International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 23 Issue 8 –NOVEMBER 2016.

FPGA Implementation of Low Power Carry Skip adder based Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Signed-Digit Encoding

SURESH SANAPALA [#1] and G V VINOD[*2]

[#] *M.Tech, VLSI AND EMBEDDED SYSTEMS, Godavari Institute of Engineering & Technology (GIET), Rajahmundry, Andhra Pradesh, India*

***Abstract*— in this paper we are discussed about the new design of pre-encoded multiplier are explored at offline the standard co efficient and storing them in system memory. The co efficient is used in non redundant radix 4 signed digit form. This encoding technique is less complex partial product implementation and more area and power efficient design. Analysis is verifies the proposed system is efficient from the existing system.**

***Index Terms*— Non redundant radix 4 signed digit (NR4SD), pre encoding.**

I. INTRODUCTION

In the multiplication in the digital signal processing application and multimedia are carry out to large number. The coefficient of these systems does not change in execution time. The multiplier is the basic component of these applications, so its affect the system architecture and its operations.

Nowadays more number of multipliers is used in the different fields. The CSD (canonic signed digit) multipliers comprise the fewest non-zero partial products which is reduce the switching activity. Another one is Booth multiplier [2]; this is reducing the partial product to half of the level, so the area is reducing more.

To generate the product based on 2's complement format, and final addition fast carry-propagation adder is required. The problem of facing in designing a final adder is that the input signals do not arrive simultaneously. Different techniques have been used to eliminate or reduce the final adder delay [1].

The three dimensional reduction methods [3], is used to increase the speed and its algorithm is design with MBE, so the area of the multiplier and is reduced and speed is increased.

Hence many more error compensation approaches [4] have been proposed to decrease the truncation error of fixed width MB multipliers. By using statistical analysis and linear regression analysis the compensation value generated reduce significantly the mean error. The proposed multiplier of NR4SD (non redundant radix 4 signed digit) multiplier this is reduce the memory size.

The rest of this paper to introduce the existing system for the paper is discussed in section II. Then, in section III, the proposed system of NR4SD multiplier is present. Section IV presents the simulation result of the paper. Finally Section V presents the conclusion of the paper.

II. EXISTING SYSTEM

Modified Booth (MB) encoding 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.

A. Modified booth algorithm:

Modified Booth (MB) is a redundant radix-4 encoding technique. 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 $\mathbf{b}^{MB}{}_j \in \{-2, -1, 0, +1, +2\}$, $0 \le j \le k-1$, are formed as follows:

$$\mathbf{b}^{MB}{}_j = -2b2j+1 + b2j + b2j-1, \quad (2)$$

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

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).

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

Equations (4) form the MB encoding signals.

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

B. *Pre-Encoded NR4SD Multipliers Design*

The architecture for existing NR4SD multiplier design is shown in figure 1. In this used to ROM, NR4SD encoder, PP Generator, CSA and CLA adder. The use of CSA and CLA adder increase the area and delay of the architecture.

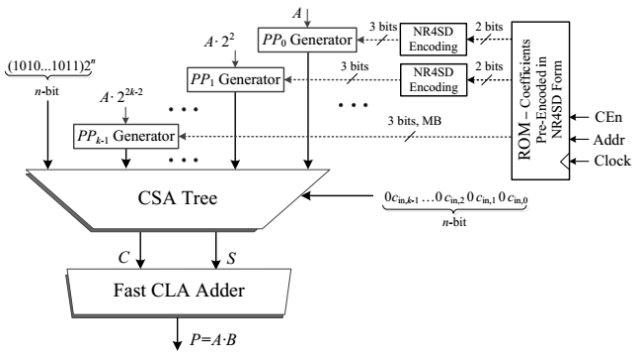

Figure 1: System Architecture of the NR4SD Multipliers with CSA and CLA adder.

III. PROPOSED SYSTEM

In the proposed system reduce the delay and efficient architecture of the pre-encoder multiplier design based on replace the CSA and CLA to the carry skip adder.

