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

Area Efficient NR4SD Encoding for Pre-Encoded Multipliers

B. Gowtam Kumar
Department of Electronics & Communication Engineering,
BVC College of Engineering, Palacharla, Rajanagaram, A.P - 533294, India.

Mr.S.A.Vara Prasad
Department of Electronics & Communication Engineering,
BVC College of Engineering, Palacharla, Rajanagaram, A.P - 533294, India.

ABSTRACT:

A multiplier is one of the key hardware blocks in most digital signal processing (DSP) systems. Typical DSP applications where a multiplier plays an important role include digital filtering, digital communications and spectral analysis. Multiplications are very expensive and slow the overall operation. The performance of many computational problems are often dominated by the speed at which a multiplication operation can be executed.. In this paper, we introduce architecture of pre-encoded multipliers for Digital Signal Processing applications based on off-line encoding of coefficients. To this extend, the Non-Redundant radix-4 Signed-Digit (NR4SD) encoding technique, which uses the digit values {− 1 ,0 , +1 , +2} or {− 2 , − 1 , 0 , +1}, is proposed leading to a multiplier design with less complex partial products implementation. Extensive experimental analysis verifies that the proposed pre-encoded NR4SD multipliers, including the coefficients memory, are more area and power efficient than the conventional Modified Booth scheme.All the synthesis and simulation results of the proposed pre-encoded NR4SD multipliers are performed on Xilinx ISE 14.7using Verilog HDL.

Index Terms—Multiplying circuits, modified Booth encoding, pre-encoded multipliers, VLSI implementation.

INTRODUCTION

Multimedia and digital signal processing (DSP) applications (e.g., fast Fourier transform (FFT), audio/video CoDecs) 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 nonzero 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 predetermined coefficients attaining at the same time high efficiency. Modified Booth (MB) encoding [4], [5], [6], [7] tackles the aforementioned 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.

Cite this article as: B. Gowtam Kumar & Mr.S.A.Vara Prasad, "Area Efficient NR4SD Encoding for Pre-Encoded Multipliers", International Journal & Magazine of Engineering, Technology, Management and Research, Volume 5 Issue 6, 2018, Page 48-56.

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

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] CoDecs, 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 SignedDigit (NR4SD) encoding scheme extending the serial encoding techniques of [6], [18]. The proposed NR4SD encoding scheme uses one of the following sets of digit values: {-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 proposed representation are encoded according to NR4SD except the most significant one that is MB encoded. Using the proposed encoding formula, we preencode the standard coefficients and store them into a ROM in a condensed form (i.e., 2 bits per digit). Compared to the preencoded 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.

MODIFIED BOOTH ALGORITHM

Modified Booth is a redundant radix-4 encoding technique [6], [7]. Considering the multiplication of the 2's complement numbers A, B, each one consisting of n = 2k bits, B can be represented in MB form as:

$$B = \langle b_{n-1} \dots b_0 \rangle_{2's} = -b_{2k-1}2^{2k-1} + \sum_{i=0}^{2k-2} b_i 2^i$$
$$= \langle \mathbf{b}_{k-1}^{MB} \dots \mathbf{b}_0^{MB} \rangle_{MB} = \sum_{j=0}^{k-1} \mathbf{b}_j^{MB} 2^{2j}. \quad (1)$$

Digits b^{MB}_j {-2,-1,0,+1,+2}, 0≤j≤k-1; are formed as follows:

$$\mathbf{b}_j^{MB} = -2b_{2j+1} + b_{2j} + b_{2j-1}, \quad (2)$$

where b_{-1}=0. Each MB digit is represented by the bits s, one and two (Table 1). The bit s shows if the digit is negative (s = 1) or positive (s = 0). One shows if the absolute value of a digit equals 1 (one = 1) or not (one = 0). Two shows if the absolute value of a digit equals 2 (two = 1) or not (two = 0). Using these bits, we calculate the MB digits b^{MB}_j as follows:

$$\mathbf{b}_j^{MB} = (-1)^{s_j} \cdot (one_j + 2two_j) \quad (3)$$

Equations (4) form the MB encoding signals.

$$s_j = b_{2j+1}, \quad one_j = b_{2j-1} \oplus b_{2j},$$
$$two_j = (b_{2j+1} \oplus b_{2j}) \wedge \overline{one}_j. \quad (4)$$

