Arithmetic Circuits for Analog Digits

ISMVL'99 - 29th International Symposium on Multiple-Valued Logic

Aryan Saed*, Majid Ahmadi, Graham A. Jullien
*Nortel Networks, Nepean, Ontario, Canada
saed@nortel.ca
VLSI Research Group, University of Windsor,
Windsor, Ontario, Canada

Abstract

The Overlap Resolution Number System (ORNS) employs bit level residue arithmetic, and opens up a powerful approach to digital computing with analog processing elements. This new redundant representation of signals, with Continuous Valued Digits, opens up new methods for binary arithmetic and digital signal processing. The number system is based on analog residues as opposed to binary or multiple-valued digit levels. Importantly, arithmetic in ORNS is tolerant to VLSI circuit tolerances. This allows simple elementary analog circuits to be employed, targeted at digital accuracy.

1. Introduction

This paper presents analog CMOS circuit techniques for elementary digit manipulations. The focus lies on binary multiplier structures that internally employ Continuous Valued Digits. The principles of arithmetic in the Overlap Resolution Number System differ significantly from the familiar concepts of arithmetic with binary or multiple-valued digits: logic gates, or respectively discrete level analog circuits, are replaced by continuous valued digit manipulation circuits.

The 1847 essay of George Boole, *The Mathematical Analysis of Logic: Being an Essay Towards a Calculus of Deductive Reasoning*, and Claude Shannon's 1938 publication, *A Symbolic Analysis of Relay and Switching Circuits*, have laid the fundamentals for computers composed of multitudes of logic gates, and for number systems that employ binary symbols assuming one of two values, *high* and *low*.

In Multiple-Valued Logic (MVL), the number of discrete signal values or logic states extends beyond two. Arithmetic units implemented with MVL achieve more efficient use of silicon resources and circuit interconnections, and current advances serve the design of fast and area efficient multipliers and memory, e.g. [1]. MVL exploits the potential accuracy of silicon circuits by relying on more than just the two states *on* and *off*. Modern circuits for MVL arithmetic units successfully employ analog circuits for elementary digit operations, in replacement of digital logic gates.

This paper discusses an alternative to binary and MVL multiplier structures. In the following section we review Continuous Valued Digits (CVD's) and we compare their properties with the familiar discrete valued digits in a positional number system (PNS). We will see that already binary CVD's are capable of exploiting silicon accuracy in arithmetic structures.

Arithmetic rules in ORNS are reviewed, and their impact on hardware VLSI architectures is discussed with an emphasis on the implementation of binary digital multiplication functions. The paper proceeds with introduction of analog CMOS circuits for CVD's. These circuits comprise the leaf cells in hardware architectures for arithmetic building blocks.

2. Radix-B Overlap Resolution

Given a real x bound by X as $|x| < X$, we shall represent it by a set of CVD's r_n, with index $n = K \ldots L$ and $K \leq L$. We recall from [2] and [3] that there exist two methods to calculate CVD's, each arriving at the same result. The first method involves a cascaded approach, whereby we start with the Most Significant Digit (MSD) r_L, and compute it as $r_L = B \cdot \frac{x}{X}$. The positive integer B is the radix, and we shall further assume $B \geq 2$. For $B = 2$ we have the important case of binary ORNS. Further digits $n < L$ are calculated by the *cascade rule*:

$$r_n = (r_{n+1} - \tilde{a}_{n+1}) \cdot B \tag{1}$$

with $\tilde{a}_n = \lfloor r_n \rfloor$, whereby $\tilde{a}_n$ is an integer associated with the CVD r_n. The operator $\lfloor . \rfloor$ denotes flooring towards zero, such that $|\tilde{a}_n| < r_n$. As a result we have $|r_n| < B$ and $|\tilde{a}_n| \leq B-1$. The CVD's r_n need not be an integer. We choose to select L such that $B^{L+1} \geq |X|$. A rule for selecting K will follow in the next section.

The second method involves the signed modulo operation $a \mathrm{mod} B = a - B\lfloor a/B \rfloor$, which we define for integer as well as real values of a for both signs. The ORNS *basic expression* is:

$$r_n = \left(\frac{x}{X} \cdot B^{L-n+1}\right) \mathrm{mod} B \tag{2}$$

Both methods may also be used to compute digits with index $n > L$. We shall see in following sections that such *excessively evolved digits* (EED's) serve arithmetic with CVD's. EED's are equivalently computed by $r_{n>L} = r_{n-1}/B$ conform the cascade rule, or by $r_{n \geq L} = B^{L-n+1} \cdot x/X$ conform the basic expression. The proof is simple, and follows from properties of the modulo operation. We conclude that CVD's in ORNS are of the general form $(aB^{-k}) \mathrm{mod} B$, with real a.