A. *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 $\mathbf{b}^{NR}{}_j^-$ take one of four values: $\{-2, -1, 0, +1\}$ or $\mathbf{b}^{NR+}{}_j \in \{-1, 0, +1, +2\}$ at the NR4SD− or NR4SD+ algorithm, respectively. Only four different values are used and not five as in MB algorithm, which leads to $0 \le j \le k-2$. As we need to cover the dynamic range of the 2's complement form, the most significant digit is MB encoded (i.e., $\mathbf{b}^{MB}{}_{k-1} \in \{-2,-1, 0, +1, +2\}$). The NR4SD− and NR4SD+ encoding algorithms are illustrated in detail in Fig. 2 and 3, respectively.

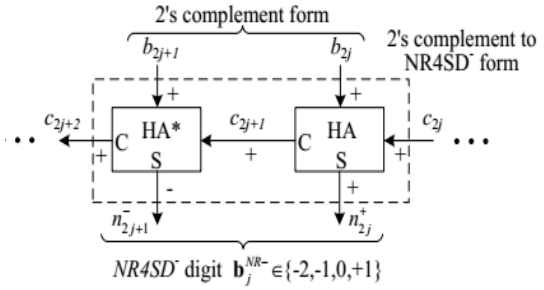

Figure 2: Block Diagram of the NR4SD− Encoding Scheme at the Digit

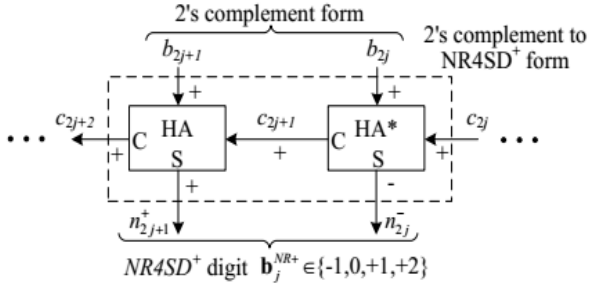

Figure 3: Block Diagram of the NR4SD+ Encoding Scheme at the Digit

B. *NR4SD− Algorithm*

Step 1: Consider the initial values $j = 0$ and $c0=0$.

Step 2: Calculate the carry $c2j+1$ and the sum $n+ 2j$ of a Half Adder (HA) with inputs $b2j$ and $c2j$ Fig. 2. $c2j+1 = b2j \wedge c2j$, $n+ 2j = b2j \oplus c2j$.

Step 3: Calculate the positively signed carry $c2j+2$ (+) and the negatively signed sum $n- 2j+1$ (-) of a Half Adder* (HA*) with inputs $b2j+1$ (+) and $c2j+1$ (+) (Fig. 2). The outputs $c2j+2$ and $n- 2j+1$ of the HA* relate to its inputs as follows: $2c2j+2 - n- 2j+1 = b2j+1 + c2j+1$.

Step 4: Calculate the value of the $\mathbf{b}^{NR}{}_j^-$ digit.

$$\mathbf{b}^{NR-}{}_j = -2n- 2j+1 + n+ 2j. \qquad (5)$$

Equation (5) results from the fact that $n- 2j+1$ is negatively signed and $n+ 2j$ is positively 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 based on the MB algorithm and considering the three consecutive bits to be b_{2k-1}, b_{2k-2} and c_{2k-2}. If ($j = k$), stop. Table 2 shows how the NR4SD− digits are formed.

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}	$\mathbf{b}_j^{NR-}$	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

A. *NR4SD+ Algorithm*

Step 1: Consider the initial values $j = 0$ and $c0=0$.

Step 2: Calculate the carry positively signed $c2j+1$ (+) and the negatively signed sum $n- 2j$ (-) of a HA* with inputs $b2j$ (+) and $c2j$ (+) (Fig. 3). The carry $c2j+1$ and the sum $n- 2j$ of the HA* relate to its inputs as follows:

$$2c2j+1 - n- 2j = b2j + c2j.$$

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

$$c2j+1 = b2j \curlyvee c2j,\ n- 2j = b2j \oplus c2j.$$

Step 3: Calculate the carry $c2j+2$ and the sum $n+ 2j+1$ of a HA with inputs $b2j+1$ and $c2j+1$.

$$c2j+2 = b2j+1 \curlywedge c2j+1,\ n+ 2j+1 = b2j+1 \oplus c2j+1.$$

Step 4: Calculate the value of the $\mathbf{b}^{NR-}{}_j$ digit.

