Pre Encoded Multiplier Based on Non Redundant Radix-4 Signed Digit Encoding

G Bharathi1, M.Kalavathi 2
[1]*M.Tech. PG Scholar, Gouthami Institute of Technology & Management For Women, Proddatur*
[2]*Assistant Professor, Gouthami Institute of Technology & Management For Women, Proddatur*

***Abstract*-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.**

I. INTRODUCTION

The speed of multiplication operation is of great importance in digital signal processing as well as in the general purpose processors today. In the past multiplication was generally implemented via a sequence of addition, subtraction, and shift operations. Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Eachstep of addition generates a partial product. In most computers, the operand usually contains the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional representation. It is possible to decompose multipliers into two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them. The basic multiplication principle is twofold i.e. evaluation of partial products and accumulation of the shifted partial products. It is performed by the successive additions of the columns of the shifted partial product matrix. The 'multiplier' is successfully shifted and gates the appropriate bit of the 'multiplicand'. The delayed, gated instance of the multiplicand must all be in the same column of the shifted partial product matrix. They are then added to form the product bit for the particular form. Multiplication is therefore a multi operand operation. To extend the multiplication to both signed and unsigned numbers, a convenient number system would be the representation of numbers in two's complement format. The MAC (Multiplier and Accumulator Unit) is used for image processing and digital signal processing (DSP) in a DSP processor.Algorithm of MAC is Booth's radix-4 algorithm, Modified Booth Multiplier; Wallace tree improves speed and reduces the power.

Translating a decimal fraction into a finite floating-point representation is prone to losing precision as a result of rounding errors. In almost all financial settings, decimal arithmetic is desired to guarantee balances are calculated correctly and lawfully. Some industrial and scientific applications require high-precision decimal arithmetic as well. Software packages have been available for most programming languages so that decimal numbers could be evaluated with decimal arithmetic to avoid error [1,2]. IBM recently departed from this software solution by incorporating a decimal floating-point arithmetic unit in the Power6 and z10 processors. A compelling reason to do such is a report showing that 55% of the numbers stored in the databases of 51 major organizations are decimal. One study shows that for a set of five benchmarks, a 1.3 to 12.8 speedup factor was obtained by simulating a processor using virtual decimal arithmetic hardware against software routines [6]. Applications that spend a large proportion of the time consuming decimal calculation stand to benefit from hardware-based decimal operations. Therefore, research into decimal arithmetic has gained momentum. Decimal renditions

of binary carry-save and carry-look ahead adders have been proposed. Decimal floating-point addition is treated. New decimal multipliers and dividers have also been proposed. New decimal encodings improve the latency and area for decimal partial product generation and reduction for multiplication. Similarly, in [22], a new redundant digit set is used with special encodings called two-valued digits (twits), resulting in a faster implementation of both addition and subtraction. The main motivation behind this paper is to introduce a new signed-digit architecture and objectively compare it with signed and unsigned digit adders. Signed-digit decimal adders have the benefit of carry-free addition although a carry-propagate adder must be used to transform the signed-digit sum into an unsigned sum. In the next two sections, we will present the theory of decimal encodings and signed-digit decimal numbers. In the subsequent sections, brief descriptions of other adders are given: the non speculative multi-operand adder, mixed binary and BCD adder , reduced delay BCD adder , dynamic decimal CLA [11], Svoboda adder, speculative signed-digit adder, decimal carry-free adder and Redundant Binary Coded Decimal (RBCD) adder. Then, the proposed method for signed-digit addition is discussed. A constant addition technique will be applied to both the correction step in signed-digit decimal addition and conversion to binary- coded decimal (BCD). Multi-operand decimal addition based on signed-digit addition will follow. Finally, the synthesis results will be discussed. The main objective of this thesis is to design and implementation of a Multiplier and Accumulator. A multiplier which is a combination of Modified Booth and SPST (Spurious Power Suppression Technique) adder are designed taking into account the less area consumption of booth algorithm because of less number of partial products and more speedy accumulation of partial products and less power consumption of partial products addition using SPST adder approach.

II. LITERATURE SURVEY

C.Grabbe,M.Bednara,J.Teich,presented four high performance GF(2233) multipliers for an FPGA realization and analyzed the time and area complexities. The finite field elements are represented as polynomial basis and normal basis. In polynomial basis, classical multiplier and Karatsuba multipliers were designed. The advantage of classical multiplier is regular structure and pipelined operation. The disadvantage is high space complexity. In Karatsuba multiplier the advantage is less number of gates are required. The normal basis multipliers are Massey-Omura and Sunar-Koc multiplier. The advantage of Massey-Omura is high flexibility and Sunar-Koc is total number of gates are reduced.

