IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 12, Issue 5, Ver. II (Sep.- Oct. 2017), PP 20-30
www.iosrjournals.org

Performance Improved Multipliers Based On Non Redundant Radix-4 Signed Digit Encoding

S.Sri Divya[1], K.s.n raju[2]
[1]*P.G Student VLSI Design (ECE), SVECW*
[2]*Assistant Professor, Dept of ECE, SVECW*

Abstract: *In this paper, we introduce architectures of pre-encoded multipliers based on non redundant radix-4 signed digit encoding technique. In this architecture of nr4sd and modified booth encoding is used. Using Modified booth is a redundant radix-4 encoding technique. To this extend radix-8 modified booth encoding which reduce the number of partial product implementation. Extensive experimental analysis verifies that the radix-8 mb encoding occupies less area and delay compared to nr4sd multipliers.*

Key words-*Multiplying circuits, modified booth encoding, pre-encoded multipliers, VLSI implementation.*

I.INTRODUCTION

Multimedia and digital signal processing (DSP) applications (e.g., fast Fourier transform (FFT), audio/video Codec's) carry out a large number of multiplications with coefficients that do not change during the execution of the application. Since the multiplier is a basic component for implementing computationally intensive applications, its architecture seriously affects their performance. Constant coefficients can be encoded to contain the least non-zero digits using the canonic signed digit (CSD) representation [1].
CSD multipliers comprise the fewest non-zero partial products, which in turn decreases their switching activity. However, the CSD encoding involves serious limitations. Folding technique [2], which reduces silicon area by time-multiplexing many operations into single functional units, e.g., adders, multipliers, is not feasible as the CSD-based multipliers are hard-wired to specific coefficients. In [3], a CSD-based programmable multiplier design was proposed for groups of pre-determined coefficients that share certain features. The size of ROM used to store the groups of coefficients is significantly reduced as well as the area and power consumption
of the circuit.

However, this multiplier design lacks flexibility since the partial products generation unit is designed specifically for a group of coefficients and cannot be reused for another group. Also, this method cannot be easily extended to large groups of pre-determined coefficients attaining at the same time high efficiency Modified Booth (MB) encoding [4], [5], [6], [7] tackles' the afore-mentioned limitations and reduces to half the number of partial products resulting to reduced area, critical delay and power consumption. However, a dedicated encoding circuit is required and the partial products generation is more complex. In [8], Kim et al. proposed a technique similar to [3], for designing efficient MB multipliers for groups of pre-determined coefficients with the same limitations described in the previous paragraph In [9], [10], multipliers included in butterfly units of FFT processors use standard coefficients stored in ROMs. In audio [11], [12] and video [13], [14] Codec's, fixed coefficients stored in memory, are used as multiplication inputs. Since the values of constant coefficients are known in advance, we encode the coefficients off-line based on the MB encoding and store the MB encoded coefficients (i.e., 3 bits per digit) into a ROM. Using this technique [15], [16], [17], the encoding circuit of the MB multiplier is omitted. We refer to this design as pre-encoded MB multiplier. Then, we explore a Non-Redundant radix-4 Signed Digit (NR4SD) encoding scheme extending the serial encoding techniques of [6], NR4SD encoding scheme uses one of the following sets of digit value{-1,0,+1,+2} or{-2,-1,0,+1} In order to cover the dynamic range of the 2's complement form, all digits of the NR4SD form.

Using the proposed encoding formula, we pre encode the standard coefficients and store them into a ROM in a condensed form (i.e., 2 bits per digit). Compared to the pre-encoded MB multiplier in which the encoded coefficients need 3 bits per digit, the proposed NR4SD scheme reduces the memory size. Also, compared to the MB form, which uses five digit values {-2,-1, 0, +1,+2}, the proposed NR4SD encoding uses four digit values. Thus, the

NR4SD-based pre-encoded multipliers include a less complex partial products generation circuit. We explore the efficiency of the aforementioned pre-encoded multipliers taking into account the size of the coefficients' ROM.

II .MODIFIED BOOTH ALGORITHM

In order to achieve high-speed multiplication, multiplication algorithms using parallel counters, such as the modified Booth algorithm has been proposed, and some multipliers based on the algorithms have been implemented for practical use. This type of multiplier operates much faster than an array multiplier for longer operands because its computation time is proportional to the logarithm of the word length of operands.