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

Equation (7) results from the fact that $n+ 2j+1$ is positively signed and $n- 2j$ is 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 $c2k-2$. If ($j = k$), stop. Table 3 shows how the NR4SD+ digits are formed.

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}	$\mathbf{b}_j^{NR+}$	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

The system architecture for the pre-encoded NR4SD multipliers is presented in Fig. 4. 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 NR4SD+ 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 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.

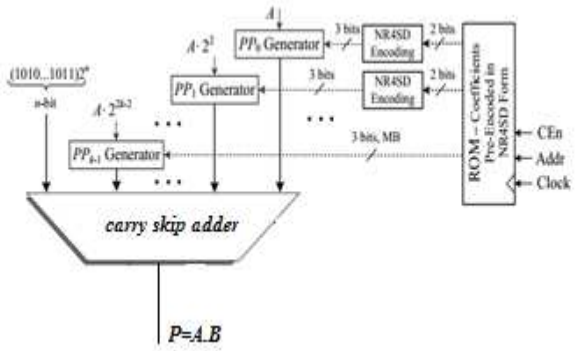

Figure 4: System Architecture of the NR4SD Multipliers.

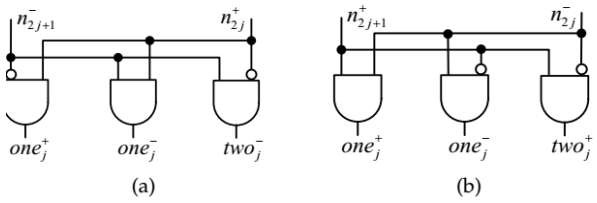

Figure 5: (a) NR4SD− and (b) NR4SD+ Encoding

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 for the NR4SD− and NR4SD+ form, respectively. The NR4SD encoding blocks of Fig. 5 implement the circuitry of Fig. 4. And the Fig 6 shows the logical diagram of the PPG unit.

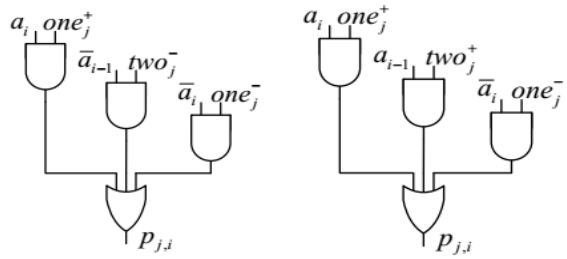

Figure 6: Generation of the ith Bit $p_{j,i}$ of PP_j

IV. SIMULATION AND RESULT

The simulate the proposed system architecture in Modelsim and to analysis the area, power, and delay of the proposed system in Spartan 6 by using Xilinx software. The simulation result for the proposed NR4SD multiplier is shows in figure 7. The synthesis report of the proposed system is shown in figure 8 and figure 9. Finally the comparison of the proposed system is detailed in table I.