TABLE 1 Modified Booth Encoding

b_{2j+1}	b_{2j}	b_{2j-1}	$\mathbf{b}_j^{MB}$	s_j	one_j	two_j
0	0	0	**0**	0	0	0
0	0	1	**+1**	0	1	0
0	1	0	**+1**	0	1	0
0	1	1	**+2**	0	0	1
1	0	0	**−2**	1	0	1
1	0	1	**−1**	1	1	0
1	1	0	**−1**	1	1	0
1	1	1	**0**	1	0	0

NON-REDUNDANT RADIX-4 SIGNED-DIGIT ALGORITHM

In this section, 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 bNR j take one of four values {− 2 , − 1 , 0 , +1}or bNR j ε{− 1 , 0 , +1 , +2} at the NR4SD or NR4SDþ 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., b^{MB}_{k-1} {-2,-1,0,+1,+2}). The NR4SD and NR4SDþ

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

encoding algorithms are illustrated in detail in Figs. 1 and 2, respectively.

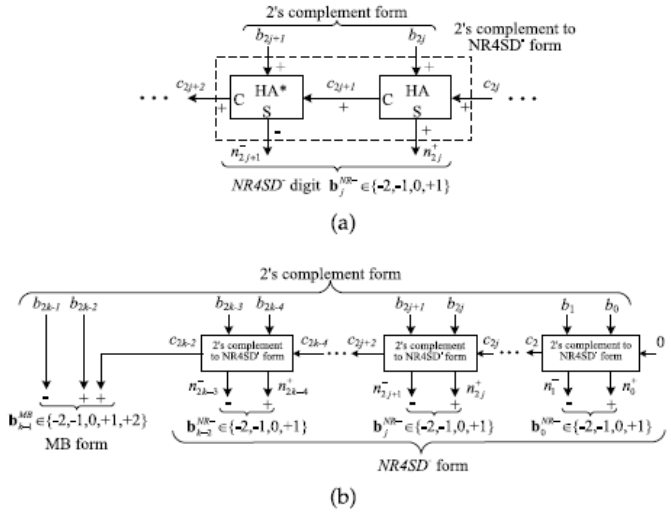

Fig.1. Block diagram of the NR4SD encoding scheme at the (a) digit and (b) word level.

NR4SD Algorithm

Step1. Consider the initial values j =0 and c0 =0.

Step2. Calculate the carry c_{2j+1} and the sum n^{+}_{2j} of a half adder (HA) with inputs b_{2j} and c_{2j} (Fig.1a).

$$c_{2j+1} = b_{2j} \wedge c_{2j}, \quad n^{+}_{2j} = b_{2j} \oplus c_{2j}.$$

Step3. Calculate the positively signed carry $_{c2j+2}$ (+) and the negatively signed sum n 2_{j+1} (-) of a HA* with inputs b_{2j+1} (+) and c_{2j+1} (+) (Fig. 1a).

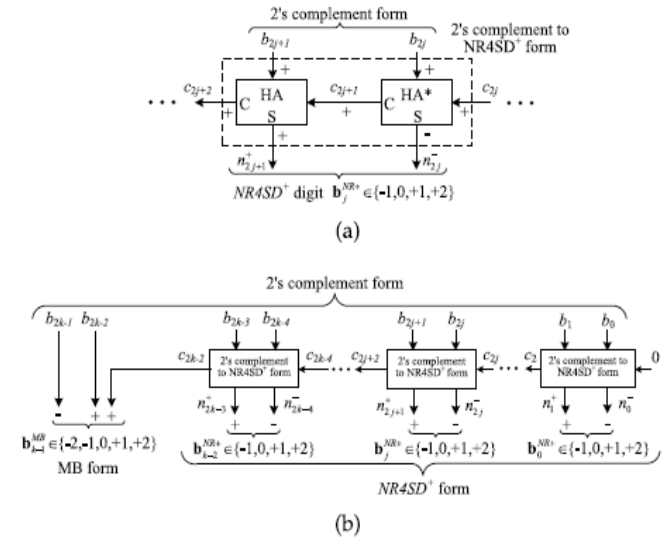

Fig.2. Block diagram of the NR4SDþ encoding scheme at the (a) digit and (b) word level.

