

A Very Small Sub-Binary Radix DAC for Static Pseudo-Analog High-Precision Memory

William G. Bliss * C. Eric Seaberg † Randall L. Geiger ‡

Abstract

A sub-binary radix DAC which achieves high effective resolution in small VLSI area is demonstrated and is shown to be especially well suited for certain static analog memory applications, including tunable analog circuits. An intentional sub-binary radix is used to create a redundant number system with radix less than two, while still using normal binary coefficients (on/off control). A current mode DAC with greater than ten bits effective resolution was implemented in only 41 mil² area in a 2 μ CMOS technology. Achieving high resolution with the sub-binary scheme isn't dependent on accurate device matching, so chip yield and high resolution are not problematic. A simple iterative algorithm to control (set) these non-linear and non-monotonic DACs is detailed.

1 Introduction

It is well known that achieving high accuracy monolithic analog integrated circuits in the presence of process variations requires that these circuits be tuned or compensated. For continuous time linear filters in particular, the tuning must compensate for initial time constant shifts as large as $\pm 50\%$, while a control resolution of $\pm 0.1\%$ or better may be required for high accuracy system performance. Thus, a desirable tuning mechanism would have one part in a thousand resolution, i.e., ten bit resolution. Typically many analog parameters must be tuned, so the tuning devices must be kept small. Since there are undesirable side effects of continuous tuning, such as clock feed through, it is often desirable to perform the tuning process infrequently, possibly during a special tuning mode when the filter is off-line. In such applications it is desirable to have long term stability (e.g., $> 10^{-3}$ sec.), or storage of tuning information. The required re-tuning rate is mainly dictated by temperature variations. Unfortunately, traditional approaches to high accuracy and highly stable analog memories have required very large area, and have often been plagued by low yield. In this paper we show that achieving high resolution itself doesn't require large area or low yield. We present a solution which requires an unconventional and sometimes inconvenient control mechanism. However, for certain applications, including tuning and analog memory, the control is shown to be simple and inexpensive.

Our solution uses a redundant number system, which we call a sub-binary radix number system, which is described in section 2. Our system uses a digital memory to drive the sub-binary DAC, which generates the desired analog values. Our method differs from conventional high-resolution DAC design in that no attempt is made to build an accurate, or linear, or monotonic basis DAC. Rather, the sub-binary DAC is intentionally designed to be non-linear and non-monotonic so as to maximize net effective resolution in small area given our control method. Conventional methods of achieving high resolution DACs tend to be very area intensive and/or require very complicated correction schemes. Our sub-binary DAC can actually be used to tune such self-calibrating DACs. The reader is referred to [4], [3], [2], [9],

[6], [10], [7], and [8] for a representative look at high resolution data converters and their complexity.

A simple iterative control algorithm for sub-binary DACs is detailed in section 3. The statistical insensitivity of the sub-binary DACs effective resolution to process variations is described in section 4. A general tuning application requiring static analog memory is demonstrated in section 5. The design and measured results of an IC 10 bit effective resolution current mode sub-binary DAC is presented in section 6.

2 Sub-binary radix number systems

The sub-binary radix system is based on a binary weighted (coefficient) number system with non-binary radices, all less than two. A quantity (number), Q , in a radix R number system of order k (k digits) can be represented as

$$Q = a_k \cdot R^k + a_{k-1} \cdot R^{k-1} + \dots + a_1 \cdot R + a_0 \quad (1)$$

An alternative representation of (1) is

$$Q = (a_k a_{k-1} \dots a_1 a_0)_R \quad (2)$$

In a conventional radix R system, R is a positive integer and the coefficients, $a_0, a_1, \dots, a_{k-1}, a_k$, are chosen $a_i \in \{0, 1, \dots, R-1\}$. While the coefficients could be fractional, we will limit them to the binary bits $a_i \in \{0, 1\}$, called binary weighting. Binary coefficients are chosen since we will use conventional digital binary storage and binary (on/off) switches. An example of the normal uniform binary radix $R = 2$ (binary) is:

$$(10)_{10} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 \quad (3)$$

If the radices, R_k in (1), vary from term to term, i.e., $R_k \neq r^k$, then we say we have a mixed radix system. Mixed radix systems are useful in describing the operation of real data converters, even if they were designed for a constant uniform radix, because process errors generate $R_i = R + e_i$. For example if a data converter is designed with a uniform radix of two, the result after random process variation of the converter components might cause the values of the k radices to spread both above and below the designed radix of two. We use the term sub-binary radix number system for both binary weighted mixed radix number systems and binary weighted constant radix number systems with radices less than two.

Visualizing how numbers are represented in a binary weighted mixed radix system is difficult, so we first consider binary weighted constant radix systems. Here the radix is not necessarily equal to two, but it stays constant throughout the terms in 1. Fig. 1 shows several examples of 4-bit binary weighted constant radix number systems. Each system has 16 possible values that it can represent

$$(0000)_R, (0001)_R, \dots, (1110)_R, (1111)_R \quad (4)$$

The x axis in Fig. 1 is labeled with the decimal integer from 0 to 15 corresponding to the normal base two (radix 2) interpretation of the four control bits. The y axis represents the actual values generated by each particular radix. Note that when the radix is equal to two, the values increase monotonically with a constant step size. If the radix is greater than two, monotonicity is preserved, but step size is no longer constant and large gaps occur

*This research was supported by a grant from the Texas Advanced Research Program. Dr. Bliss was with the Dept. Elec. Eng., Texas A&M Univ., College Station, Tx. 77843.

