Arithmetic with Signed Analog Digits

Aryan Saed[1], Majid Ahmadi[2], Graham A. Jullien[2]
[1]Nortel Semiconductors, Nepean, Ontario, Canada
saed@nortel.ca
[2]VLSI Research Group, University of Windsor,
Windsor, Ontario, Canada

Abstract

This paper presents mathematical foundtions of the Overlap Resolution Number System (ORNS) which employs signed Continuous Valued Digits (CVD's). ORNS is a redundant Number System employing residue arithmetic. In contrast to the implementation of arithmetic by binary or multiple-valued logic circuits, arithmetic operations in this novel number system are performed by analog digit manipulation circuitry. The redundancy in an ensemble of Continuous Valued Digits that comprises a number, provides tolerance to implementation imprecisions. Processing with these analog digits implies carry-free arithmetic structures with systematic redundancy.

1. Introduction

The concept of numbers and measures stems from farming, astrology, trade and navigation, and in all the history of mathematics, numbers are essentially written as combinations of symbols [1], [2]. A digit is such a symbol. It is a discrete symbol and it requires a discrete notation. A number is represented by a set of such discrete symbols and possibly some ancillary operators, which in turn are again symbols. The reason for selecting discrete symbols lies in the convenience of depiction. It is easier to recognise the shape of a symbol, or to count the number of symbols, than it is to recognise the size of a symbol. Early notations therefore involved for instance countable marks on wood and bones, or consisted of pebble arrangements.

The Positional Number System (PNS) is an example of a number system with such digits, [3]. In PNS, a number is represented by a set of positive integer digits and the location of a radix point. The value of the number is determined by the values of the digits, the value of the radix, and the convention of performing a weighted sum over the digits, using powers of the radix. Alternative systems employ non-integer radix values or multiple radices, or negative values for the digits, [4], or they employ sophisticated methods for arriving from a set of digits to a value, other than through a weighted sum, as for example by the Chinese Remainder Theorem in the Residue Number System [5].

Today's semiconductor circuits for binary arithmetic owe much to the insight of George Boole in his 1847 essay *The Mathematical Analysis of Logic: Being an Essay Towards a Calculus of Deductive Reasoning*, and to Claude Shannon's 1938 publication *A Symbolic Analysis of Relay and Switching Circuits*. These works 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 extends beyond two. Arithmetic units implemented with MVL achieve more efficient use of silicon resources and circuit interconnections, and MVL is not limited to PNS representations. The early commercial and research applications were focused on memory elements, while current advances serve the design of fast and area efficient multipliers [6]. An MVL number representation requires fewer digits (or fewer non-zero digits) for a given number range, compared to binary. This reduces the number of devices involved in storage and arithmetic. With fewer digits, carry propagation chains are shorter and they can even be limited to one neighbouring digit when signed digits are employed. MVL exploits the potential accuracy of silicon circuits by relying on more than just the two states *on* and *off.*

This paper discusses an alternative to MVL. In the following section we introduce Continuous Valued Digits (CVD's) and their role in exploiting the potential accuracy of circuits. Then we discuss the reliability of CVD's in an imprecise, faulty, environment, after which we define arithmetic rules for addition and multiplication within the proposed Overlap Resolution Number System (ORNS).

2. Continuous Valued Digits

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 $K \le L$. We propose 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 $B \ge 2$. Further digits are calculated by the *cascade rule*:

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

whereby $\tilde{a}_n = \lfloor r_n \rfloor$ 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 $|\tilde{a}_n| \le B - 1$. The CVD's r_n need not be an integer. We choose to select L such that $B^{L+1} \ge |X|$. A rule for selecting K will follow in the next section.

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

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

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 $r_{n \ge L} = B^{L-n+1} \cdot x/X$ conform the basic expression.