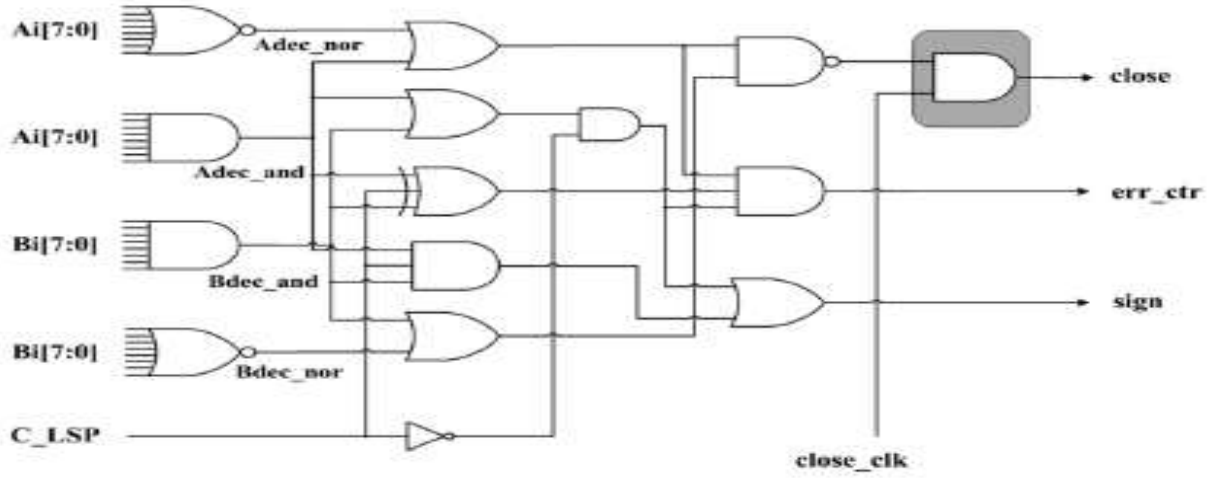

Fig1. Modified booth encoder

Booth multiplication is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied. It is possible to reduce the number of partial products by half, by using the technique of radix-4 Booth recoding. The basic idea is that, 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. The advantage of this method is the halving of the number of partial products. To Booth recode the multiplier term, we consider the bits in blocks of three, such that each block overlaps the previous block by one bit. Grouping starts from the LSB, and the first block only uses two bits of the multiplier. Figure 3 shows the grouping of bits from the multiplier term for use in modified booth encoding.

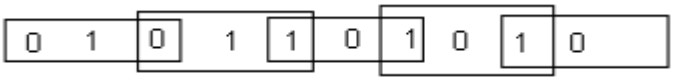

fig2 .Grouping of bits from the multiplier term

For the partial product generation, we adopt Radix-4 Modified Booth algorithm to reduce the number of partial products for roughly one half. For multiplication of 2's complement numbers, the two-bit encoding using this algorithm scans a triplet of bits. When the multiplier B is divided into groups of two bits, the algorithm is applied to this group of divided bits.

III . NON-REDUNDANT RADIX-4 SIGNED DIGIT ALGORITHM

We present the Non-Redundant radix-4 Signed-Digit (NR4SD) encoding technique. As in MB form, the number of partial products is reduced to half. When encoding the 2's complement number B, digits bNRj take one of four values: f2; 1; 0; +1g or bNR+j2 f 1; 0; +1; +2g at the NR4SD- or NR4SD+ algorithm, respectively. Only four different values are used and not five as in MB algorithm, which leads to 0 j k 2. As we need to cover the dynamic range of the 2's complement form, the most significant digit is MB encoded (i.e., bMBk 12 f 2; 1; 0; +1; +2g).The NR4SD- and NR4SD+encoding algorithms are illustrated in detail in Fig. 3 and 4, respectively.

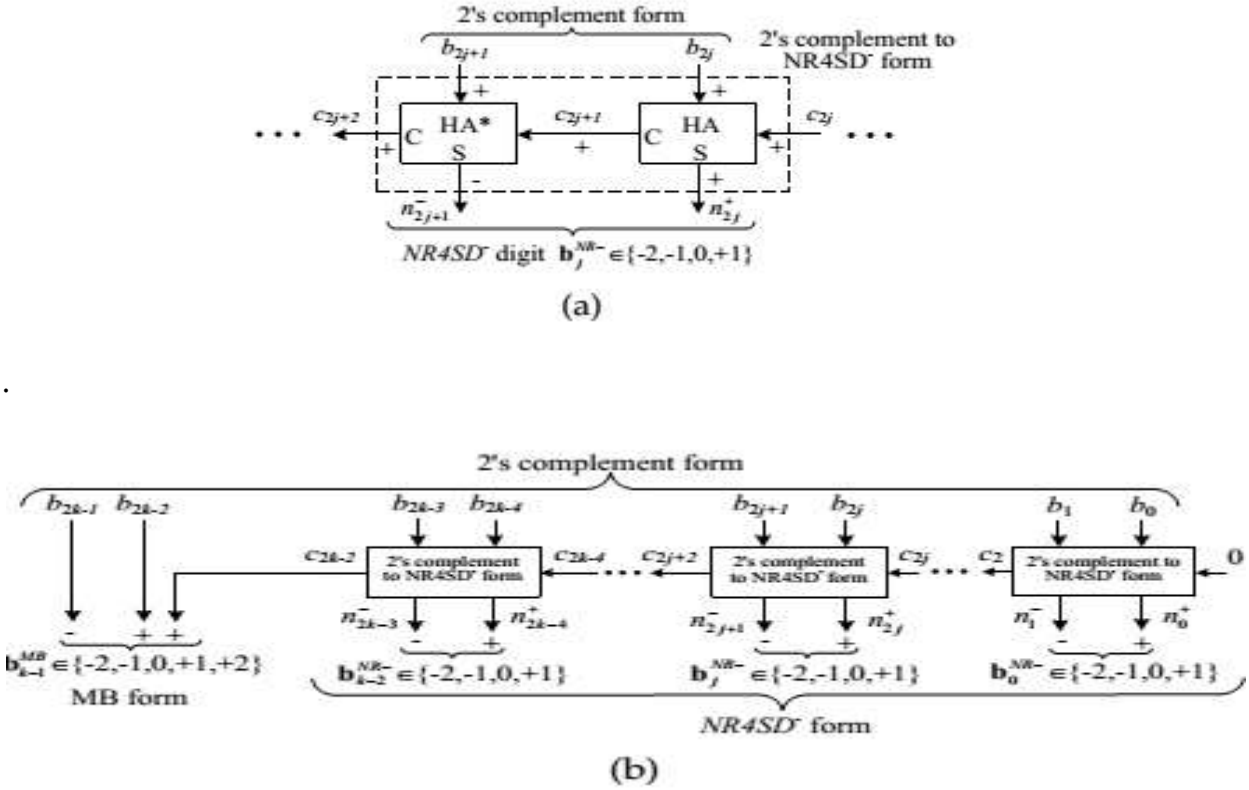

Fig. 3 Block Diagram of the NR4SD Encoding Scheme at the (a) Digit and (b) Word Level.