P.L.Montgomery,[2] presented Karatsuba Ofman algorithm for multiplying two polynomials. Here multiplication of 5 -term, 6-term and 7-term polynomials are provided with scalar multiplication of 13,17 and 22.Using 6-term polynomial only leads to better asymptotic performance than standard karatsuba.

C.Paar, [3] presented a new bit parallel structure for a multiplier with low space complexity in Galois field is introduced. Finite field of GF(2n) is considered and field extension of GF((2n)m). The field elements are represented in the canonical base or in standard basis. Field of the form GF((2n)) are referred as composite field. Karatsuba Ofman algorithm is used to multiply two polynomials effectively. Advantages are complexity is reduced by introducing the composite field. The main disadvantage is security is less and does not have a regular structure.

C.Rebeirno and D.Mukhopadhyay, [4] presented a hybrid technique which has a better area delay product. Masking strategies are introduced to prevent power based side channel attacks on the multiplier. SCAs are the biggest threat to modern cryptography systems. In basic recursive KM, the number LUTs required to combine the partial products is much lower but it cannot applied directly to ECC. The hybrid KM requires least resources as compared to other KMs used for elliptic curve arithmetic; also it has a unique architecture. Demerits are it is not efficient for FPGA platform as the number of utilized LUTs is 65%.

A.Reyhani-Masoleh and A.Hasan, [5] presented a new bit parallel structure for the polynomial basis multiplication which is applicable to all type of irreducible binary polynomial. The main advantage of this new formulation is that it can be used with any field defining irreducible polynomial. Then a bit parallel hardware architecture generalization is provided. The architecture consist of two parts IP network and Q network.

The space and time complexities are analysed as a function of reduction matrix . the main advantage is only fewer number of lines are required on the bus.

F.Rodriguez and C.K.Koc,[6] presented the KaratsubaOfman Algorithm in which the degree of defining the irreducible polynomial can be arbitrarily selected by the designer allowing the usage of prime degrees. Here finite field and composite field are considered. Composite multiplication is performed by n-bit Karatsuba multiplier. The main advantage is number of multiplication is reduced. The composite field multiplication is performed by binary Karatsuba multipliers. The advantage is improved gate complexity .The disadvantage is wastage of several arithmetic operation.

B.Sunar,[7] presented the subquadratic complexity multipliers for even characteristic field extension. A short convolution algorithm named Winograd short convolution algorithm were designed to improve the space and time complexity. A certain Winograd short convolutional algorithm is essentially identical to the Karatsuba algorithm. The merits of Winograd techniques are it can be easily built for any desired length; it is simple and uniform construction. The main disadvantages are appears to have less structure and cause additional wire delay in VLSI implementation.

A.Weimerskirch and C.Paar, presented the classical Karatsuba algorithm for polynomial multiplication. Three methods considered are digital method, Fast Fourier transform method and Karatsuba method. The Karatsuba algorithm is derived in two ways namely Chinese Remainder Theorem and Simple Algebraic Transform KA is applied recursively if the degree of polynomial is 2i, where i>1 is a positive integer. Advantage-squaring the polynomial is easily performed; adding a dummy coefficients the complexity is reduced. Disadvantage is a number of intermediate results have been stored due to the recursive nature of KA.

J.VonzurGathen and J.Shokrollahi, presented different possibilities for implementing the Karatsuba multiplier for polynomials over F2 on FPGA. Classical multiplier, Karatsuba multiplier and a hybrid design were provided. The Karatsuba multiplier has the lowest crossover point with the classical algorithm. In hardware, the algorithmic and platform dependent optimizations yield efficient designs. The resources usage of polynomial multipliers is decreased by using both the algorithmic and platform dependent method. The hybrid design is used to minimize the total arithmetic cost and results in significant area savings.

G.Zhou, H.Michalik and L.Hinsenkamp, [10] addresses the efficient and high throughput implementations of AES-GSM optimized for FPGAs. The two main components in GCM are an AES engine and a finite field multiplier over GF (2128).The complexity analysis presented is based on FPGA primitives (LUTs). Modular multiplication consists of two steps: first a classical multiplication and then a modular reduction. The straight forward multiplier is used to get speed efficient design while a Karatsuba multiplier is used to get an area efficient design. Merits are reduced hardware complexity and high throughput.

III. PROPOSED SYSTEM

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.

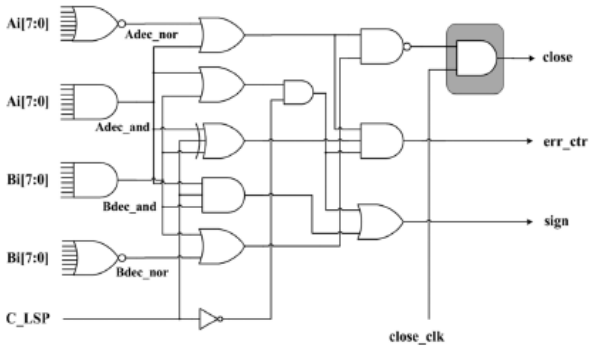


Fig .1 *Modfed booth encoder*

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.