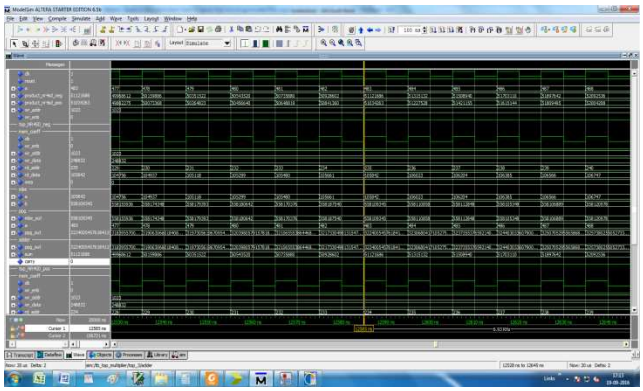

Fig.7: simulation result

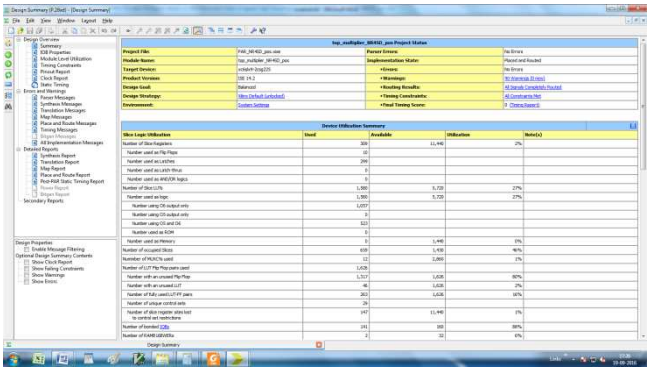

Fig.8: synthesis report for NR4SD+

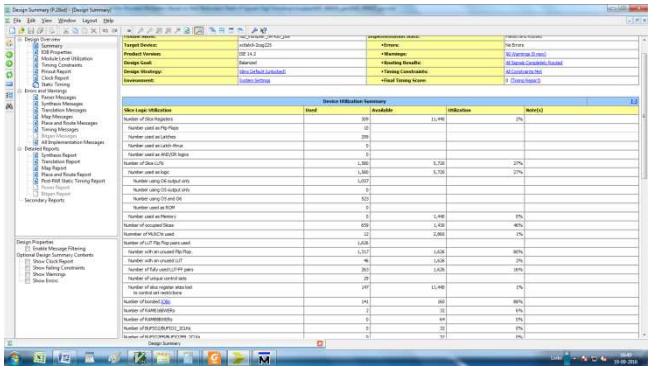

Fig.9: synthesis report for NR4SD-

Table 4: comparison

Parameter	Existing system		Proposed system	
	NR4SD+	NR4SD-	NR4SD+	NR4SD-
Power	27.50mW	26.40mW	0.171W	0.176W
Delay	2.29ns	2.28ns	2.140ns	2.142ns

V. CONCLUSION

We are discussed about the architecture of pre-encoded multiplier are explored at offline the standard co efficient and storing them in system memory. The coefficient is used in non redundant radix 4 signed digit form. This encoding technique is less complex partial product implementation and more area and power efficient design. Analysis is verifies the proposed system is efficient from the existing system and the delay of the proposed system is 2.14ns.

REFERENCES

[1]. R.K. Kolagotla et al., ªVLSI Implementation of a 200-Mhz 16 16 Left-to-Right Carry-Free Multiplier in 0.35m CMOS Technology for Next-Generation DSPs,º Proc. IEEE 1997 Custom Integrated Circuits Conf., pp. 469-472, 1997.

[2]. O.L. MacSorley, ªHigh Speed Arithmetic in Binary Computers,º Proc. IRE, vol. 49, pp. 67-91, 1961.

[3]. V.G. Oklobdzija, D. Villeger, and S.S. Liu, ªA Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach,º IEEE Trans. Computers, vol. 45, no. 3, pp. 294-306, Mar. 1996.

[4]. S.J.Jou,M.H.Tsai, and Y.L.Tsao , "Low-error reduced-width Booth multipliers for DSP applications", IEEE Trans. Circuits Syst.I. Fundam Theory Application, Vol.50, no.11, pp.1470-1474, Nov 2003.

[5]. 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 *Int. Symp. on Integrated Circuits (ISIC 2007)*, Sep. 2007, pp. 81–84.

[6]. 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.

[7]. Y. T. Han, J. S. Koh, and S. H. Kwon, "Synthesis filter for mpeg-2 audio decoder," Patent US 5812979, Sep., 1998.

[8]. M. Kolluru, "Audio decoder core constants rom optimization," Patent US 6108633, Aug., 2000.

[9]. H.-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 *IEEE Int. Symp. on Circuits and Syst. (ISCAS 2005)*, vol. 2, May 2005, pp. 1802–1805.

[10]. 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.

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

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

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