The basic expression allows the calculation of a digit r_m from any more significant digit r_n by

$$r_m = (r_n \cdot B^{n-m}) \bmod B \qquad n \ge m \qquad (3)$$

The proof is simple, and follows from properties of the modulo operation. Using (), a digit r_{L-i} with $i \le L$ equals $r_{L-i} = (r_L \cdot B^i) \bmod B$, which can be written as $r_{L-i} = r_L \cdot B^i + I_0 \cdot B$ with a determinable integer I_0. With $r_L = r_{L-i}/B^i - I_0 \cdot B^{1-i}$ we find for another digit r_{L-k} with $k \ge 0$: $r_{L-k} = (r_{L-i} \cdot B^{k-i} - I_0 \cdot B^{1+k-i}) \bmod B$. If $k \ge i$, then the term $-I_0 \cdot B^{1+k-i}$ is an integer multiple of B for any I_0. Moreover, the sign of that term equals the sign of $r_{L-i} \cdot B^{k-i}$, and hence $r_{L-k} = (r_{L-i} \cdot B^{k-i}) \bmod B$. With the substitutions $n = L - i$ and $m = L - k$ the proof is complete.

The relationship between two neighbouring digits r_n and r_{n+1} becomes $r_n = (r_{n+1} \cdot B) \bmod B$, from which we arrive at the cascade rule () with the above definition of the modulo operation. We conclude that CVD's in ORNS are of the general form $(aB^{-k}) \bmod B$, with real a.

An ORNS number is written as $N_x = (r_L, \ldots, r_0 \,|\, r_{-1}, \ldots r_K)$, with a radix point 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. If $X = B^{L+1}$, then digits $r_{-1} \ldots r_K$ represent the fractional part of x with respect to the radix B. The remaining digits represent combined integer and fractional parts. 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 $(xB^{L+1})/X$.

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

To the knowledge of the authors, to date, no concept of analog digits exists as part of a number system. A similar principle is in use with mechanical devices that employ a gear box and rotating indicators to represent a continuous value [7], [8]. The modular and continuous nature of Continuous Valued Digits is present in kilo-Watt-hour meters, gas volume meters, aeronautical altimeters and the analog watch. An example follows in the next section.

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. With an increasing precision of x, it becomes difficult, if not eventually impossible, to maintain that precision in r_L when a practical circuit implementation of q_L is envisioned. We therefore need to discuss the consequences of limited precision digits in N_x.

The set of digits in an ORNS number comprise, in a sense, a systematic code. The term *systematic code* stems from the field of error control coding, and indicates a redundant representation of a *k*-dimensional point in a discrete *n*-dimensional space, $n > k$, by concatination of some form of check and correction information. In a systematic code, the information is separated from the redundancy. Considering the MSD r_L from this perspective, it contains the information, and all the other digits are separate and redundant entities. This thought can be carried over to any pair or sub-set of digits. If we extend the concept of the discrete *n*-dimensional space to an $(L - K)$-dimensional space, then per () a digit together with any one of its neighbours, more or lesser significant, immediate or far, forms a systematic code. The digit contains the information and a neighbour contains some redundant check information for that digit.

Let us consider errored digits $r'_n = r_n + \varepsilon_n$. For a non-zero error ε_L, the root of the number N'_x with errored digits no longer equals x, but it equals

$$\hat{x} = x + \varepsilon_L \cdot X/B \qquad (4)$$

and thus N'_x is an errored representation of x. We now present a method for reducing the difference between the root x of N_x and the root $\hat{x}$ of N'_x. The method, termed *reverse evolution*, utilises the redundant information in the digits $r_{L-1\ldots K}$. We will explain the proposed method and discuss its merits.