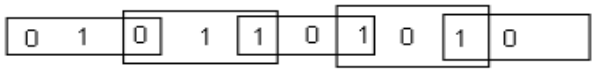


Fig.2 *Grouping of bits from the multiplier term*

Each block is decoded to generate the correct partial product. The encoding of the multiplier Y, using the modified booth algorithm, generates the following five signed digits, -2, -1, 0, +1, +2. Each encoded digit in the multiplier performs a certain operation on the multiplicand, X, as illustrated in Table 1

Block	Re - coded digit	Operation on X
000	0	0 X
001	+1	+1 X
010	+1	+1 X
011	+2	+2 X
100	-2	-2 X
101	-1	-1 X
110	-1	-1 X
111	0	0 X

Table 1: Generaton of signed bits

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.

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;+1gor 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. 1 and 2, respectively.

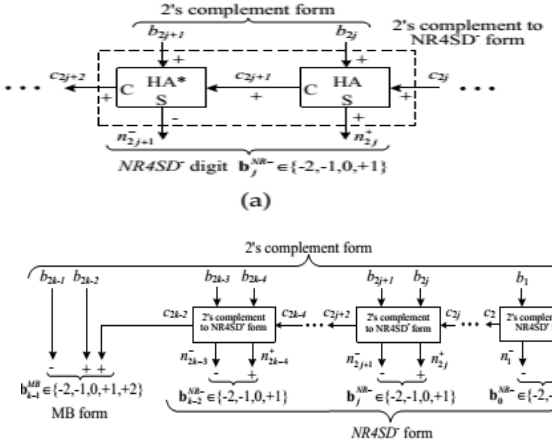


Fig. 3. *Block Diagram of the NR4SD Encoding Scheme at the (a) Digit and (b) Word Level.*

NR4SD¯Algorithm

Step 1: Consider the initial values j = 0 and C_0=0.

Step 2: Calculate the carry C_{2j+1} and the sum n+2jof a Half Adder (HA) with inputs b2j and C_{2j} (Fig. 1a).

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

Step 3: Calculate the positively signed carry c2j+2 (+) and the negatively signed sum n2j+1(-) of a Half Adder* (HA*) with inputs b2j+1 (+) and c2j+1 (+) (Fig. 1a). The outputs c2j+2 and n2j+1of 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:

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

Step 4: Calculate the value of the bNRj digit.

$$\mathbf{b}_j^{NR-} = -2n^{-}_{2j+1} + n^{+}_{2j}.$$

Equation (5) results from the fact that n2j+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 b2k 1,b2k 2 and c2k 2 (Fig. 1b). If (j = k), stop.

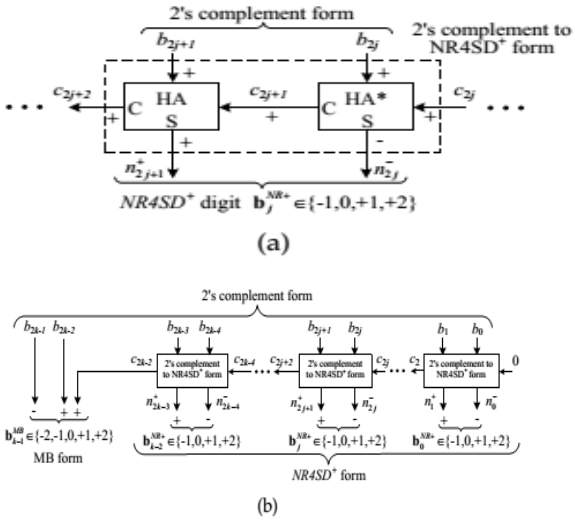


Fig.4. *Block Diagram of the NR4SD + Encoding Scheme at the (a) Digit and (b) Word Level.*