Table-1

NR4SD⁻ Encoding

2's complement			NR4SD⁻ *form*			*Digit*	NR4SD⁻ *Encoding*		
b_{2j+1}	b_{2j}	c_{2j}	c_{2j+2}	n^-_{2j+1}	n^+_{2j}	$\mathbf{b}_j^{NR-}$	one_j^+	one_j^-	two_j^-
0	0	0	0	0	0	**0**	0	0	0
0	0	1	0	0	1	**+1**	1	0	0
0	1	0	0	0	1	**+1**	1	0	0
0	1	1	1	1	0	**-2**	0	0	1
1	0	0	1	1	0	**-2**	0	0	1
1	0	1	1	1	1	**-1**	0	1	0
1	1	0	1	1	1	**-1**	0	1	0
1	1	1	1	0	0	**0**	0	0	0

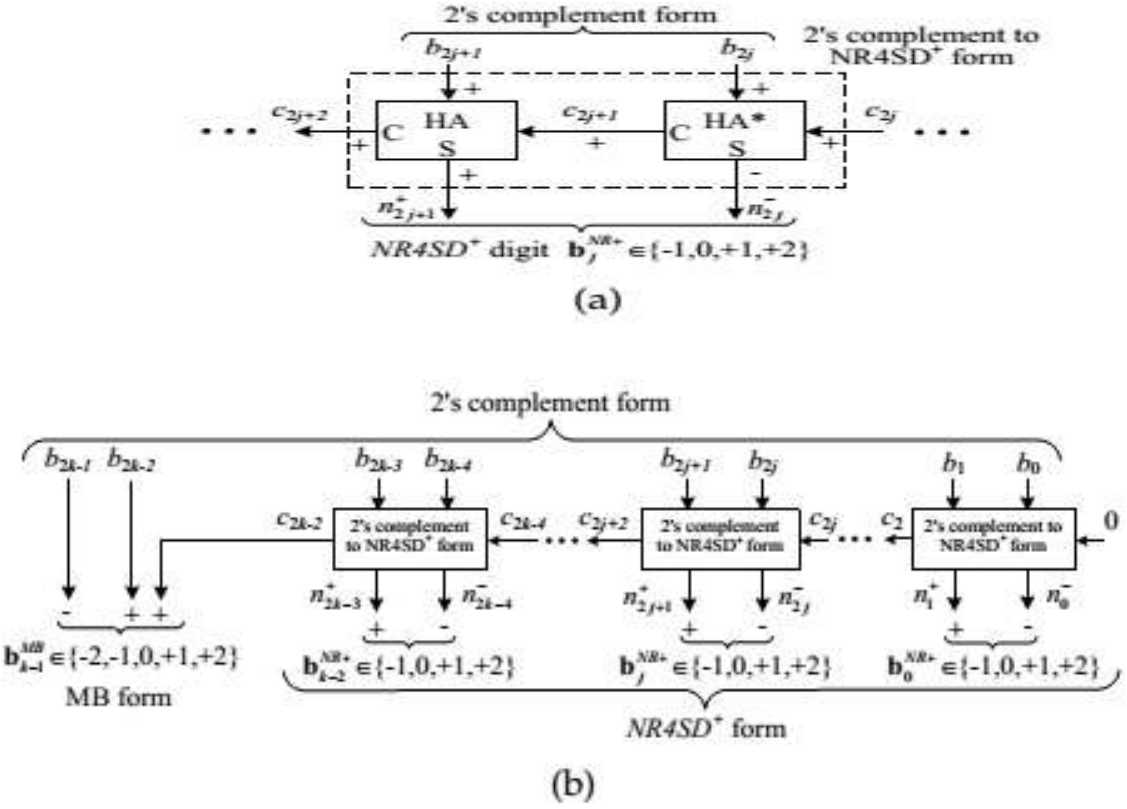

Fig. 4 .Block Diagram of the NR4SD + Encoding Scheme at the (a) Digit and (b) Word Level.