A 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 (), 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 propose to restore r'_n to $r''_n = \tilde{a}'_n + r'_{n-1}/B$, whereby we calculate the associated integer as $\tilde{a}'_n = \lceil r'_n - r'_{n-1}/B \rceil$, whereby $\lceil . \rceil$ 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 $\tilde{a}'_n = \lceil \tilde{a}_n + \varepsilon_n - \varepsilon_{n-1}/B \rceil$. Hence, under the condition $|\varepsilon_n - \varepsilon_{n-1}/B| < 1/2$, 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 $\varepsilon'_n = \varepsilon_{n-1}/B$. If we alternatively employ $\tilde{a}'_n = \lfloor r'_n \rfloor$, then the success of obtaining $\tilde{a}'_n = \tilde{a}_n$ inconveniently depends on the value of r_n in addition to ε_n.

It is safe to assume that all digit errors have a substantially equal error probability density function (EPDF) $p(\varepsilon)$, considering that the analog digits of a number are implemented within the same circuit. In particular we assume $p(\varepsilon_n) = p(\varepsilon_{n-1})$, and then clearly

$p(\varepsilon'_n) = B \cdot p(\varepsilon_{n-1} \cdot B)$. As a result, the EPDF after restoration is shrunk towards the vertical axis, implying a reduction in error variance. It also implies a reduction of the maximum and minimum error values if, of course, these bounds are finite. In *reverse evolution* the digits are corrected pair-wise and in sequence from LSD to MSD, using $\tilde{a}'_n = \lceil r'_n - r''_{n-1}/B \rceil$ and $r''_n = \tilde{a}'_n + r''_{n-1}/B$. The LSD remains uncorrected, but it is used to restore r'_{K+1}. The last digit to be corrected is r'_L, and its error equals $\varepsilon'_L = \varepsilon_K/(B^{L-K})$. If the root $\hat{x}$ is obtained from the restored MSD, we find in comparison to (1) the significantly improved case

$$(x - \hat{x})/X = \varepsilon_K/(B^{L-K+1}) \qquad (5)$$

Considering an equal EPDF for all digits (11), the condition $|\varepsilon_n - \varepsilon_{n-1}/B| < 1/2$ translates to $|\varepsilon_n| < B/(2(B+1))$, which implies a generous analog digit tolerance of 17% for binary ORNS. The resulting relative error in for instance an 8 digit number is then $(x - \hat{x})/X = 2^{-7}17\%$.

We now have an understanding of the important role of the redundant digits $r_{L-1\ldots K}$. We shall proceed to modify the above method, in order to accommodate the type of digit errors that may occur during arithmetic operations that we propose in the next sections. The modification involves the necessary extension of the condition $|\varepsilon_n - \varepsilon_{n-1}/B| < 1/2$ to

$$\left|(\varepsilon_n)\mathrm{mod}^{+}B - \frac{(\varepsilon_{n-1})\mathrm{mod}^{+}B}{B}\right| < \frac{1}{2} \qquad (6)$$

A *positive modulo* operation $a\mathrm{mod}^{+}B$ shall be defined by $a\mathrm{mod}^{+}B = a + I' \cdot B$ with integer I' such that $0 \le a + I' \cdot B < B$, and a *negative modulo* operation $a\mathrm{mod}^{-}B$ shall be defined such that $-B < a + I' \cdot B \le 0$. Modulo rounding operations are defined as $\lceil a \rceil_B^{+} = \lceil a \rceil\mathrm{mod}^{+}B$ and $\lceil a \rceil_B^{-} = \lceil a \rceil\mathrm{mod}^{-}B$, whereby $\lceil . \rceil$ denotes rounding to the nearest integer. It can be shown that $a\mathrm{mod}^{-}B = -(a\mathrm{mod}^{+}B)$.