An ORNS number is written as $N_x = (r_L, \ldots, r_0 | r_{-1}, \ldots r_K)$, with a radix 'bar' between r_0 and r_{-1}. We typically use a decimal notation for the value of a CVD, and hence a vertical bar shall be used for the radix point of the ORNS number. In this paper we shall limit our discussions to non-negative values of x. Nevertheless, all digits r_n follow the sign of x, and the sign inversion $r_n(-x) = -r_n(x)$ holds.

A remarkable characteristic of a CVD is, as the name implies, that a digit value requires a form of continuous symbolization. Hence, we may also term CVD's as *analog digits*, if we have an electronic implementation in mind. If a linear electronic medium of our choice, for instance a current, charge or voltage, ranges from 0 to $\pm Q$ units, then each CVD is matched proportionately to an electronic quantity q_n by $q_n = r_n \cdot Q/B$.

Example 1: A value $x \geq 0$, limited by $X = 100$, shall be represented by CVD's in the range $0\mu A$ to $50\mu A$. We select two radix values, $B = 10$ for decimal ORNS, and $B = 2$ for binary ORNS. To satisfy $X \leq B^{L+1}$ we select $L = 1$ for the decimal case, and $L = 6$ for the binary case. For both we select $K = -1$. The CVD's for $x = 58.742$ are presented in Table 1. The decimal digit r_2 is an EED. We observe, that $\tilde{a}_n$ are the PNS digits of $(xB^{L+1})/X$.

3. Redundancy

The value x shall be termed the *root* of the number N_x. Clearly, the root is retrieved from N_x by the MSD alone, without any error: $x = r_L \cdot X/B$. N_x is not an approximate representation, since r_L and x may assume any real value. However, there is a practical limitation. With an increasing precision of x, it becomes difficult, if not eventually impossible, to maintain that precision in r_L when a circuit implementation with q_L is envisioned. We therefore need to discuss the consequences of limited precision digits in N_x.

In a positional number system the integers $\tilde{a}_n$ are stored as multiple-valued digits, and arithmetic is performed with these discrete valued digits, ranging $0 \ldots (B-1)$ for $x \geq 0$.

Table 1: ORNS Example for *x*=58.742

	Decimal $(B=10)$			**Binary** $(B=2)$		
n	r_n	$\tilde{a}_n$	$q_n/(\mu A)$	r_n	$\tilde{a}_n$	$q_n/(\mu A)$
6	-		-	1.17484	1	29.371
5	-		-	0.34968	0	8.742
4	-		-	0.69936	0	17.484
3	-		-	1.39672	1	34.918
2	0.58742	0	2.9371	0.79744	0	19.936
1	5.87420	5	29.3710	1.59488	1	39.872
0	8.74200	8	43.7100	1.18796	1	29.699
-1	7.42000	7	37.1000	0.37952	0	9.488

An ORNS number N_x stores the digits r_n and discards the integers $\tilde{a}_n$. Given two neighbouring digits r_n and r_{n-1}, we are able to retrieve $\tilde{a}_n$ by $\tilde{a}_n = \lfloor r_n \rfloor$, and recalculate r_n as $r_n = \tilde{a}_n + r_{n-1}/B$. This is a trivial result from (1), and equally trivial we can retrieve $\tilde{a}_n$ by $\tilde{a}_n = r_n - r_{n-1}/B$. Given an errored digit pair r'_n and r'_{n+1}, we restore r'_n to $r''_n = \tilde{a}'_n + r'_{n-1}/B$, whereby we calculate the associated integer as $\tilde{a}'_n = \lceil r'_n - r'_{n-1}/B \rfloor$. The symbol $\lceil . \rfloor$ denotes rounding to the nearest integer. With $r'_n = r_n + \varepsilon_n$ and $r'_{n-1} = r_{n-1} + \varepsilon_{n-1}$, we find $\tilde{a}'_n = \lceil \tilde{a}_n + \varepsilon_n - \varepsilon_{n-1}/B \rfloor$. Hence, under the condition

