

Analog Representations in Digital Arithmetic: A Review

Behrooz Parhami

Dept. of Electrical and Computer Engr., Univ. of California, Santa Barbara, USA; parhami@ece.ucsb.edu

Abstract

After decades of being overshadowed by digital systems, analog computing is back in the spotlight, given its speed and energy economy and the proliferation of compute-intensive applications that do not require high precision. While pure analog processing has quite a few niche uses, hybrid digital/analog schemes satisfy the needs of many more application domains. In this paper, we review number representation schemes that are based on the analog encoding of numeric values in an otherwise digital arithmetic system, showing them to be closer than ever to practical realizability. We also point to several possible extensions and generalizations of the proposed schemes.

Keywords—Analog computing; Computer arithmetic; Continuous-valued digits; Hybrid digital/analog numbers; Multi-resolution number system; Residue number system.

1. Introduction

For several decades now, the world has moved continuously to replace analog data with digital data, given the latter's robustness and affinity with digital computing. From 1986 to 2007, the share of stored digital data increased from near-zero to more than 90% [1], increasing further to full dominance since then. There are now indications that in some cases, use of analog data and analog computing may be beneficial. As performance requirements and energy limitations drive us to embrace approximate computing [2], the low-precision drawback of analog circuits is no longer a disqualifying factor. Furthermore, in hybrid digital/analog representations, the speed and low power of analog computing [3] can be supplemented by the high precision and reliability of digital arithmetic, where needed. One can visualize multi-resolution number systems that allow us to combine fast low-precision with slower high-precision computation.

2. Continuous-Digit Positional Systems

The continuous-valued number system (CVNS) was motivated by the desire to build circuits with high immunity to noise [4], [5]. In CVNS, each digit has a weight that is a power of the radix r , as in ordinary positional representations, but the digit value, instead of being an integer in $[0, r-1]$, ranges continuously in $[0, r)$. A good analogy for understanding CVNS is the way multiple dials in an electric meter, depicted in Fig. 1, collectively represent a value such as 23,825 KWh. The first (most-significant) dial represents the entire amount, albeit in an approximate fashion, not just 20,000, as is the case in digital decimal representation.

The CVNS representation contains a form of natural redundancy, in that digit values in various positions exhibit some overlap. In the example of Fig. 1, the leftmost analog digit value ~ 2.38 tells us something about the next two analog digits ~ 0.383 and ~ 0.825 , which in turn embed imprecise forms of the digits to their right. This overlap is what makes the representation and associated arithmetic circuits more immune to noise. Additional noise immunity can be provided through the use of differential-style analog circuits.

CVNS can utilize signed, rather than unsigned, digits. In the radix-10 example of Fig. 1, the analog digits can be taken to be in $(-5, 5)$, instead of in $[0, 10)$. This kind of signed-digit representation, preferably one that is symmetric around 0, does not create any additional implementation challenges and may be preferred, particularly in implementing multiplication.

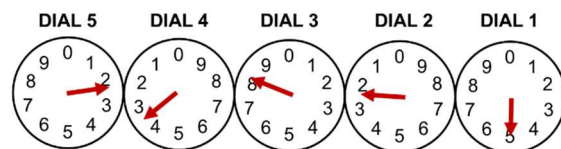


Fig. 1 The electric-meter analogy to CVNS [4].

3. RNS with Continuous Residues

Continuous-digit residue number system (CD-RNS) provides a theoretical framework for the way rats handle the positional information required for navigation [6]. Conventional RNS with discrete digits is based on methods that were known in ancient China, as evident from a 1500-year-old puzzle, attributed to Sun Tzu: Find an integer that has the remainders 2, 3, and 2 when divided by 7, 5, and 3, respectively? Modern RNS embodiments are used in DSP applications [7].

Like discrete RNS, CD-RNS is based on remainders or residues with respect to a set of moduli. As depicted in Fig. 2, the residues ϕ and ψ with respect to a and b , arising from $R = ia + \phi = jb + \psi$, collectively pinpoint the value R in the one-dimensional space, a value that has both the remainder ϕ with respect to the modulus a (heavy black dots in Fig. 2) and the remainder ψ with respect to the modulus b (hollow dots in Fig. 2).