An error ε_n can always and uniquely be decomposed into two terms $\underline{\varepsilon}_n$ and $\hat{I}_\varepsilon B$, such that $\varepsilon_n = \underline{\varepsilon}_n + \hat{I}_\varepsilon B$ with $|\underline{\varepsilon}_n| < B/2$. Condition (3) now implies $|\underline{\varepsilon}_n| < B/(2(B+1))$, meaning that $\hat{I}_\varepsilon B$ is disregarded. To allow for this flexibility, $\tilde{a}'_n$ must be calculated as follows:

$$\tilde{a}'_n = \begin{cases} \left\lceil r'_n - \dfrac{r''_{n-1}}{B} \right\rceil_B^{+} & \hat{x} \ge 0 \\ \left\lceil r'_n - \dfrac{r''_{n-1}}{B} \right\rceil_B^{-} & \hat{x} < 0 \end{cases} \qquad (7)$$

Methods for determining the sign of $\hat{x}$ based on N'_x are not the topic of this paper, and we shall simply assume that the sign is known, for instance by restricting the root to $x \ge 0$. Digit r'_n is restored to

$$r''_n = \begin{cases} \left(\dfrac{r'_{n-1}}{B} + \tilde{a}'_n\right)\bmod^{+}B & \hat{x} \geq 0 \\ \left(\dfrac{r'_{n-1}}{B} + \tilde{a}'_n\right)\bmod^{-}B & \hat{x} < 0 \end{cases} \tag{8}$$

By substituting (4) in (5), we find $r''_n = (r_n + \underline{\varepsilon}_{n-1}/B)\bmod^{+/-}B$ if (3) is satisfied, and it follows that $(x - \hat{x})/X = \underline{\varepsilon}_K/(B^{L-K+1})$. The steps that lead to these equations are omitted for brevity.

> *Example 2:* In Figure 1 we give an example resembling a utilities meter with $B = 10$. It is read as $N'_x = (4, 9, 7.5|4.5)$ with $X = 10^4$. By applying reverse evolution with (4) and (5) we find $\tilde{a}'_0 = 7$ and hence $r''_0 = 74.5$. Then $\tilde{a}'_1 = 8$ and $r''_1 = 874.5$, and finally $\tilde{a}'_2 = 3$ and $r''_2 = 3874.5$, resulting in $\hat{x} = 3874.5$. An imprecision in the setting of a dial, and an inaccuracy in the reading of it, is the equivalent of generating and sensing voltages, currents and charges in analog circuits with limited accuracy. It is important to note, that the applied method *with* modulo operations tolerates the critical cases where a dial setting of for instance $r_n = 9.9$ is read as $r'_n = 0$. The error is $\varepsilon_n = -9.9$, yet fortunately $\underline{\varepsilon}_n = 0.1$ and $B/(2(B+1)) = 0.45$.

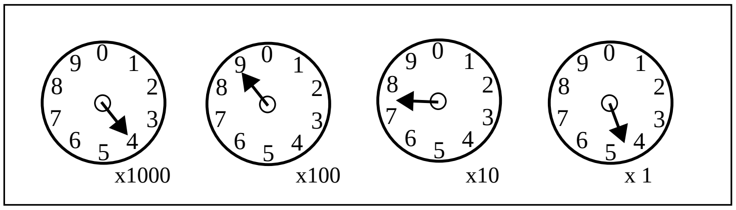

Figure 1: ORNS Example

Reverse evolution considers only pairs of digits, and converges to a solution $\hat{x}$ within $L - K$ steps. As such it is practical and fast. Other methods may be conceived that consider more digits in fewer steps, and as such utilise the systematic digit redundancy in a different way.

4. Addition

The CVD’s $r_n(x + y)$ of a sum $x + y$ are simply obtained by substituting x by $x + y$ in (). It is advantageous to develop a method for addition that employs the digits $r_n(x)$ and $r_n(y)$, without calculating $x + y$. In a circuit implementation, this means that it is not required to produce a signal $x + y$, and hence it is not necessary to build circuitry that support the target accuracy of $x + y$. In a manner similar to digital signal processing, a value x is converted to digits $r_n(x)$ in a circuit equivalent to an analog-to-digital converter, and arithmetic is performed with those digits in an imperfect, noisy, and non-linear environment. Output digits are then converted back to analog in an ORNS-to-analog converter.