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

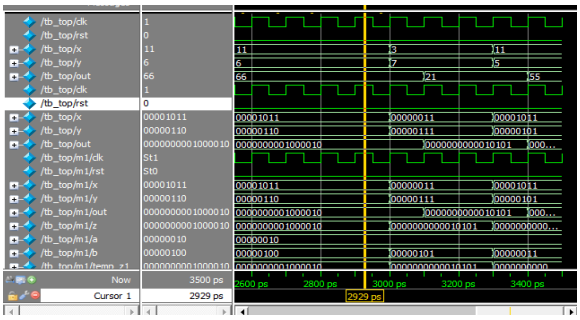
Table 2 NR4SD⁻ ENCODING

TABLE 3
NR4SD⁺ Encoding

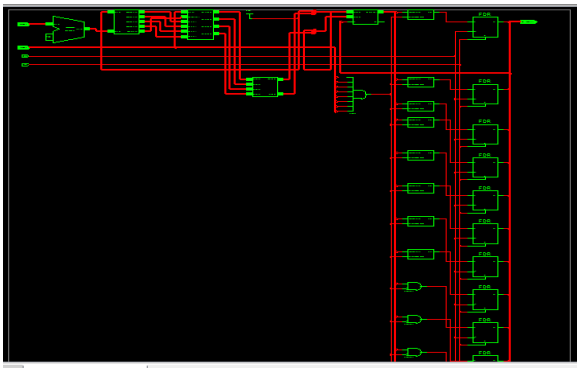
2's complement			NR4SD⁺ *form*			*Digit*	NR4SD⁺ *Encoding*		
b_{2j+1}	b_{2j}	c_{2j}	c_{2j+2}	n^+_{2j+1}	n^-_{2j}	$\mathbf{b}_j^{NR+}$	one_j^+	one_j^-	two_j^+
0	0	0	0	0	0	**0**	0	0	0
0	0	1	0	1	1	**+1**	1	0	0
0	1	0	0	1	1	**+1**	1	0	0
0	1	1	0	1	0	**+2**	0	0	1
1	0	0	0	1	0	**+2**	0	0	1
1	0	1	1	0	1	**-1**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

Table 3 NR4SD⁺ENCODING

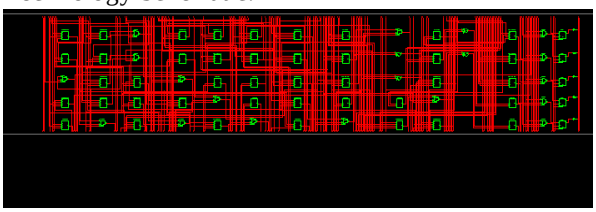
IV. RESULTS



RTL schematic:



Technology Schematic:



Timing Report:

Speed Grade: -5

Minimum period: No path found

Minimum input arrival time before clock: 19.392ns

Maximum output required time after clock: 4.040ns

Maximum combinational path delay: 19.392ns

Design Summary:

hklhlkh Project Status (09/12/2016 - 13:17:30)			
Project File:	hklhlkh.ise	**Current State:**	Synthesized
Module Name:	mul_NR4SD	**• Errors:**	No Errors
Target Device:	xc3s500e-5fg320	**• Warnings:**	26 Warnings
Product Version:	ISE 10.1 - Foundation Simulator	**• Routing Results:**	
Design Goal:	Balanced	**• Timing Constraints:**	
Design Strategy:	Xilinx Default (unlocked)	**• Final Timing Score:**	

hklhlkh Partition Summary [-]
No partition information was found.

Device Utilization Summary (estimated values) [-]			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	105	4656	2%
Number of Slice Flip Flops	8	9312	0%
Number of 4 input LUTs	181	9312	1%
Number of bonded IOBs	34	232	14%
Number of GCLKs	1	24	4%

VI. CONCLUSION

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 power consumption compared to the conventional MB multiplier.

REFERENCE

[1] G. W. Reitwiesner, "Binary arithmetic," Advances in Computers,vol. 1, pp. 231–308, 1960.

[2] K. K. Parhi, VLSI Digital Signal Processing Systems: Design andImplementation. John Wiley & Sons, 2007.

[3] K. Yong-Eun, C. Kyung-Ju, J.-G. Chung, and X. Huang, "Csdbased 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, Department of Computer Science, University of California, Los Angeles, CA, 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 Int. Symp. on Integrated Circuits (ISIC 2007), 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 IEEEInt. Symp. on Circuits and Syst. (ISCAS 2009), 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 US 6 108 633, Aug., 2000.

[13] 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.

[14] G. Pastuszak, "A high-performance architecture of the doublemode 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. (VLSI) Syst., vol. 11, no. 2, pp. 244–253, Apr.2003.

[16] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "A 16-channel lowpower 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 low-power 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: Morgan Kaufmann, 2004, pp. 181–245.

[19] N. Weste and D. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th ed. USA: Addison-Wesley Publishing Company, 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 memristor-based rram cross-point structures," in Design, Automation Test in Europe Conf. Exhibition (DATE), Mar. 2011,pp. 1–6.

[22] [14] C. K. Chen et al., "A low power independent component analysis processor in 90 nm CMOS technology for portable EEG signal

processing systems," in Proc. IEEE Int. Symp. Circuits Syst., May 15–18, 2011, pp. 801–804.

[23] H. Qi and X. Wang, "Comparative study of VLSI solutions to independent component analysis," IEEE Trans. Ind. Electron., vol. 54, no. 1, pp. 548–558, Feb. 2007.