$$|\varepsilon_n - \varepsilon_{n-1}/B| < 1/2 \quad (3)$$

we have $\tilde{a}'_n = \tilde{a}_n$. Provided that $\tilde{a}'_n = \tilde{a}_n$, we find that $r''_n = r_n + \varepsilon'_n$ contains the error $\varepsilon'_n = \varepsilon_{n-1}/B$. The success of obtaining $\tilde{a}'_n = \tilde{a}_n$ conveniently does not depend on the value of r_n. We show in [4] that the condition in (3) implies

$$|\varepsilon_n| < B/(2(B+1)) \quad (4)$$

if we assume equal error statistics among all digits. In comparison, the error condition for Multiple-Valued digits depends on the digit level distances, which equals one regardless of the radix. Hence, the error of discrete valued digits (DVD's) must adhere to $|\varepsilon_n| < 1/2$.

The important *relative* implementation error condition for CVD's and DVD's shall be defined as $|\varepsilon_n|/B < \hat{\vartheta}$ and various values of $\hat{\vartheta}$ are reported in Table 2. We recognize the familiar noise margin of 50% for binary systems, and clearly that noise tolerance is inferior for CVD's. It is therefore not particularly envisioned that ORNS be employed for storage of x. The important gain of CVD's results from arithmetic properties which we shall now discuss.

4. Addition and Multiplication

The CVD's $r_n(x+y)$ of a sum $x+y$ are simply obtained digit-wise and there is no carry or other interaction required between neighbouring digits, [3], [6]:

$$r_n(x+y) = (r_n(x) + r_n(y)) \bmod B \quad (5)$$

Table 2: DVD and CVD Error Threshold Examples

Radix B	2	4	8	10	100
$\hat{\vartheta}$ DVD	50.0%	16.7%	7.14%	5.56%	0.51%
$\hat{\vartheta}$ CVD	16.7%	10.0%	5.56%	4.55%	0.50%

We recall that we have limited our discussion to $x, y \geq 0$. The CVD's of a product $\lambda \cdot x$ with integer λ are:

$$r_n(\lambda x) = (\lambda \cdot r_n(x)) \bmod B \quad (6)$$

In the special case $\lambda = B^k$, we have $r_n(B^k \cdot x) = r_{n-k}(x)$, and if $y = \sum_{\forall k} \lambda_k B^k$, then

$$r_n(y \cdot x) = \left(\sum_{\forall k} \lambda_k \cdot r_{n-k}(x) \right) \bmod B \quad (7)$$

This is the equivalent of the familiar shift-and-add principle for Radix-B multiplication. In binary ORNS the values of λ_k are limited to $\{0, 1\}$, and hence they serve as an on-off switch for summing analog voltages, currents or charges $r_{n-k}(x)$. We intend to represent the multiplier y in binary, with digits λ_k.

The rules for addition and multiplication not only hold for the perfect CVD's r_n, but also for imperfect values r'_n. Of course, CVD imperfections of operands in addition and multiplication propagate into the result CVD's, but it has been demonstrated here, as well as in [2] and [4], that ORNS is robust against such errors.

Example 2: Given $x = 31.89$ and $y = 3.54$, they shall be summed in decimal ORNS with $X = 100$. Their ORNS numbers with approximate CVD's are $N'_x = (3.2, 1.9|8.9)$ and $N'_y = (0.4, 3.5|5.4)$. The perfect sum of errored digits is $(3.6, 5.4|4.3)$, yet we shall consider the *errored* sum with *errored* digits $N'_{x+y} = (3.61, 5.39|4.31)$. We find $\tilde{a}'_0 = \lceil 5.39 - 4.31/10 \rceil = 5$ and $r''_0 = 5 + 4.31/10 = 5.431$, and further $\tilde{a}'_1 = 3$ and $r''_1 = 3.5431$. We conclude, with a minor error, that $x + y = 35.431$. The error is reduced by increasing the number if digits.

5. ORNS-Digital Interface

In [5] we presented a method for generating CVD's directly from binary bits, and we discussed an ORNS based architecture for a binary multiplier. In the remaining sections of this paper we will employ our combined knowledge of the rule for addition and the error tolerance of CVD's, to develop CMOS current-mode analog circuits for Radix-2 (binary) ORNS.

Considering the novelty of this topic, we will review the generation of CVD's from bits, and the retrieval of bits from CVD's. This will allow us to use the arithmetic properties of CVD's within a binary multiplier. This encapsulation of ORNS within a digital cast is one application of the number system. It allows the development of arithmetic standard cells in a digital circuit library.