The outputs c_{2j+2} and n 2_{j+1} of the HA* relate to its inputs as follows:

$$2c_{2j+2} - n^{-}_{2j+1} = b_{2j+1} + c_{2j+1}.$$

The following Boolean equations summarize the HA* operation:

$$c_{2j+2} = b_{2j+1} \vee c_{2j+1}, \quad n^{-}_{2j+1} = b_{2j+1} \oplus c_{2j+1}.$$

Step4. Calculate the value of the b^{NR-}_{j} digit.

$$\mathbf{b}^{NR-}_{j} = -2n^{-}_{2j+1} + n^{+}_{2j} \quad (5)$$

Equation (5) results from the fact that n^{-}_{2j+1} is negatively signed and $n^{+}2j$ is positively signed.

Step5. j :=j + 1.

Step6. If If (j < k_ 1), go to Step 2. If (j = k -1), encode the most significant digit based on the MB algorithm and considering the three consecutive bits to be b2k-1, b2k-2 and c2k-2 (Fig. 1b). If (j = k), stop.

Equations (6) show how the NR4SD⁻ encoding signals one^{+}_{j}, one^{-}_{j} and two^{-}_{j} of Table 2 are generated.

$$\begin{aligned} one^{+}_{j} &= \overline{n^{-}_{2j+1}} \wedge n^{+}_{2j}, \\ one^{-}_{j} &= n^{-}_{2j+1} \wedge n^{+}_{2j}, \\ two^{-}_{j} &= n^{-}_{2j+1} \wedge \overline{n^{+}_{2j}}. \end{aligned} \quad (6)$$

TABLE 2 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}	$\mathbf{b}^{NR-}_{j}$	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

Table 2 shows how the NR4SD⁻ digits are formed.

The minimum and maximum limits of the dynamic range in the NR4SD⁻ form are -2^{n-1} -2^{n-3} -2^{n-5} -….-2 < -2n-1 and 2n-1 + 2n-4 + 2n-6 +…. +1 > 2^{n-1} - 1. We observe

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

that the NR4SD$^-$ form has larger dynamic range than the 2's complement form.

NR4SDþ Algorithm

Step1. Consider the initial values j = 0 and c0 =0.

Step2. Calculate the positively signed carry c_{2j+1} (+) and the negatively signed sum n^-_{2j} (-) of a HA* with inputs b_{2j} (+) and c_{2j} (+) (Fig. 2a). The carry c_{2j+1} and the sum n^-_{2j} of the HA* relate to its inputs as follows:

$$2c_{2j+1} - n^-_{2j} = b_{2j} + c_{2j}.$$

The outputs of the HA* are analyzed at gate level in the following equations:

$$c_{2j+1} = b_{2j} \vee c_{2j}, \quad n^-_{2j} = b_{2j} \oplus c_{2j}.$$

Step3. Calculate the carry c_{2j+2} and the sum n^+_{2j+1} of a HA with inputs b_{2j+1} and c_{2j+1}.

$$c_{2j+2} = b_{2j+1} \wedge c_{2j+1}, \quad n^+_{2j+1} = b_{2j+1} \oplus c_{2j+1}.$$

Step4. Calculate the value of the b^{NR+}_j digit

$$\mathbf{b}^{NR+}_j = 2n^+_{2j+1} - n^-_{2j} \quad (7)$$

Step5. j := j + 1.

Step6. 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 NR4SDþ digits are formed. Equations (8) show how the NR4SDþ encoding signals one^+_j, one^-_j and two^+_j of Table 3 are generated.

TABLE 3 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}	$\mathbf{b}^{NR+}_j$	one^+_j	one^-_j	two^+_j
0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	+1	1	0	0
0	1	0	0	1	1	+1	1	0	0
0	1	1	0	1	0	+2	0	0	1
1	0	0	0	1	0	+2	0	0	1
1	0	1	1	0	1	−1	0	1	0
1	1	0	1	0	1	−1	0	1	0
1	1	1	1	0	0	0	0	0	0

$$\begin{aligned} one^+_j &= n^+_{2j+1} \wedge n^-_{2j}, \\ one^-_j &= n^+_{2j+1} \wedge n^-_{2j}, \\ two^+_j &= n^+_{2j+1} \wedge \overline{n^-_{2j}}. \end{aligned} \quad (8)$$

The minimum and maximum limits of the dynamic range in the NR4SDþ form are $-2^{n-1} - 2^{n-4} - 2^{n-6} - \ldots -1 < -2^{n-1}$ and $2^{n-1} + 2^{n-3} + 2^{n-5} + \ldots + 2 > 2^{n-1} - 1$. As observed in the NR4SD_ encoding technique, the NR4SDþ form has larger dynamic range than the 2's complement form.

