], [11] and [17] recoding schemes. Table VIII represents area and path delay of proposed scheme and existing methods in [10], [11] and [17].

TABLE VII
AREA AND DELAY OF VARIOUS COMPONENTS IN THE GATE MODEL.

Components	Area (gate equivalent)	Delay (gate equivalent)
NAND-2, NOR-2	A_g	T_g
NAND-3, NOR-3	$2A_g$	$2T_g$
XOR, XNOR	$2A_g$	$2T_g$
HA	$3A_g$	$T_{HA,\ carry}=T_g$ $T_{HA,\ sum}=2T_g$
FA	$7A_g$	$T_{FA,\ carry}=3T_g$ $T_{FA,\ sum}=4T_g$

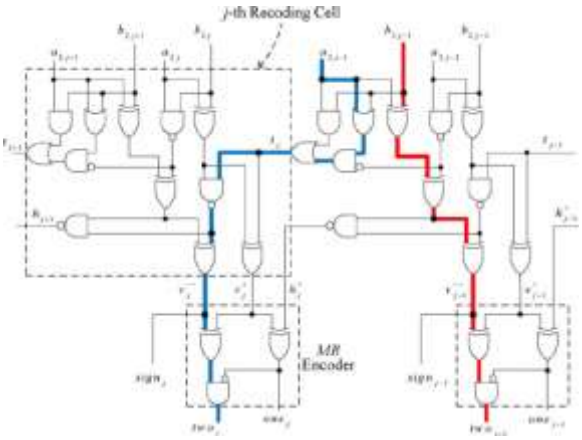

Fig. 12. Recoding scheme of [10] at level of recoding cell and critical paths

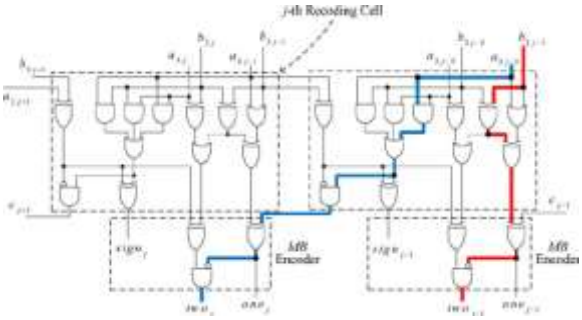

Fig. 13. Recoding scheme of [11] at level of recoding cell and critical paths.

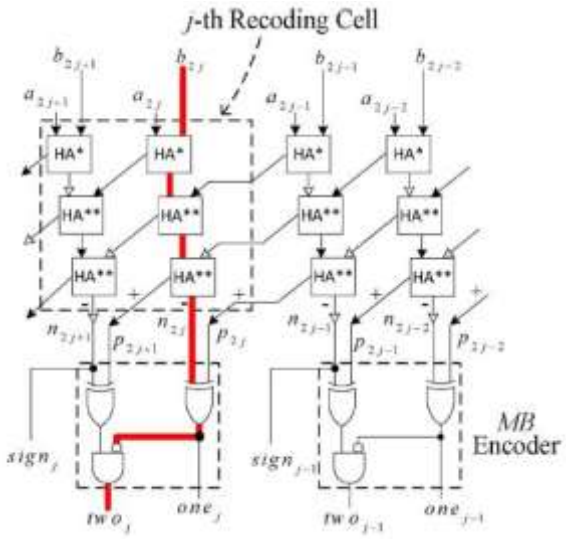

Fig. 14. Recoding method of [17] basic recoding cell and critical path.

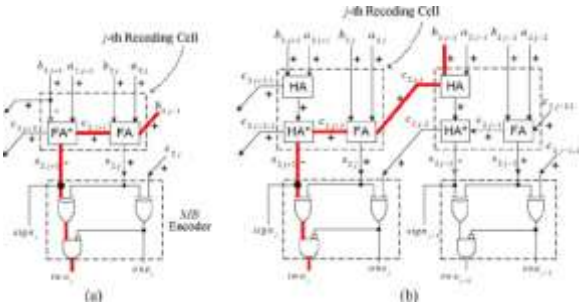

Fig. 15. Critical path delay of proposed (a) *S-MB*1 and (b) *S-MB*2 recoding scheme.

TABLE VIII

AREA AND DELAY COMPARISON OF PROPOSED RECODING SCHEMES WITH EXISTING RECODERS

Design	Area complexity	Critical delay
[12]	$7A_{xor}+8A_g=22A_g$	$3T_{xor}+3T_g=9T_g$ or $2T_{xor}+5T_g=9T_g$
[13]	$7A_{xor}+9A_g=23A_g$	$T_{xor}+5T_g=7T_g$ or $3T_{xor}+3T_g=7T_g$
[23]	$64_{HA}+2A_{xor}+A_g=23A_g$	$T_{HA,sum}+T_{HA,sum}+T_{HA,sum}+T_{xor}+T_g=9T_g$
S-MB1	$2A_{FA}+2A_{xor}+A_g=19A_g$	$T_{FA,carry}+2T_g+2T_{xor}+T_g=8T_g$
S-MB2	$A_{FA}+2A_{HA}+2A_{xor}+A_g=18A_g$	$T_{HA,carry}+2T_g+T_{HA,sum}+T_{xor}+T_g=8T_g$
S-MB3	$A_{FA}+2A_{HA}+2A_{xor}+A_g=18A_g$	$T_{HA,carry}+2T_g+T_{HA,sum}+T_{xor}+T_g=8T_g$

Table VIII represents the proposed methods attains area reduction of up to $5A_g$ compared with existing methods [10], [11] and [17]. The proposed methods are faster by T_g of path delay than [10] and [17] and very close to path delay of [11].