A second application of ORNS lies in the immediate representation of a signal value x by CVD's, without the intervention of a binary number system. In essence this is an alternative to quantization and Digitalization of a signal, whereby we are still able to effectively distribute x over a set of digits. In this paper are concerned with the first application, in particular the development of a binary multiplier.

A number x is represented by a weighted sum over binary bits a_n

$$x = \sum_{n=K'}^{L'} a_n \cdot 2^n \quad (8)$$

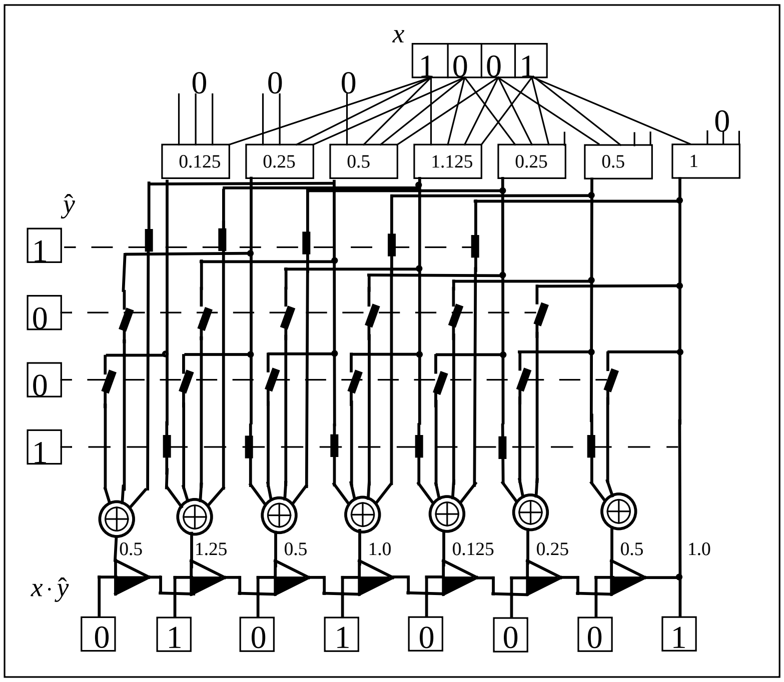

Figure 1: [1001]x[1001] Multiplication Example

Considering the range of x in Eqn. (8), we select X in ORNS such that $X = 2^{L'+1} = 2^{L+1}$. We require $L' \ge K'$, but allow $L', K' \le 0$. Inserting Eqn. (8) in Eqn. (2), and selecting $B = 2$ we find the CVD's of x directly from a weighted sum of the bits:

$$r_{L-j} = \sum_{n=K'}^{L'-j} a_n \cdot 2^{n+j-L'} \qquad (9)$$

With a modification of the radix in Eqn. (8) we can obtain CVD's from non-binary PNS digits by selecting B accordingly. We shall exploit the error tolerance of Eqn. (4), and limit the scope of the summation in Eqn. (9) to only a few (ψ) bits per CVD:

$$r'_{L-j} = \sum_{n=L'-j-\psi+1}^{L'-j} a_n \cdot 2^{n+j-L'} \qquad (10)$$

The relative digit error $\hat{\varepsilon}_n = \varepsilon_n/2$ results from the truncation, and it is bound by $|\hat{\varepsilon}_n| < 2^{-\psi}$. The bits a_n of x equal the associated integers $\tilde{a}_n$. A typical value of $\psi = 4$ implies a 4-bit digital-to-analog (DA) conversion per CVD, regardless of the binary word length of x. Bits are calculated from CVD's by

$$\tilde{a}'_n = \left\lceil r'_n - \frac{r''_{n-1}}{B} \right\rceil \mathrm{mod}^+ B \qquad (11)$$

where $\lceil . \rceil$ denotes rounding to the nearest integer. We define $a \mathrm{mod}^+ B = a + I' \cdot B$ with integer I' such that $0 \le a + I' \cdot B < B$ for real a. Digit r'_n is now restored by

$$r''_n = \left(\frac{r''_{n-1}}{B} + \tilde{a}'_n\right) \mathrm{mod}^+ B \qquad (12)$$

We obtain $\tilde{a}'_n$ and r''_n from least to most significant, in sequence, utilising the *corrected* neighbour r''_{n-1}. Of course, the least significant digit remains uncorrected: $r''_K = r'_K$.