Considering the 8-bit 2's complement number N, Table 4 exposes the limit values $-2^8 = -128$, $2^8 - 1 = 127$, and two typical values of N, and presents the MB, NR4SD_ and NR4SDþ digits 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.

TABLE 4 Numerical Examples of the Encoding Techniques

2's Complement	10000000	10011010	01011001	01111111
Integer	−128	−102	+89	+127
Modified Booth	$\bar{2}000$	$2\bar{2}1\bar{2}$	$12\bar{2}1$	$200\bar{1}$
NR4SD$^-$	$\bar{2}000$	$1\bar{2}1\bar{2}$	$2\bar{2}\bar{2}1$	$200\bar{1}$
NR4SD$^+$	$\bar{2}000$	$\bar{2}122$	1121	$200\bar{1}$

PRE-ENCODED MULTIPLIERS DESIGN

In this section, we explore the implementation of pre-encoded multipliers. One of the two inputs of these multipliers is pre-encoded either in MB or in NR4SD$^-$/NR4SD$^+$ representation. We consider that this input comes from a set of fixed coefficients (e.g., the coefficients for a number of filters in which this multiplier will be used in a dedicated system or the sine table required in an FFT implementation). The coefficients are encoded off-line based on MB or NR4SD algorithms and the resulting bits of encoding are stored in a ROM. Since our purpose is to estimate the efficiency of the proposed multipliers, we first present a

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

review of the conventional MB multiplier in order to compare it with the pre-encoded schemes.

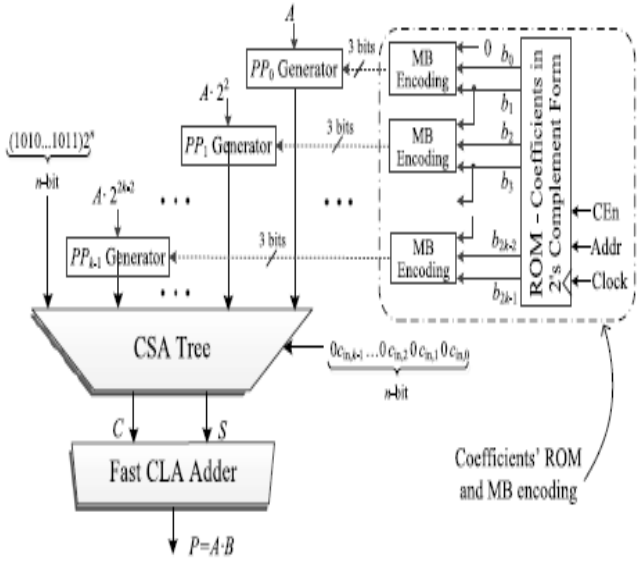

Fig.3.System architecture of the conventional MB multiplier.

Conventional MB Multiplier

Fig.3 presents the architecture of the system which comprises the conventional MB multiplier and the ROM with coefficients in 2's complement form. Let us consider the multiplication A •B. The coefficient B =$(b_{n\_1} \ldots b_0)_{2's}$ consists of n = 2k bits and is driven tothe MB encoding blocks from a ROM where it is stored in 2's complementform. It is encoded according to the MB algorithm(Section 2) and multiplied by A =$(a_{n\_1} \ldots a_0)_{2's}$, which is in 2'scomplement representation. We note that the ROM data bus widthequals the width of coefficient B (n bits) and that it outputs onecoefficient on each clock cycle.

The k partial products are generated as follows:

$$PP_j = A \cdot \mathbf{b}_j^{MB} = \bar{p}_{j,n}2^n + \sum_{i=0}^{n-1} p_{j,i}2^i \quad (9)$$

The generation of the ith bit $p_{j,i}$of the partial product PP_j is illustrated at gate level in Fig. 4a [6], [7]. For the computation of the least and most significant bits of PP_j, we consider $a_{-1} = 0$ and $a_n = a_{n-1}$, respectively.

After shaping the partial products, they are added, properly weighted, through a carry save adder (CSA) tree along with the correction term (COR):

$$P = A \cdot B = COR + \sum_{j=0}^{k-1} PP_j 2^{2j} \quad (10)$$

$$COR = \sum_{j=0}^{k-1} c_{in,j}2^{2j} + 2^n\left(1 + \sum_{j=0}^{k-1} 2^{2j+1}\right) \quad (11)$$

wherecin,j = ($one_j \Lambda two_j$)Λs_j (Table 1). The CS output of the tree is leaded to a fast carry look ahead (CLA) adder [19] to form the final result P ¼=A • B (Fig. 3).