3.1 NR4SD+ Algorithm

Step 1: Consider the initial values j = 0 and C_0=0.
Step 2: Calculate the carry positively signed $C_{2j+1\ (+)}$ and the negatively signed sum n2j(-) of a HA* with inputs b2j (+) and c2j(+) (Fig. 2a). The carry c2j+1 and the sum n2j of the HA* relate to its inputs as follows:
2c2j+1 n2j= b2j + c2j The outputs of the HA* are analyzed at gate level in the following equations:
c2j+1 = b2j _ c2j; n2j= b2j c2j:
Step 3: Calculate the carry c2j+2 and the sum n+2j+1ofa HA with inputs b2j+1 and c2j+1.c2j+2 = b2j+1 ^ c2j+1;
n+2j+1= b2j+1 c2j+1:
Step 4: Calculate the value of the b NR +j digit. B NR +j= 2n+2j+1n2j: (7)
Equation (7) results from the fact that n+2j+1is positively signed and n2jis negatively signed.
Step 5: j: = j + 1.
Step 6: If (j < k 1), go to Step 2. If (j = k 1), encode the most significant digit according to MB algorithm and considering the three consecutive bits to be b2k 1, b2k 2 and c2k 2 (Fig. 2b). If (j = k), stop.
Table 3 shows how the NR4SD+digits are formed. Equations (8) show how the NR4SD+ encoding signals one +j , one j and two +j of Table 4 are generated.
One +j= n+2j+1^ n2j;
One j= n+2j+1^ n2j;
Two +j= n+2j+1^ n2j:
The minimum and maximum limits of the dynamic range in the NR4SD+form are 2n 12n 42n 6 1 < 2n 1and 2n 1+2n 3+2n 5+ +2 > 2n 11. As observed in the NR4SD encoding technique, the NR4SD+form has larger dynamic range than the 2's complement form.

That result when applying the corresponding encoding techniques to each value of N we considered. We added a bar above the negatively signed digits in order to distinguish them from the positively signed ones.

IV. Pre-Encoded NR4SD Multipliers Design

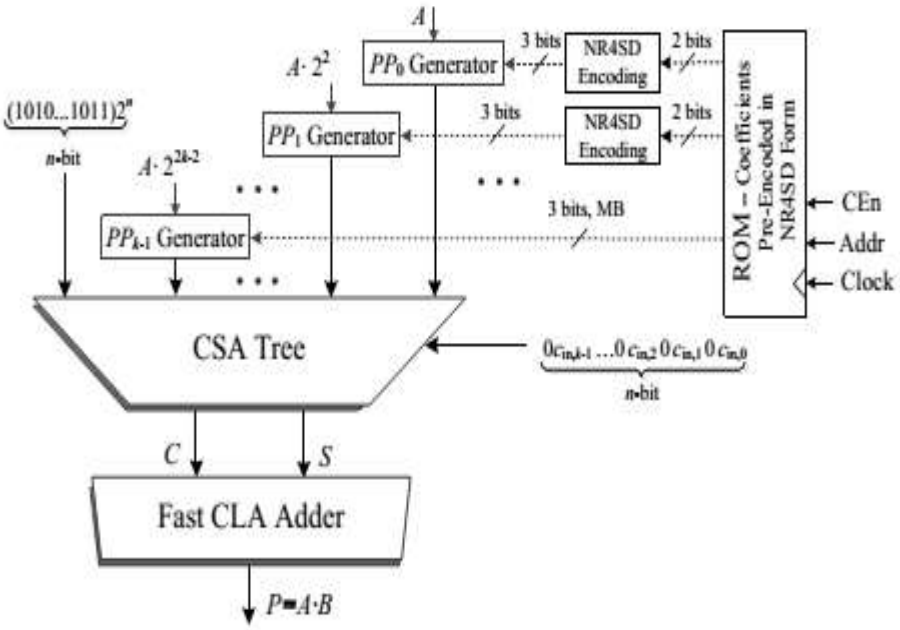

Fig.5 System Architecture of the NR4SD Multiplier

The system architecture for the pre-encoded NR4SD multipliers is presented in Two bits are now stored in ROM: n2j+1, n+2j (Table 2) for the NR4SDor n+2j+1, n2j(Table 3) for the NR4SD+form. In this way, we reduce the memory requirement to +1 bits per coefficient while the corresponding memory required for the pre-encoded MB scheme is 3n/2 bits per coefficient. Thus, the amount of stored bits is equal to that of the conventional MB design, except for the most significant digit that needs an extra bit as it is MB encoded. Compared to the pre-encoded MB multiplier, where the MB encoding blocks are omitted, the pre-encoded NR4SD multipliers need extra hardware to generate the signals of (6) and (8) for the NR4SD and NR4SD+ form, respectively.

Each partial product of the pre-encoded NR4SD– and NR4SD+ multipliers is implemented respectively, except for the P Pk 1 that corresponds to the most significant digit. As this digit is in MB form, we use the PPG for the s j bit. The partial products, properly weighted, and the correction term (COR) of (11) are fed into a CSA tree. The input carry cin; j of (11) is calculated as cin; j = two j_ one j and cin; j = one for the NR4SD– and NR4SD+ pre-encoded multipliers, respectively, The carry-save output of the CSA tree is finally summed using a fast CLA adder.

V.PROPOSED METHOD

5.1 Pre-encoded modified booth encoding:

The system architecture for the pre-encoded modified booth multiplier is presented in two bits are now stored in Rom $n^-_{2j+1,}\ n^+_{2j}$ for nr4sd$^-$ or $n^+_{2j+1,}n^-_{2j}$ for nr4sd$^+$. In this way we reduce the memory requirements to n+ 1 bit per coefficients while corresponding memory required for the pre-encoded mb scheme 3n/2 bits per coefficient.