6. Binary Multiplication

In Figure 1 we present a 4bit-by-4bit multiplication example. It is noted that the applied word-length of 4 bits is not related to $\psi = 4$. There are seven DA

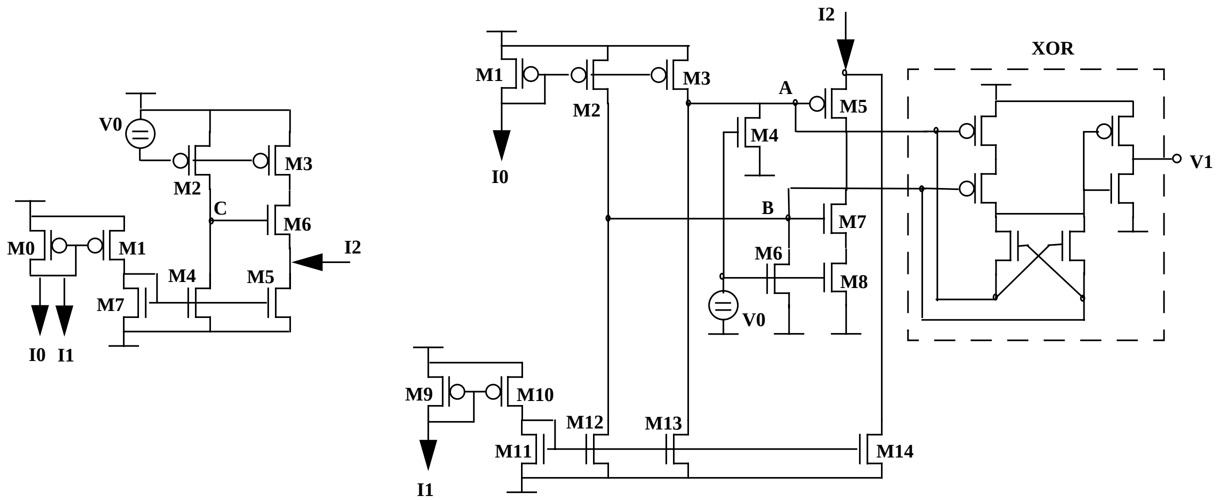

Figure 2: CMOS analog current-mode circuits for a CVD full adder (left) and CVD correction with binary conversion (right)

converters in the top, producing digits $0 \le r_n < 2$. The four rows of switches are driven by λ_k 's. The summers at the bottom are 4-input modulo summers, and the triangles are symbols for the digit correction and bit retrieval operations of Eqn. (11) and Eqn. (12). Digit r'_n is entered at the top, and r''_{n-1} at the right. The output r''_n is obtained from the bottom, and $\tilde{a}'_n$ from the left.

The size of the multiplier has been selected such, that it illustrates the principles of the architecture. The number of columns can easily be extended to ν bits, with typical values of 16, 32 and 64. A ν-by-ν multiplier of such dimensions will consist of μ-bit layers, each ν wide. The value of μ is determined by the analog accuracy of the employed circuits. In our example we employed 4-bit analog accuracy $(\mu = 4)$, and hence a 32-by-32 bit multiplier consists of eight 4-bit layers.

Layers of switches are separated by layers of summers and correction units. The latter are depicted by triangles in Figure 1, and in essence they refresh the digit values. The DA converters are only needed once. Their word-length is determined by the targeted analog accuracy. It is only required to provide a DA resolution that is intended to be supported by the accuracy of the applied circuits.

7. CMOS Current-Mode Analog Circuits

We will now introduce two characteristic circuits for radix-2 (binary ORNS) analog digits: the digit correction circuit that performs the operations of Eqn. (11) and Eqn. (12), and a 2-input modulo summer, which is a building block for a 4-input summer. In Figure 2 we present the two circuits. Of course, eventually the 4-input summer is best designed directly.

The left circuit performs the rule for addition in Eqn. (5). Since the sum within the modulo operation is limited to $0 \le (r_n(x) + r_n(y)) < 2B$, we can write

$$r_n(x+y) = r_n(x) + r_n(y) - \kappa B \qquad (13)$$

whereby $\kappa \in \{0, 1\}$. The draining currents *I0* and *I1* represent the summands $r_n(x)$ and $r_n(y)$, and the modulo sum $r_n(x+y)$ is drained as *I2*. The voltage at node *C* represents κ. The sum $r_n(x) + r_n(y)$ at *M4* is compared with a reference current at *M2*, representing the digit value B. We recall that Q corresponds to B.