Pre-Encoded MB Multiplier Design

In the pre-encoded MB multiplier scheme, the coefficient B is encoded off-line according to the conventional MB form (Table 1). The resulting encoding signals of B are stored in a ROM. The circled part of Fig. 3, which contains the ROM with coefficients in 2's complement form and the MB encoding circuit, is now totally replaced by the ROM of Fig. 5. The MB encoding blocks of Fig. 3 are omitted. The new ROM of Fig. 5 is used to store the encoding signals of B and feed them into the partial product generators (PPj Generators—PPG) on each clock cycle.

Targeting to decrease switching activity, the value '1' of sj in the last entry of Table 1 is replaced by '0'. The sign sj is now given by the relation:

$$s_j = b_{2j+1} \oplus (b_{2j+1} \wedge b_{2j} \wedge b_{2j-1}) \quad (12)$$

As a result, the PPG of Fig. 4a is replaced by the one of Fig. 4b. Compared to (4), (12) leads to a more complex design. However, due to the pre-encoding technique, there is no area/delay overhead at the circuit.

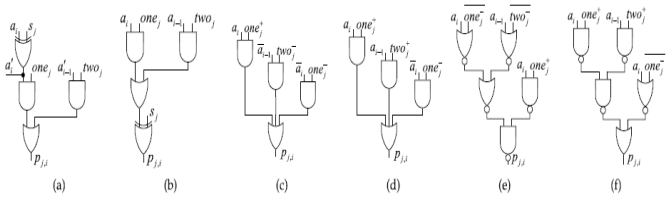

Fig. 4. Generation of the ith Bit pj,i of $PP_j$for a) Conventional, b) Pre-Encoded MB Multipliers, c) $NR4SD^-$, d) $NR4SD^+$ Pre-Encoded Multipliers, and e) $NR4SD^-$, f) $NR4SD^+$ Pre-Encoded Multipliers after reconstruction.

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

The partial products, properly weighted, and the COR of (11) are fed into a CSA tree. The input carry cin;j of (11) is computed as cin;j ¼ sj based on (12) and Table 1. The CS output of the tree is finally merged by a fast CLA adder. However, the ROM width is increased. Each digit requests three encoding bits (i.e., s, two and one (Table 1)) to be stored in the ROM. 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 coeffi- cient. Thus, the width and the overall size of the ROM are increased by 50 percent compared to the ROM of the conventional scheme (Fig. 3).

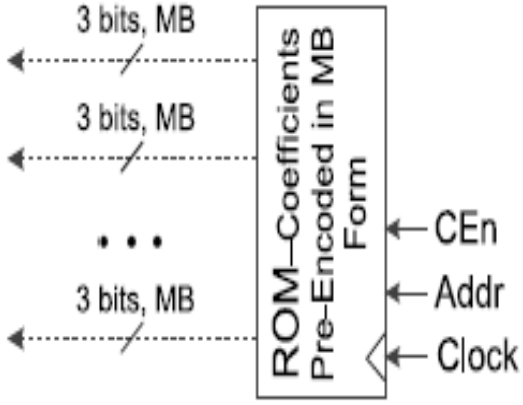

Fig.5. The ROM of pre-encoded multiplier with standard coefficients in MB Form.

Pre-Encoded NR4SD Multipliers Design

The system architecture for the pre-encoded NR4SD multipliers is presented in Fig. 6. Two bits are now stored in ROM: n^{-}_{2j+1}, n^{+}_{2j} (Table 2) for the NR4SD or n^{+}_{2j+1}, n_{-2j} (Table 3) for the NR4SDþ form. In this way, we reduce the memory requirement to n + 1 bits per coefficient while the corresponding memory required for the pre-encoded MB scheme is 3n/2 bits per coeffi- cient. 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 preencoded 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 NR4SDþ form, respectively. The NR4SD encoding blocks of Fig. 6 implement the circuitry of Fig. 7.