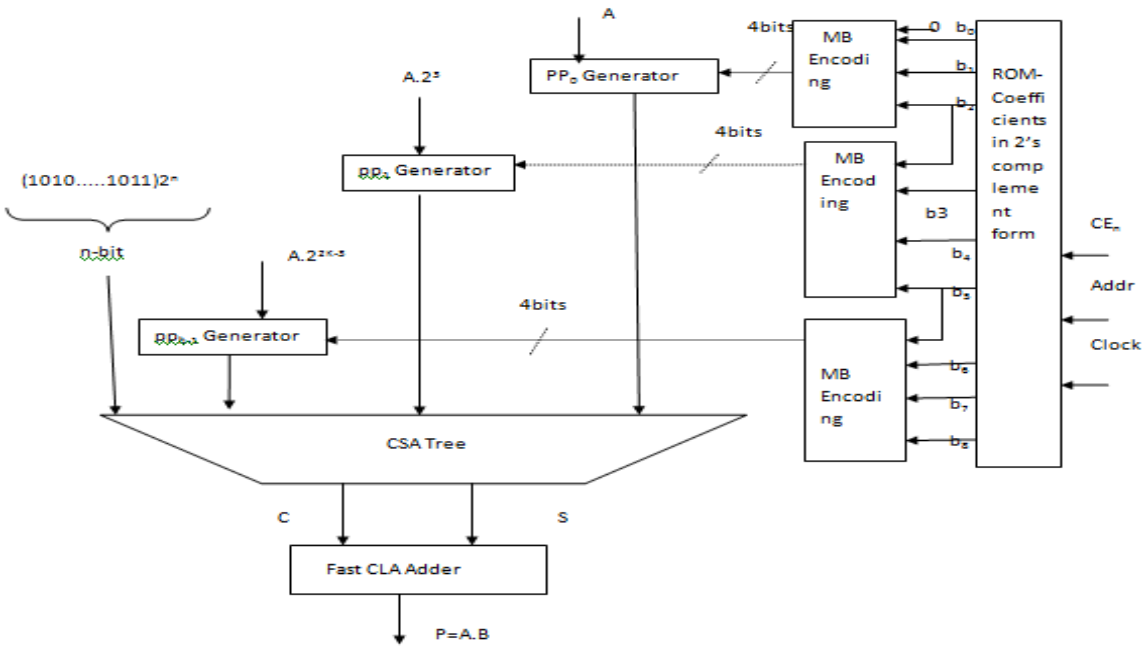

Fig.6.System architecture of mb multiplier

In this Rom is stored as coefficients pre-encoded in mb form. In nr4sd encoding technique radix-4 is used. In modified booth encoding technique radix value increased to 8.as the radix value increases number of the partial products will be reduced. Each of the pp generator output given to carry save adder. The carry save output of the csa tree is finally summed using a fast cla adder.

The generation of the bit pi of the partial product PPj is illustrated at gate level. For the computation of the least and most significant bits of PPj, a1=0 and an= an-1After shaping the partial products, they are added, properly weighted, through a Carry Save Adder (CSA) tree along with the correction term. The CS output of the tree is leaded to a fast Carry Look Ahead (CLA) adder to form the final result P = X * Y. In the pre-encoded MB multiplier scheme, the coefficient B is encoded off-line according to the conventional MB form. The resulting encoding signals of B are stored in a ROM. This contains the ROM with coefficients in 2's complement form and the MB encoding circuit, is now totally replaced by the ROM with coefficients in MB form. The MB encoding blocks of Fig. 1 are omitted. The new ROM is used to store the encoding signals of B and feed them into the partial product generators (PPj Generators PPG) on each clock cycle. Since the n-bit coefficient B needs three bits per digit when encoded in MB form, the ROM width requirement is 3n/2 bits per coefficient. Thus, the width and the overall size of the ROM are increased by 50% compared to the ROM of the conventional scheme.

5.2 Radix 8 Booth encoder:

Radix-8 Booth recoding applies the same algorithm as that of Radix-4, but now we take quartets of bits instead of triplets. Each quartet is codified as a signed digit using Radix-8 algorithm reduces the number of partial products to n/3, where n is the number of multiplier bits. Thus it allows a time gain in the partial products summation

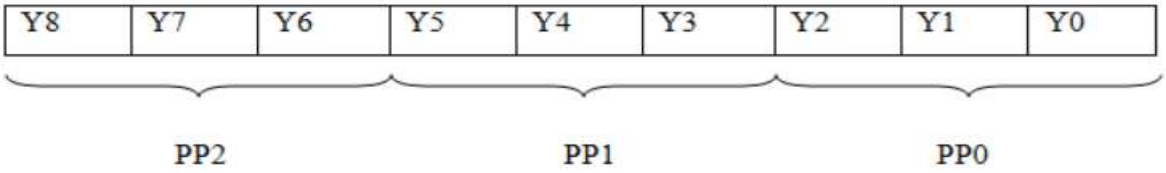

Fig.7.Grouping of multiplier terms

Table-2

Radix-8 Modified booth encoding

B_{2j+2}	B_{2j+1}	B_{2j}	B_{2j-1}	b_j^{mb}
0	0	0	0	0
0	0	0	1	+1
0	0	1	0	+1
0	0	1	1	+2
0	1	0	0	+2
0	1	0	1	+3
0	1	1	0	+3
0	1	1	1	+4
1	0	0	0	+4
1	0	0	1	-3
1	0	1	0	-3
1	0	1	1	-2
1	1	0	0	-2
1	1	0	1	-1
1	1	1	0	-1
1	1	1	1	0

VI .SIMULATION RESULT

The architecture of pre-encoded multipliers is designed. The programming language used in this is verilog HDL and simulated using 13.2 and isim simulator. Design properties are Spartan 3E family, XC3S500E device with speed grade -5.