B.*Experimental Analysis*

In this section, we compare performance of three proposed recoding methods explored in Section III with the three schemes described in [10], [11], [17].We included each of recoding schemes in a fused Add-Multiply (FAM) operator (Fig. 1(b)) and implemented them using structural Verilog HDL foreven and odd number of bit. The tool used in this paper is Xilinx13.2 for the calculation of area, delay and power.

The performance of FAM designs that include proposed methods is considered with respect to bit widths. Tables IX and X showsarea and power of FAM units for even (i.e., 8 bits) and odd (i.e., 9 bits) bit-width.

As shown in Table IX, there are timing delays and power consumption among the three recoders. The recoder of [11] is faster by $2T_g$thanrecoders of [10] and [17] which is verified by the practical results where the [11] based FAM design gives in low critical delays compared torecoders of [10] and [17]. The recoding of [17] includes subcomponents whilerecoders of [10] and [11] are designed directly in gate-level. The ungrouping of the hierarchy ofrecoder of [17] gives to larger timing optimizations than the ones of [10] and [11]. Thus, in Tables IX and X,the FAM designs achieves lower critical delays than the FAM design which uses that of [10].

TABLE IX
CRITCICAL PATH DELAY AND POWER CONSUMPTION FOR THE PROPOSED SYSTEM

Recoding schemes	Even bit width		Odd bit width	
	Delay (ns)	Power consumption (W)	Delay (ns)	Power consumption (W)
S-MB1	17.343	0.034	18.227	0.034
S-MB2	17.943	0.038	18.320	0.038
S-MB3	17.208	0.035	17.378	0.035

TABLE X
AREA FOR THE PROPOSED SYSTEM

Recoding schemes	Area for even bit width			Area for odd bit width		
	LUTs	Slices	Flip flops	LUTs	Slices	Flip flops
S-MB1	175	101	7	205	110	10
S-MB2	165	96	8	197	110	10
S-MB3	170	105	14	209	124	18

V. CONCLUSION

This paper focuses on enhanced design of design of the Fused-Add Multiply (FAM) operator.We presents a structured technique for direct recoding of sum of two numbers to MB form. We analyze three different designs of proposed *S-MB* recoder and compared to existing ones [10], [11] and [17]. The proposed recoding methods, when implemented in FAMdesigns, gives improvement in performance when compared to the efficient recoding schemes.

REFERENCES

[1] 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.

[2] 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. Integr. Circuits Syst.*, vol. 19, no. 1, pp. 164–169, Jan. 2000.

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

[4] 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.

[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] A. Peymandoust and G. de Micheli, "Using symbolic algebra in algorithmic level DSP synthesis," in *Proc. Design Automation Conf.*, Las Vegas, NV, 2001, pp. 277–282.

[7] 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.

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

[9] J. D. Bruguera and T. Lang, "Implementation of the FFT butterfly with redundant arithmetic," *IEEE Trans. Circuits Syst. Il, Analog Digit. Signal Process.*, vol. 43, no. 10, pp. 717–723, Oct. 1996.

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

[11] 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.

[12] O. L. Macsorley, "High-speed arithmetic in binary computers," *Proc. IRE*, vol. 49, no. 1, pp. 67–91, Jan. 1961.

[13] N. H. E. Weste and D. M. Harris, "Datapath subsystems," in *CMOS VLSI Design: A Circuits and Systems Perspective*, 4th ed. Readington: Addison-Wesley, 2010, ch. 11.

[14] S. Xydis, I. Triantafyllou, G. Economakos, and K. Pekmestzi, "Flexible datapath synthesis through arithmetically optimized operation chaining," in *Proc. NASA/ESA Conf. Adaptive Hardware Syst.*, 2009,pp. 407–414.

[15] Z. Huang, "High-Level Optimization Techniques for Low-Power Multiplier Design," Ph.D., University of California, Department of Computer Science, Los Angeles, CA, 2003. [16] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron. Comput.*, vol. EC-13, no. 1, pp. 14–17, 1964.

[17] M. Daumas 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.

Mr. K. Venkata Prasad Received B Tech Degree in (ECE) From SKUCET, Anantapuramu in 2014. Currently he is doing M.Tech in JNTU College of Engineering Anantapur. His areas of Interest are Digital systems and computerelectronics.

Dr. M.N. Giri Prasad completed M.Tech and Ph.D. Working as Professor & Head, ECE Department JNTU College of Engineering, Anantapur. Fields of Interest are IMAGE PROCESSING AND BIO- MEDICAL SIGNAL PROCESSING