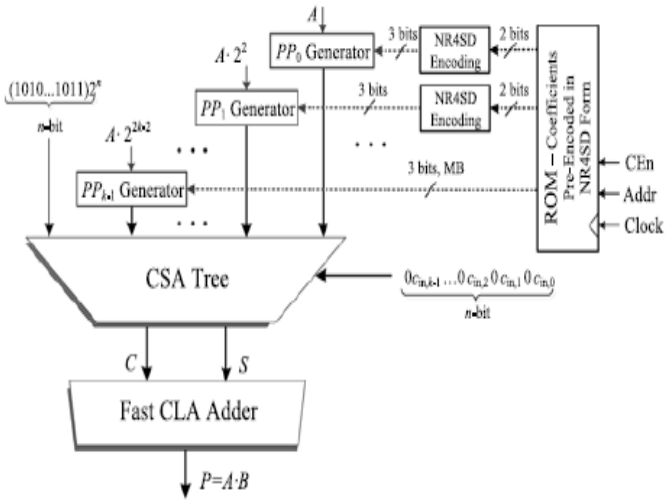

Fig.6. System architecture of the NR4SD multipliers.

Each partial product of the pre-encoded NR4SD and NR4SDþ multipliers is implemented based on Figs. 4c and 4d, respectively, except for the PP_{k-1} that corresponds to the most significant digit. As this digit is in MB form, we use the PPG of Fig. 4b applying the change mentioned in Section 4.2 for the s_j bit. The partial products, properly weighted, and the COR of (11) are fed into a CSA tree. The input carry cin,j of (11) is calculated as cin,j = $two^{-}_j Vone^{-}_j$ and cin,j = one^{-}_j for the $NR4SD^{-}$ and $NR4SD^{+}$ pre-encoded multipliers, respectively, based on Tables 2 and and 3. The carry-save output of the CSA tree is finally summed using a fast CLA adder.

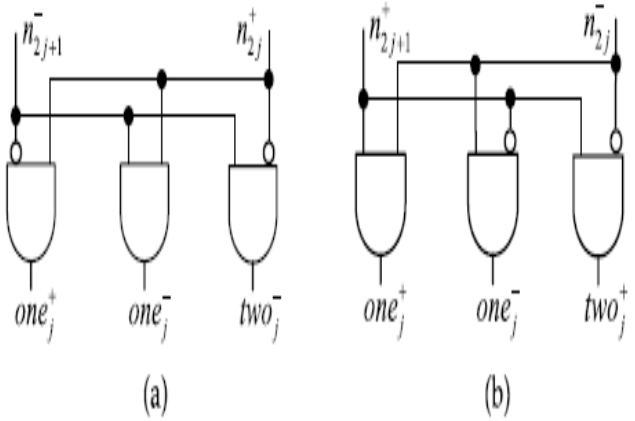

Fig.7. Extra circuit needed in the NR4SD multipliers to complete the (a) NR4SD and (b) NR4SDþ encoding.

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

SYNTHESIS AND SIMULATION RESULTS

We implemented in Verilog the multiplier designs of Table 5. The PPGs for the NR4SD, NR4SDþ multipliers (Figs. 4c and 4d, respectively) contain a large number of inverters since all the A bits are complemented in case of a negative digit. In order to avoid these inverters and, thus, reduce the area/ power/delay of NR4SD, NR4SDþ pre-encoded multipliers, the PPGs for the NR4SD, NR4SDþ multipliers were designed based on primitive NAND and NOR gates, and replaced by Figs. 4e and 4f, respectively.

TABLE 5 Multiplier Designs

Design		Input A	Input B Encoding	
			Type	Technique
Conventional MB		2's complement *n*-bit	MB	MB encoding
Pre-Encoded	MB		MB	Fully Pre-Encoded
	$NR4SD^-$		$NR4SD^-$	Partially Pre-Encoded
	$NR4SD^+$		$NR4SD^+$	

All the synthesis and simulation results of the proposed pre-encoded NR4SD multipliers are performed using Verilog HDL. The synthesis and simulation are performed on Xilinx ISE 14.7. The corresponding simulation results of the proposed pre-encoded NR4SD multipliers are shown below.