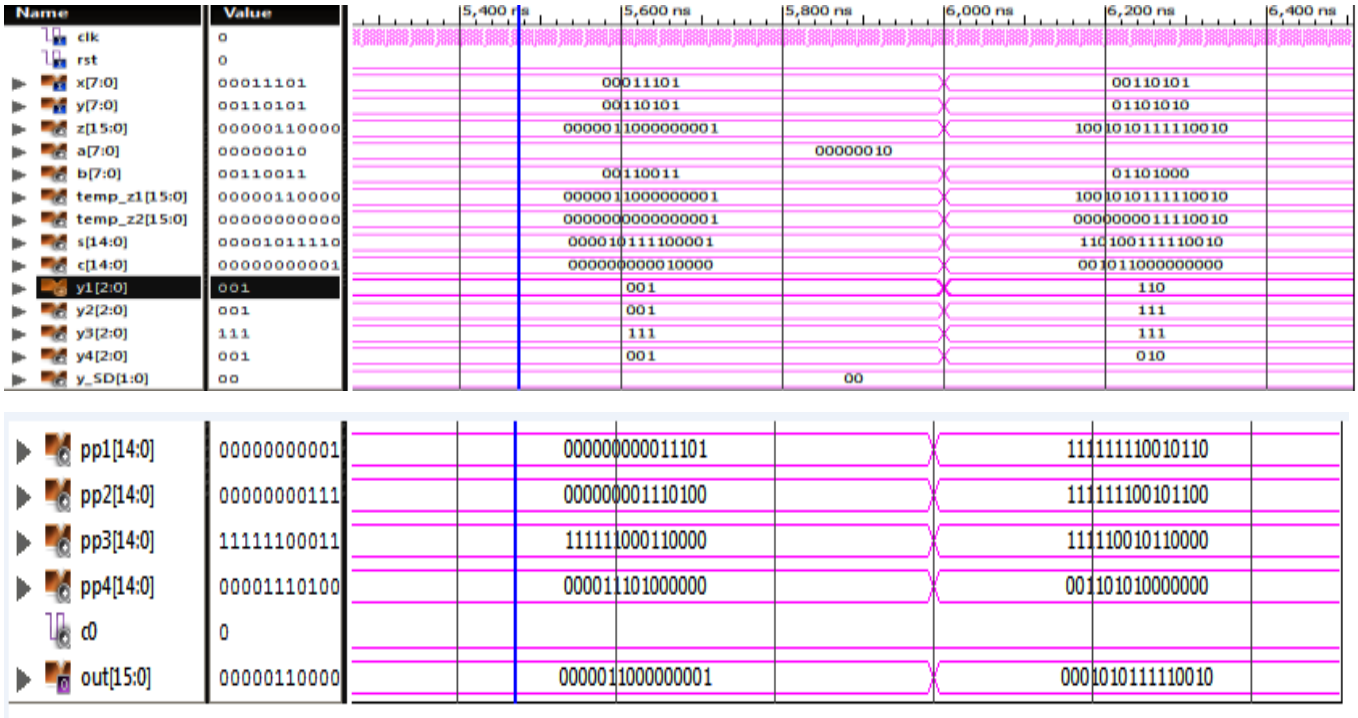

Fig8. Simulation result for nr4sd multiplier

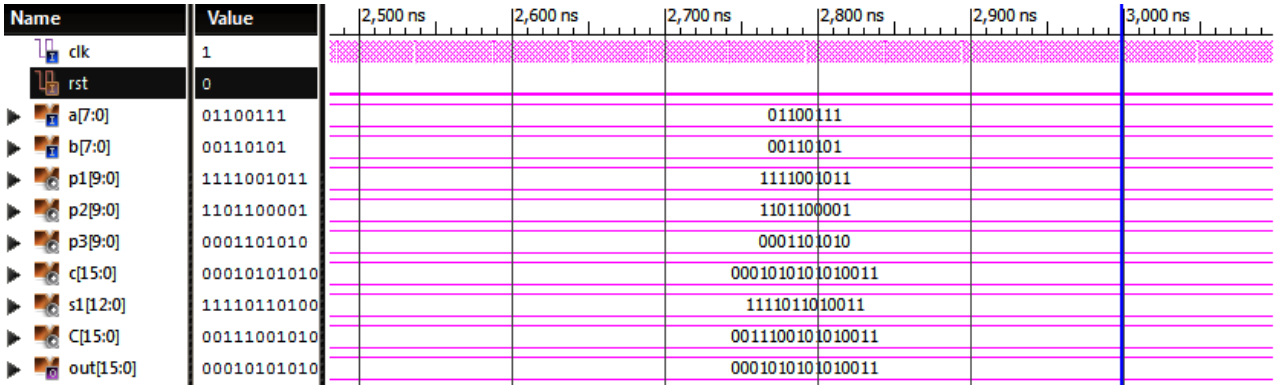

Fig.9. Simulation result for radix-8 modified booth multiplier

VII .COMPARISIONS

In sectionVI simultaion result are presented .comparision of nr4sd multiplier and radix-8 modified booth multiplier is discussed in this section for synthesis results and device properties spartan3E family, VQ100 package,XC3S500E device with a speed grade of -5 is used.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Latches	8	9,312	1%	
Number of 4 input LUTs	209	9,312	2%	
Number of occupied Slices	118	4,656	2%	
Number of Slices containing only related logic	118	118	100%	
Number of Slices containing unrelated logic	0	118	0%	
Total Number of 4 input LUTs	209	9,312	2%	
Number of bonded IOBs	34	66	51%	
IOB Flip Flops	15			
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	3.75			

Fig.10.area report for nr4sd mutiplier

```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: No path found
   Minimum input arrival time before clock: 23.302ns
   Maximum output required time after clock: 6.216ns
   Maximum combinational path delay: No path found
```

Fig.11.Delay report for nr4sd multiplier

Area and delay reports of the nr4sd multiplier is shown in fig.10 and fig.11 from device utilization summary it is noticed that 4656 slices are available in this device,but only 118 slices are used in this design and number of 4 input luts are available 9312 but this design utilizes only 209 4 input luts and delay observed in this method is 23.302ns.