Given two continuous valued (analog) residues $r = a \operatorname{mod}^{+} B$ and $s = b \operatorname{mod}^{+} B$ with real a and b, we find $(r+s) \operatorname{mod}^{+} B = (a+b) \operatorname{mod}^{+} B$. The proof is simple, since $r + s = a + I_a B + b + I_b B$ for given integers I_a and I_b, and $(a+b) \operatorname{mod}^{+} B = (a + b + IB) \operatorname{mod}^{+} B$ for any integer I. We specifically choose $I = I_a + I_b$. This property does not hold for the signed modulo operation $a \operatorname{mod} B$, yet since $(a+b) \operatorname{mod}^{+} B = (a+b) \operatorname{mod} B + iB$ with $i \in \{0, 1\}$, we have $(a \operatorname{mod} B + b \operatorname{mod} B) \operatorname{mod} B = (a+b) \operatorname{mod} B + kB$ with $k \in \{-1, 0, 1\}$. With the abbreviation $\xi = B^{L-n+1}/X$ we substitute $a = x\xi$ and $b = y\xi$ to find

$$r_n(x+y) = (r_n(x) + r_n(y)) \operatorname{mod} B - k_n B \quad (9)$$

Since $r_n(-y) = -r_n(y)$, subtraction is performed as $r_n(x-y) = (r_n(x) - r_n(y)) \operatorname{mod} B - k'_n B$.

Importantly, addition is digit-wise and there is no carry or other interaction required between neighbouring digits. If we dimension $|x+y| < X$, then the MSD is obtained from $r_L(x+y) = r_L(x) + r_L(y)$, without the ambiguity of k. If $|x| < X$ and $|y| < X$, then (6) with modulo operation applies to the MSD, and the EED is computed by $r_{L+1}(x+y) = r_{L+1}(x) + r_{L+1}(y)$. The EED serves as the ORNS equivalent of an arithmetic overflow digit in PNS.

Addition with (6) also holds for errored summand digits $r'_n(x)$ and $r'_n(y)$. Their errors, together with an imprecision in the summation and an incorrect modulo operation, are collected in an error $\varepsilon_n = \underline{\varepsilon}_n + \hat{I}_{\varepsilon} B$, and digit restoration is performed as discussed in the previous section. The integer k_n is of course determinable, yet it is simpler to include it in $\hat{I}_{\varepsilon}$.

Example 3: Given $x = 31.89$ and $y = -43.54$, 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 $N'_y = (-4.4, -3.5 | -5.4)$. The perfect sum of errored digits is $(-1.2, -1.6 | 3.5)$, yet we shall consider the *errored* sum with *errored* digits $N'_{x+y} = (-1.192, -1.622, 3.513)$, and after reverse evolution the sum $x + y$ is found from the MSD to be -11.6487. Notice how the LSD of the sum is affected by an error term $\hat{I}_{\varepsilon} B$.

5. Multiplication

The CVD's of a product $\lambda \cdot x$ with integer λ are:

$$r_n(\lambda x) = (\lambda \cdot r_n(x)) \operatorname{mod} B \quad (10)$$

The proof is as follows. Given the residue $r = a \bmod^{+} B$, it equals $r = a + I_1 B$ for a given integer I_1. Hence $\lambda r = \lambda a + \lambda I_1 B$. If λI_1 is integer, then $(\lambda r) \bmod^{+} B = (\lambda a) \bmod^{+} B$. The same holds for $r = a \bmod^{-} B$. If $r = a \bmod B$ and $\lambda r \geq 0$, then $(\lambda r) \bmod B = (\lambda r) \bmod^{+} B$, and if $\lambda r \leq 0$ then $(\lambda r) \bmod B = (\lambda r) \bmod^{-} B$. Also, if $r = a \bmod B$ then the sign of λr equals the sign of λa. Hence, $(\lambda r) \bmod B = (\lambda a) \bmod B$, for any real a and integer λ. We conclude that $(\lambda(a \bmod B)) \bmod B = (\lambda a) \bmod B$. In completion we substitute $a = x\xi$.