CD-RNS offers a tradeoff between precision and range. If residues are known and processed with greater precision, they allow the representation range to be extended. Conversely, if only a limited dynamic range is needed, fewer residues and/or residues with lower precision will be needed, thus simplifying the circuitry, reducing power consumption, and increasing speed. An important result for a 2-modulus (μ_0, μ_1)-system [6], easily extensible to the case of m moduli, is that to provide a dynamic range of μ with a maximum residue error ε , we must have $\mu\varepsilon \leq \mu_0\mu_1$. This is the CD-RNS counterpart to an ordinary RNS with the moduli m_1 and m_2 having a dynamic range $M \leq m_1m_2$.

The study of CD-RNS has confirmed the hypothesis of computational neuroscientists on the use of modular number representation by rats and has also supplied a theoretical basis for their experimental verification of the scheme's feasibility and the adequacy of the dynamic range. As in ordinary RNS, selection of the moduli affects the dynamic range of CD-RNS. Intriguing open questions include how close the moduli in rat's navigation system are to optimal values and whether there is evidence of evolutionary refinement of the moduli values.

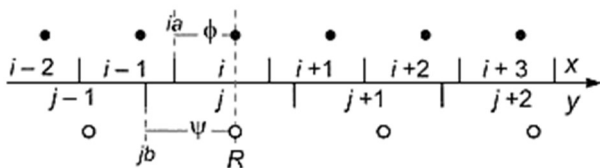


Fig. 2 Localization with two grids in 1D space [6].

4. Other Multi-Resolution Number Systems

There exist at least two other proposals for hybrid D/A number representation, which are newer and lesser-known. Both schemes are based on representing numbers in terms of timing and temporal separation of signals, rather than by means of signal direction or amplitude, as is the practice in today's digital systems.

In race logic [8], information is represented as timing delay, and computation results are derived based on relative propagation times of signals through circuits. With numbers represented as delays, the operations max, min, and add-constant are very easy to realize. It is too early to judge the usefulness of race logic in general, but in sample problem domains, such as DNA sequence alignment, practical benefits have been demonstrated over conventional CMOS circuits.

For example, to determine the best alignment of two strings using edit-distance as a figure of merit, a 2D array of simple cells (quadratic complexity in string length) can be configured in a way that various alignments correspond to different paths going from one corner of the array to the opposite corner. Horizontal, vertical, and diagonal signal moves have different latencies associated with their edit "cost." A signal inserted in the input corner will travel through multiple paths in the array, with the one corresponding to the lowest edit-cost getting to the output corner first.

Space-time computing [9], which mimics spike-based neural computation in the neocortex, provides a key role for time as a communication and computation resource. To elaborate, in conventional digital circuits, the travel times of signals are relevant only to the extent that they determine the clock cycle in synchronous circuits or the latency in asynchronous logic. In space-time computation, signal timing also carries information.

For example, to transmit one byte of data, one may send a start/synch spike, followed by 8 data spikes/no-spikes, requiring 9 units of time in all and an average of 5 energy units. The same byte can be transmitted via start and end spikes, separated by a time interval in [1, 256]. The latter approach needs 2 units of energy and an average latency of about 128 cycles.

The two delay-based representations discussed in this section were provided only for completeness. As new methods that have seen limited use, we cannot yet judge their usefulness and applicability, or compare them usefully with the methods described in Sections 3-4 and with those of conventional digital arithmetic [10].

5. CVNS and CD-RNS Similarities

Two-level scheme: Both CVNS and CD-RNS use analog representation at the low (digit) level and digital interpretation at the high (inter-digit) level. The performance of the two schemes cannot be compared in an application-independent manner. For most application domains, CVNS will have an edge in speed, the exceptions being the same kinds of computational problems for which RNS has been found suitable [7], given the latter's overheads and inherently difficult operations, such as division [11].