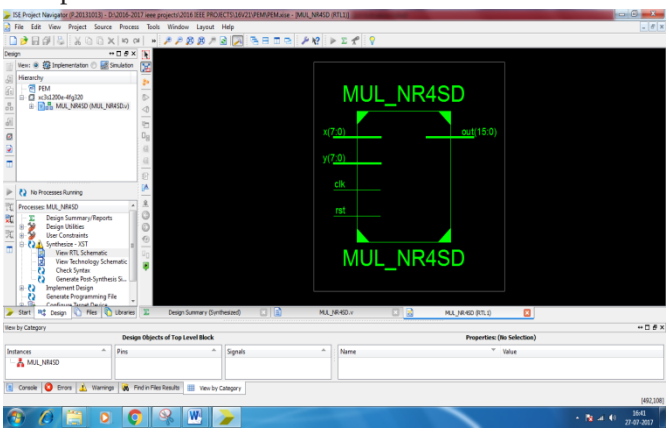

Fig.8. RTL schematic of Top-level of Proposed Pre-Encoded NR4SD Multipliers

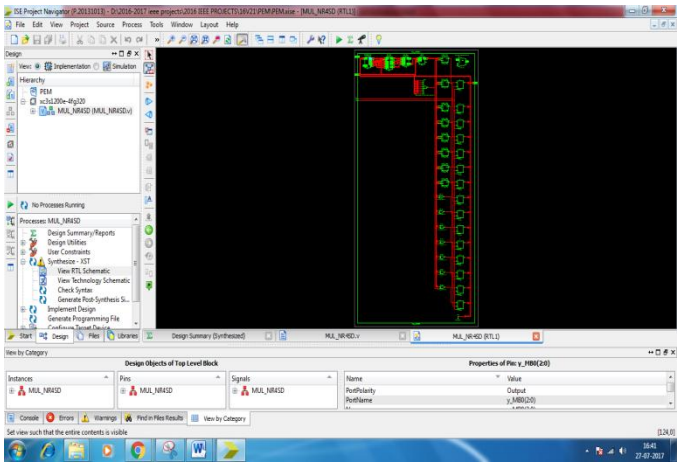

Fig.9.RTL schematic of internal block of Proposed Pre-Encoded NR4SD Multipliers

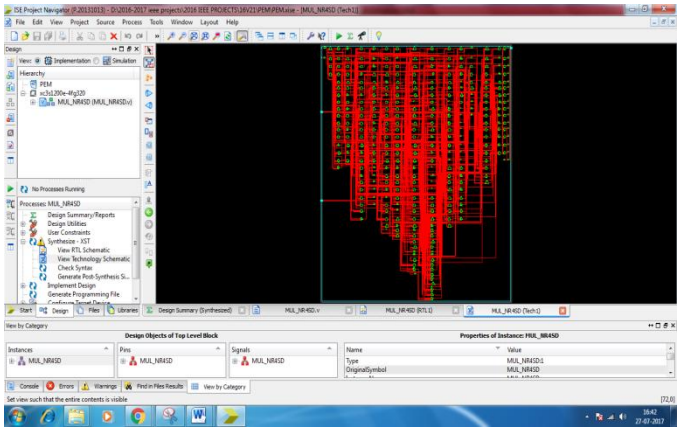

Fig.10. Technology schematic of internal block of Proposed Pre-Encoded NR4SD Multipliers

MUL_NR4SD Project Status (07/27/2017 - 16:41:06)			
Project File:	PEM.xise	**Parser Errors:**	No Errors
Module Name:	MUL_NR4SD	**Implementation State:**	Synthesized
Target Device:	xc3s1200e-4fg320	• **Errors:**	No Errors
Product Version:	ISE 14.7	• **Warnings:**	27 Warnings (0 new)
Design Goal:	Balanced	• **Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	
Environment:	System Settings	• **Final Timing Score:**	

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	120	8672	1%
Number of Slice Flip Flops	8	17344	0%
Number of 4 input LUTs	210	17344	1%
Number of bonded IOBs	34	250	13%
Number of GCLKs	1	24	4%

Detailed Reports					[-]
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Thu 27. Jul 16:41:05 2017	0	27 Warnings (0 new)	1 Info (0 new)

Fig.11.Synthesis report of Proposed Pre-Encoded NR4SD Multipliers

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

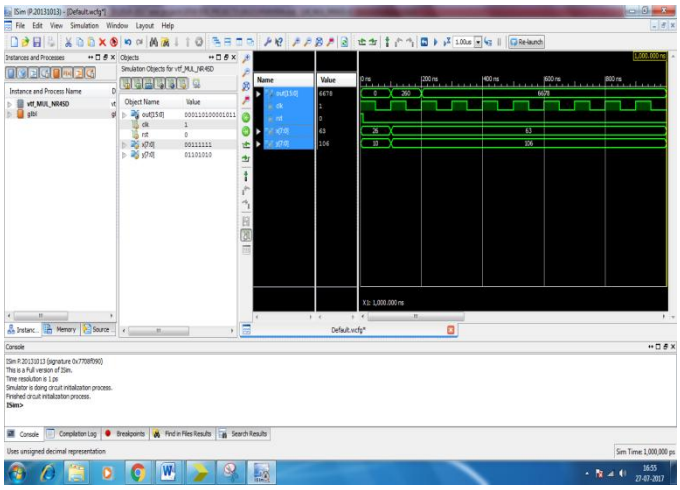

Fig.12. Simulated output for Proposed Pre-Encoded NR4SD Multipliers

CONCLUSION

All the synthesis and simulation results of the proposed pre-encoded NR4SD multipliers are performed on Xilinx ISE 14.7using Verilog HDL.