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

$$r_n(y \cdot x) = \left(\sum_{\forall k} \lambda_k \cdot r_{n-k}(x) \right) \bmod B \qquad (11)$$

This is the PNS equivalent of the familiar shift-and-add principle for multiplication, whereby x is now given in ORNS and y is given in PNS of the same radix. If $B = 2$, then y is given in binary, and λ_k serve as an on-off switch for summing analog voltages, currents or charges $r_{n-k}(x)$.

> *Example 4:* Given $x = 31.89$ with $B = 10$ and $X = 100$, we have $N_x = (3.189, 1.89, 8.9)$ with $L = 1$ and $K = -1$. For $y = 2.08$ we find $\lambda_0 = 2$ and $\lambda_{-2} = 8$. In the particular case $k = -2$ we require digits $r_{n+2}(x)$ for all $n = L \ldots K$. Therefore the EED's $r_2(x) = 0.3189$ and $r_3(x) = 0.03189$ are required in the calculation of the product CVD's. With (8) we find $r_1(x \cdot y) \approx 6.633$, $r_0(x \cdot y) \approx 6.331$, and $r_{-1}(x \cdot y) = 3.312$. After reverse evolution we correctly obtain the product with full precision: $x \cdot y = 6.63312$.

If, as with addition, errored digits $r'_n(x)$ are employed, then of course the product digits are also errored. The integers λ_k amplify the digit errors ε_n in (8), and it is beneficial to limit the range of values of λ_k. If $y \geq 0$, then the range $0 \leq \lambda_k \leq B - 1$ is required, but if we allow signed values of λ_k, then $|\lambda_k| \leq B/2$. The latter implies in general a smaller amplification of ε_n when $B \geq 3$, and it is preferred to represent y such that $|\lambda_k|$ are minimised. Multiplications with $\lambda = B^k$ involve shifts without amplification of ε_n, and these shifts are thus preferred above multiplications by λ_k. If we allow λ_k with mixed signs to represent y, then a multitude of representations of y exist, and the ones that minimise amplification of ε_n are preferred.

6. Conclusions

ORNS and the concept of CVD's were first introduced in [9], and basic methods for addition and array multiplication were given in

[10]. In [12] we present a scheme for efficient interfacing between ORNS and binary. This paper has provided formal foundations for addition and multiplication with *signed* CVD’s, and discusses an important method for reverse evolution that is robust against error terms $\underline{\varepsilon}_n$ as well as $\hat{I}_{\varepsilon}B$. The latter term may even occur in the perfect addition of errorless but signed CVD’s, and may also result as an inaccurate reading of, for instance, non-negative CVD’s when $r_n \approx B$ or equivalently $r_n \approx 0$.

References

[1] ___, New Encyclopaedia Britannica, 15th ed., 1986, Mathematics 13:441:1b, Arithmetic 14:75:2b

[2] Clawson, Calvin C., Mathematical Mysteries - The Beauty and Magic of Numbers, Plenum Press, 1996

[3] Knuth, Donald E., “The Art of Computer Programming”, Vol. 2 - Semi-numerical Algorithms, Addison-Wesley, 1969

[4] Koren, Israel, Yoram Maliniak, “On Classes of Positive, Negative, and Imaginary Radix Number Systems”, IEEE Transactions on Computers, Vol. c-30, No. 5, May 1981, pp. 312-317

[5] Soderstrand, M.A., W.K. Jenkins, G.A. Jullien, F.J. Taylor, Residue Number System Arithmetic: Modern Applications in Digital Signal Processing, IEEE Press, 1986, New York

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

[7] ___, Handbook for Electricity Metering, Ninth Edition, Edison Electric Institute, 1992

[8] Chironis, Nicholas P., ed., Mechanics, Linkages and Mechanical Controls, Mc Graw-Hill Book Company, 1965

[9] 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.

[10] 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.

[11] 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.

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