Mixed-radix format: There is a superficial difference between CVNS and CD-RNS, in that the former uses a fixed integer radix, whereas the latter has different, (possibly) non-integer, moduli. Both ordinary RNS and CD-RNS have a close relationship with mixed-radix representations (such as the one used to denote time periods in days, hours, minutes, and seconds), which are sometimes used for magnitude comparisons in RNS. However, the idea of CVNS can be readily extended to mixed-radix representations, and, from there, to non-integer radices. As an example, consider a CD-RNS with the two moduli $\mu_1 = 4.4$ and $\mu_0 = 3.6$ (Example 2 in [6]). Assuming an error bound of $\varepsilon = \pm 0.4$, this CD-RNS has a range of $[0.0, 17.6)$, where the upper bound 17.6 is, counterintuitively, larger than the product of the moduli. A mixed-radix CVNS, with radices 3.6 and 4.4, will have a range of about $4.4 \times 3.6 = 15.84$. The representation of the number 9.0 is depicted in Fig. 3 for such a mixed-radix CVNS ($9.0 = 2.5 \times 3.6 = 2 \times 3.6 + 1.8$).

Approximate computing: Both CVNS and CD-RNS have an affinity with approximate computing [2] and related notions of adaptive-precision [12] and lazy [13] arithmetic. One can compute with several high-order digits in CVNS, switching to more precision only when needed, thus saving time and energy in the process. Similarly, using a subset of residues in CD-RNS allows lower-precision (or more error-prone) arithmetic, leading to energy savings and, for certain computations that need inter-residue operations, higher speed.

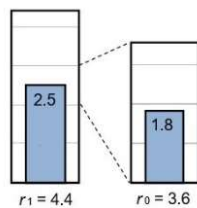


Fig. 3 Example of mixed-radix CVNS.

6. CVNS and CD-RNS Differences

Word-level parallelism: A key difference between CVNS and CD-RNS is that the latter offers the possibility of parallel processing of the digits for addition/subtraction and multiplication, with no inter-digit dependencies. Thus, in applications that are dominated by addition/subtraction and multiplication operations, CD-RNS may offer greater speed, provided that the faster operations occur frequently enough to compensate for the greater overheads elsewhere. Of course, various parallel-processing techniques have been applied to positional representations as well [14], but at the expense of higher area and energy costs.

Input/output overheads: Input conversion and output reconversion are more direct and, thus, much less costly in time and energy, for CVNS than the corresponding forward/reverse conversions of CD-RNS, which appear to be more difficult than those of ordinary RNS [15] [16]. On the other hand, the arithmetic operations themselves are, on average, simpler for CD-RNS, especially in add-and-multiply-heavy computations. Approximate reverse conversion using analog circuitry is rather simple [6], but not universally applicable.

Noise immunity: The range of CVNS is dictated by the number of digits, and its noise immunity by the precision of the individual analog digits. In CD-RNS, the range is dictated by the chosen moduli and the precision of the analog residues. Consider for concreteness a 2-digit CVNS and a 2-modulus CD-RNS with comparable ranges. Doubling the analog representation error will force us to change the CVNS radix from r to $r/2$, if the same noise immunity is to be maintained. This halving of the radix reduces the representation range quadratically. In CD-RNS, by contrast, the equation $\mu\varepsilon \leq \mu_0\mu_1$, introduced in Section 3, indicates that the dynamic range decreases linearly with an increase in the representation error ε . Thus, even though the two systems are similar in the way they deal with imprecision, CD-RNS has an edge in terms of the effect of imprecision on dynamic range and thus on VLSI area cost and energy consumption.

Fault tolerance: Many methods of encoding numerical values and performing arithmetic in fault-tolerant mode are available. Such methods can be applied to CVNS, which is a positional/weighted number representation system. CD-RNS has a built-in capability for fault tolerance via a precision-robustness trade-off. Providing additional moduli has the dual effect of increasing the range and/or improving fault tolerance.

7. Conclusion and Future Work

There is ample evidence that analog computing is making a comeback and that hybrid digital/analog arithmetic is becoming more attractive in light of evolving circuit technologies and computing applications. Our review indicates that existing hybrid representations are becoming ripe for practical realizations and that additional alternatives may emerge from further advances in technology and greater understanding of computational paradigms and application characteristics.

There are many ways in which the strengths of analog and digital computing can be combined, with the number-representation domain constituting but one of these ways. Where approximate values are needed, either as end results or as seed values in iterative refinement processes, they can be obtained quickly and with low circuit and energy costs via analog computations [17]. Neuromorphic computing (e.g., [18]), a paradigm that mimics neurobiological architectures, is another domain where analog circuit elements are used to great advantage.