The comparison results in ‘greater’ or ‘less’, leading to *C* ‘low’ or ‘high’ respectively. In the case of ‘less’, the sum is simply presented at the output by *M5* $(\kappa = 0)$. In the case of ‘greater’, *M6* is turned on, and the value B at *M3* is subtracted from the sum $(\kappa = 1)$.

The right circuit in Figure 2 performs the digit correction and delivers a binary bit for each CVD. Digit r'_n is drained as *I0*, and r''_{n-1} as *I1*. The output r''_n is drained as *I2*. The corresponding bit $\tilde{a}'_n$ is provided as a voltage *V1*.

The modulo rounding operation in Eqn. (11) is performed by comparing the difference $r'_n - r''_{n-1} / B$ to the digit values $1/2$ and $2/3$. These values corresponding to quantities $Q/4$ and $3Q/4$. If the difference lies between the two, we conclude $\tilde{a}'_n = 1$, otherwise $\tilde{a}'_n = 0$. The current through *M4* equals $Q/4$, and the current through *M6* equals $3Q/4$. The comparisons are separately evaluated at nodes *A* and *B*, and the *XOR* delivers the bit as a voltage *V1*. In accordance with Eqn. (12), for $\tilde{a}'_n = 1$ the cascade of switches *M5* and *M7* adds a current of $Q/2$ at *M18* to r''_{n-1}/B at *M14*.

The presented circuits have been designed and simulated in a $0.8\mu m$ CMOS environment, with $3.3V$ supply voltage and $Q = 50\mu A$. The 4-by-4 multiplier of Figure 1 has been successfully designed, based on these circuits. The optimization for speed remains a topic of future research.

The reference voltage *V0* in both circuits may be fixed, or adjusted to compensate for temperature variations. The accuracy of the current mirrors chiefly determines the quality of the circuit. With increased accuracy the accumulated digit error is reduced, and the column size in a multiplier may be increased, resulting in fewer layers. Simpler circuits with smaller transistors introduce more error, and hence the architectural complexity increases. The design of an ORNS based arithmetic library cell is therefore directed by such trade-off, whereby speed, power consumption, area and switching noise are components in the cost function.

8. Conclusions

This paper has discussed the role of analog digits in the Overlap Resolution Number System, and it has reviewed arithmetic rules and interfaces between multiple-valued digits and continuous valued digits, with a particular focus on the binary case. We have learned that a binary multiplier of any dimension can be assembled from layers of switches and digit refreshment circuits. Analog circuit tolerance is not a prohibitive limitation for large multipliers.

ORNS exploits VLSI circuit accuracy without resorting to a radix higher than $B = 2$. Nevertheless, the presented theories allow for instance quaternary ORNS to interface with 4-level MVL.

CMOS analog current mode circuits for ORNS are closely related to current mode circuits in MVL. They consist of current comparators and current mirrors.

9. References

[1] Butler, Jon T. ed., Multiple-Valued logic in VLSI, IEEE Computer Society Press, 1991

[2] Saed, A., M. Ahmadi, G.A. Jullien, W.C. Miller, “Overlap Resolution: Continuous Valued Digits for Hybrid Architectures”, 40th Midwest Symposium on Circuits and Systems, August 1997, Sacramento, California.

[3] Saed, A., M. Ahmadi, G.A. Jullien, W.C. Miller, “Overlap Resolution: Arithmetic with Continuous Valued Digits in Hybrid Architectures”, Thirty First Annual Asilomar Conference on Signals, Systems, and Computers, November 1997, Pacific Grove, California.

[4] Saed, A., M. Ahmadi, G.A. Jullien, W.C. Miller, “Circuit Tolerances and Word Lengths in Overlap Resolution”, The 1998 IEEE International Symposium on Circuits and Systems (ISCAS), June 1998, Monterey, California.

[5] Saed, A., M. Ahmadi, G.A. Jullien, “Analog Digits: Bit Level Redundancy in a Binary Multiplier”, Thirty Second Annual Asilomar Conference on Signals, Systems, and Computers, November 1998, Pacific Grove, California.

[6] Saed, A., M. Ahmadi, G.A. Jullien, “Arithmetic with Signed Analog Digits”, to appear in the proceedings of the 14th IEEE Symposium on Computer Arithmetic, April 1999, Adelaide Australia.