In this paper, new designs of pre-encoded multipliers are explored by off-line encoding the standard coefficients and storing them in system memory. We propose encoding these coefficients in the Non-Redundant radix-4 Signed-Digit (NR4SD) form. The proposed pre-encoded NR4SD multiplier designs are more area and power efficient compared to the conventional and pre-encoded MB designs. Extensive experimental analysis verifies the gains of the proposed pre-encoded NR4SD multipliers in terms of area complexity and computational delays compared to the conventional MB multiplier.

REFERENCES

[1] G. W. Reitwiesner, "Binary arithmetic," Adv. Comput., vol. 1, pp. 231–308, 1960.

[2] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation, Hoboken, NJ, USA: Wiley, 2007.

[3] Y.-E. Kim, K.-J.Cho, J.-G.Chung, and X. Huang, "CSD-based programmable multiplier design for predetermined coefficient groups," IEICE Trans. Fundam.Electron.Commun.Comput.Sci., vol. 93, no. 1, pp. 324–326, 2010.

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

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

[6] Z. Huang, "High-level optimization techniques for low-power multiplier design," Ph.D. dissertation, Dept. Comput. Sci., Univ. California, Los Angeles, CA, USA, 2003.

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

[8] Y.-E. Kim, K.-J.Cho, and J.-G. Chung, "Low power small area modified booth multiplier design for predetermined coefficients," IEICE Trans. Fundam. Electron.Commun.Comput. Sci., vol. E90-A, no. 3, pp. 694–697, Mar. 2007.

[9] C. Wang, W.-S.Gan, C. C. Jong, and J. Luo, "A low-cost 256-point FFT processor for portable speech and audio applications," in Proc. Int. Symp.Integr. Circuits, Sep. 2007, pp. 81–84.

[10] A. Jacobson, D. Truong, and B. Baas, "The design of a reconfigurable continuous-flow mixed-radix FFT processor," in Proc. IEEE Int. Symp. Circuits Syst., May 2009, pp. 1133–1136.

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

[12] M. Kolluru, "Audio decoder core constants rom optimization," Patent US6108633, Aug. 2000. [13] H.-

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

Y. Lin, Y.-C.Chao, C.-H.Chen, B.-D.Liu, and J.-F. Yang, "Combined 2-d transform and quantization architectures for h.264 video coders," in Proc. IEEE Int. Symp. Circuits Syst., May. 2005, vol. 2, pp. 1802–1805.

[14] 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.

[15] J. Park, K. Muhammad, and K. Roy, "High-performance fir filter design based on sharing multiplication," IEEE Trans. Very Large Scale Integr. Syst., vol. 11, no. 2, pp. 244–253, Apr. 2003.

[16] K.-S. Chong, B.-H.Gwee, and J. S. Chang, "A 16-channel low-power nonuniform spaced filter bank core for digital hearing aids," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 9, pp. 853–857, Sep. 2006.

[17] B. Paul, S. Fujita, and M. Okajima, "Rom-based logic (RBL) design: A lowpower 16 bit multiplier," IEEE J. Solid-State Circuits, vol. 44, no. 11, pp. 2935–2942, Nov. 2009.

[18] M. D. Ercegovac and T. Lang, "Multiplication," in Digital Arithmetic. San Francisco, CA, USA: Morgan Kaufmann, 2004, pp. 181–245.

[19] N. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. Reading, MA, USA: Addison-Wesley, 2010.

[20] "Dual dsp plus micro for audio applications," Feb. 2003, TDA7503 Datasheet, STMicroelectronics.

[21] C. Xu, X. Dong, N. Jouppi, and Y. Xie, "Design implications of memristorbasedrram cross-point structures," in Proc. Design, Automation Test Eur. Conf. Exhib., Mar. 2011, pp. 1–6.