There is, of course, no reason why the benefits of hybrid analog-digital representation (circuit level) cannot be combined with hybrid computation at the architecture level, as discussed in the preceding paragraph.

Further work on analog-digital arithmetic can proceed in several directions, including more detailed comparisons of CVNS and CD-RNS, perhaps using implementations in representative application domains to derive quantitative results, comparisons with the newer methods discussed in Section 4, and identifying alternative approaches. Selection of the number representation radix in CVNS and the moduli in CD-RNS, and how the choices impact the relative merits of the two systems, is also worth studying. It would be interesting to find out whether a single analog-circuit technology can be used to implement both systems, because such an eventuality will facilitate comparisons in circuit complexity and energy costs.

References

- [1] M. Hilbert and P. Gomez, "The World's Technological Capacity to Store, Communicate, and Compute Information," *Science*, Vol. 332, pp. 60-65, 2011.
- [2] S. Mittal, "A Survey of Techniques for Approximate Computing," *ACM Computing Surveys*, Vol. 48, No. 4, pp. 62:1-62:33, 2016.
- [3] G. Korn, and T. Korn, *Electronic Analog Computers*, 2nd ed., McGraw-Hill, 1956.
- [4] A. Saed, M. Ahmadi, and G. A. Jullien, "A Number System with Continuous Valued Digits and Modulo Arithmetic," *IEEE Trans. Computers*, Vol. 51, No. 11, pp. 1294-1305, 2002.
- [5] M. Mirhassani, M. Ahmadi, and G. A. Jullien, "Low-Power Mixed-Signal CVNS-Based 64-bit Adder for Media Signal Processing," *IEEE Trans. VLSI Systems*, Vol. 16, No. 9, pp. 1141-1150, 2008.
- [6] B. Parhami, "Digital Arithmetic in Nature: Continuous-Digit RNS," *Computer J.*, Vol. 58, No. 5, pp. 1214-1223, May 2015
- [7] M. A. Soderstrand *et al.* (eds.), *Residue Number System Arithmetic: Modern Applications in Digital Signal Processing*, IEEE Press, 1986.
- [8] A. Madhavan, T. Sherwood, and D. Strukov, "Race Logic: Abusing Hardware Race Conditions to Perform Useful Computation," *IEEE Micro*, Vol. 35, No. 3, pp. 48-57, May 2015.
- [9] J. E. Smith, *Space-Time Computing with Temporal Neural Networks*, Synthesis Lectures on Computer Architecture, Vol. 12, No. 2, 215 pp., 2017.
- [10] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford, 2nd ed., 2010.
- [11] C. Y. Hung and B. Parhami, "An Approximate Sign Detection Method for Residue Numbers and Its Application to RNS Division," *Computers & Mathematics with Applications*, Vol. 27, No. 4, pp. 23-35, 1994.
- [12] J. R. Shewchuk, "Robust Adaptive Floating-Point Geometric Predicates," *Proc. 12th ACM Symp. Computational Geometry*, pp. 141-150, 1996.
- [13] D. Michelucci and J. M. Moreau, "Lazy Arithmetic," *IEEE Trans. Computers*, Vol. 46, pp. 961-975, 1997.
- [14] B. Parhami, "Parallelism in Computer Arithmetic: A Historical Perspective," *Proc. 61st IEEE Int'l Midwest Symp. Circuits and Systems*, August 2018, to appear.
- [15] B. Parhami and C. Y. Hung, "Optimal Table Lookup Schemes for VLSI Implementation of Input/Output Conversions and Other Residue Number Operations," *Proc. Workshop VLSI Signal Processing (VLSI Signal Processing VII)*, pp. 470-481, 1994.
- [16] A. Persson and L. Bengtsson, L., "Forward and Reverse Converters and Moduli Set Selection in Signed-Digit Residue Number Systems," *J. Signal Processing Systems*, Vol. 56, No. 1, pp. 1-15, 2009.
- [17] G. E. R. Cowan, "A VLSI Analog Computer / Math Co-Processor for a Digital Computer," PhD dissertation, Columbia Univ., 2005.
- [18] R. J. Douglas, M. A. Mahowald, and K. A. C. Martin, "Hybrid Analog-Digital Architectures for Neuromorphic Systems," *Proc. IEEE World Congress Computational Intelligence*, Vol. 3, pp. 1848-1853, 1994.