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	87	9,312	1%	
Number of occupied Slices	44	4,656	1%	
Number of Slices containing only related logic	44	44	100%	
Number of Slices containing unrelated logic	0	44	0%	
Total Number of 4 input LUTs	87	9,312	1%	
Number of bonded IOBs	22	66	33%	
Average Fanout of Non-Clock Nets	3.71			

Fig.12. Area report for radix-8 modified booth multiplier

```
Timing Summary:
---------------
Speed Grade: -5

    Minimum period: No path found
    Minimum input arrival time before clock: No path found
    Maximum output required time after clock: No path found
    Maximum combinational path delay: 13.756ns
```

Fig.13. delay report for radix-8 modified booth multiplier

Area and delay reports of the radix-8 modified booth multiplier is shown in fig.12 and fig.13 from device utilization summary it is noticed that 4656 slices are available in this device,but only 44 slices are used in this design and number of 4 input luts are available 9312 but this design utilizes only 87 4 input luts and delay observed in this method is 13.756ns. This radix-8 modified booth multiplier shows better result when compared to nr4sd multiplier in terms of area and delay.

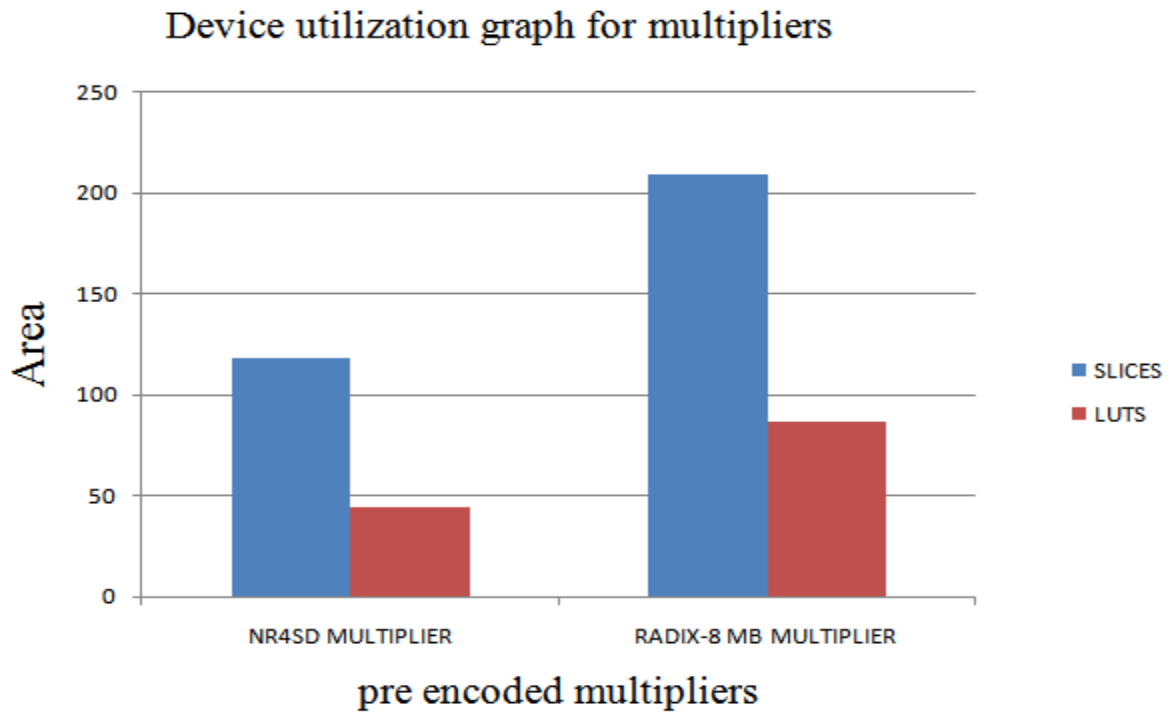

Fig.14 area report for nr4sd multiplier and radix-8 mb multiplie

By device utilization summary area utilized for the nr4sd multiplier and radix-8 mb multiplier is noticed and shows the comparision clearly.radix-8 mb multiplier occupies less area compared to nr4sd multiplier .

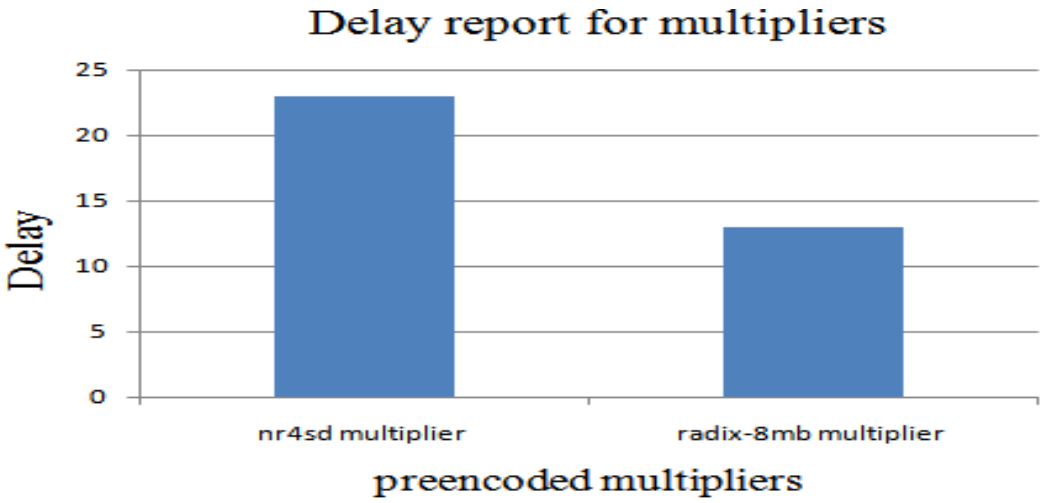

fig15.comparision delay report for nr4sd multiplier and radix-8 mb multiplier

comparision of delay for nr4sd multiplier and radix-8 multiplier are noticed.radix-8 mb multiplier has less delay than nr4sd multiplier.

VII. CONCLUSION

New designs of pre-encoded multipliers are evaluated by off-line encoding the standard coefficients and storing them in system memory. The proposed encoding these coefficients in the pre encoded radix-8 modified booth form. This pre-encoded mb multiplier design reduce the number of partial products because of increasing the radix value when compared to non redundant radix-4 signed digit encoding[NR4SD] form. Extensive experimental analysis verifies the better results in terms of area and delay when compared to nr4sd multiplier.

ACKNOWLEDGEMENT

The authors would like to thank the our beloved principal Dr .G. Srinivasa Rao and vice Principal Dr. P.Srinivasa Raju for shri Vishnu engineering college for women for guidance and encouragement to do research by providing facilities and also like to acknowledge the anonymous reviewers for their suggestions that helps to improve the presentation of this paper.

REFERENCES

[1] K. Yong-Eun C. Kyung Ju, J.G. Chung, and X. Huang, Csd based programmable multiplier design for predetermined coefficient groups, IEICE Trans. Fundamentals. Electron. Commun. Computer. Sci., vol. 93, no. 1, pp. 324–326, 2010.

[2] A. Jacobson, D. Truong, and B. Baas, The design of a reconfigurable continuous-flow mixed-radix fft processor, in IEEE Int. Symp on Circuits and Syst. (ISCAS 2009), May 2009, pp.1133–1136

[3] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. John Wiley & Sons, 2007.

.[4] Z. Huang and M. Ercegovac, High-performance low-power left-to-right array multiplier design, IEEE Trans. Computer., vol. 54, no. 3, pp. 272–283, Mar. 2005.

[5] Z. Huang, High-level optimization techniques for low-power multiplier design, Ph.D. dissertation, Department of Computer Science, University of California, Los Angeles, CA, 2003.

[6] W.-C. Yeh and C.-W. Jen, High-speed booth encoded parallel multiplier design, IEEE Trans. Computer., vol. 49, no. 7, pp. 692–701, Jul. 2000.

[7] M. Kolluru, Audio decoder core constants rom optimization, Patent US 6 108 633, Aug., 2000.

[8] Y. T. Han, J. S. Koh, and S. H. Kwon, Synthesis filter for mpeg-2 audio decoder, Patent US 5 812 979, Sep., 1998.

[9] M. Kolluru, Audio decoder core constants room optimization, Patent US 6 108 633, Aug., 2000.
[10] H.-Y. Lin, Y.-C. Chao, C.-H. Chen, B.-D. Liu, and J.-F. Yang, Combined 2d transform and quantization architectures for h.264 video coders, in IEEE Int. Symp. on Circuits and Syst. (ISCAS 2005), vol. 2, May 2005, pp. 1803–1805.
[11] G. Pastuszak, A high performance architecture of the double mode binary coder for h.264.avc, IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 7, pp. 949–960, Jul.2008.
[12] J. Park, K. Muhammad, and K. Roy, High performance fir filter design based on sharing Multiplication, IEEE Trans. Very Large Scale Integer. (VLSI) Syst., vol. 11, no. 2, pp. 244253, Apr. 2003.
[13] K.S. Chong, B.H. Gwee , and J. S. Chang, A 16channel low power non uniform spaced filter bank core for digital hearing aids," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 9, 853-857, Sep. 2006.
[14] B. Paul, S. Fujita, and M. Okajima, Rom based logic design: A low power 16 bit multiplier, IEEE J. Solid State Circuits, vol. 44, no. 11, pp. 2935-2942, Nov. 2009.
[15] M. D. Ercegovac and T. Lang, "Multiplication, in Digital Arithmetic. San Francisco: Morgan Kaufmann, 2004, pp. 181245.
[16] N. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4thed. USA: Addison Wesley Publishing Company, 2010.
[17] Dual dsp plus micro for audio applications, feb2003, TDA7503 Data sheet, S micro electronics.
[18] C. Xu, X. Dong, N. Jouppi, and Ixia, Design implications of memoristor based rram cross point structures, in Design, Automation Test in Europe Conf. Exhibition (DATE), Mar. 2011