†Mr. Seaberg is now with Motorola Inc., Austin, Tx.

‡Prof. Geiger now heads the Dept. of Elec. Eng., Iowa State Univ., Ames, Iowa.

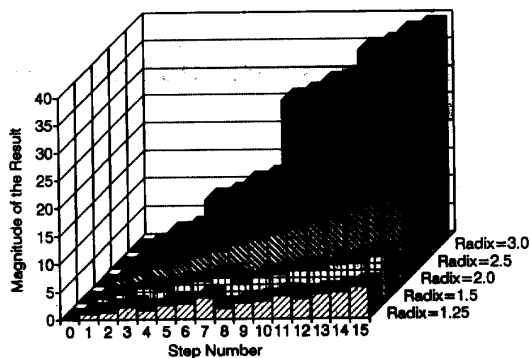


Figure 1: A visual demonstration of the values realized using different radices with binary control bits.

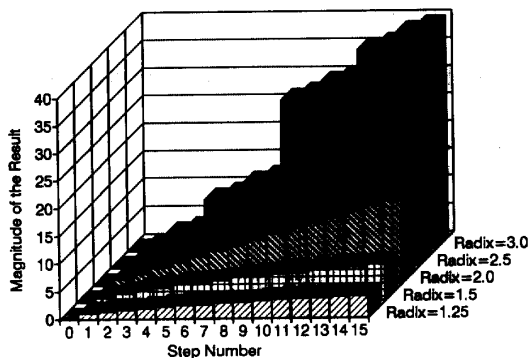


Figure 2: The values realized by different radices displayed sorted.

in the represented numbers. If the radix is decreased below two, neither monotonicity nor constant step size is maintained. Also, significant overlaps in the values represented occur, i.e., several binary control words may represent approximately the same value Q . While this would be a disadvantage in most systems, we exploit the natural trade off between redundancy and resolution. Every non-monotonic step (jog down) in the binary counting sequence results in redundant values, meaning that the sub-binary systems have less dynamic range than normal radix two systems. To achieve equal dynamic range and effective resolution as binary systems, the sub-binary systems must therefore use one or two extra DAC bits.

An important performance measure of data converters is their resolution. The tuning DAC must have sufficient resolution to manipulate the parameters of the circuit being tuned to within specifications. In Fig. 2 the values represented in the example 4-bit binary coefficient constant radix systems have been sorted to illustrate the effective resolution. In Fig. 3 the difference between adjacent sorted values is plotted. These differences are called the incremental step size. Notice that as the radix is increased past two, the maximum error that can occur increases due to the widening gaps between successive numbers. As the radix is decreased below two (sub-binary radix), the difference between successive sorted numbers decreases. The maximum step size also decreases, but the sub-binary radix system doesn't cover as large a range as the base two system. This lack of coverage can be compensated for by adding additional bits to the sub-binary systems, extending the range to the desired level. Sub-binary radix number systems are thus redundant, but if the radix is close to $R = 2$, then they are not highly redundant. The sub-binary system can achieve any desired resolution, but control is now non-linear. Obviously we would prefer not to have to learn and/or measure the exact values of the implemented radices. And further, we don't

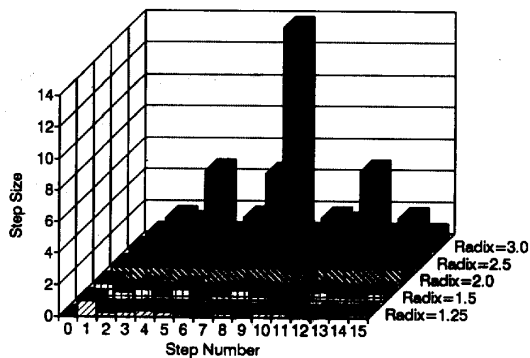


Figure 3: Step size between sorted values for different radices.

want to have to sort through the large number of possibly close representations. Fortunately, a simple iterative algorithm exists that avoids sorting or table look-up.

3 A simple iterative control algorithm for sub-binary radix DAC

A control algorithm is needed for practical use of high resolution sub-binary DACs. If the DAC must have a normal digital power of two binary number interface, which additionally must run at the maximum speed, then there is little choice but to learn the radices exactly and perform a complicated digital binary to sub-binary radix conversion. However, the typical analog-memory or tuning environment has far different requirements. Typically there is no binary number digital input, but rather a simple test indicating that the analog value is either too large or too small. Also, the speed of tuning is typically not limited by DAC speed but by the response time of the circuit being tuned and by the measurement circuit itself. A simple iterative algorithm using $k + 1$ iterations to control $k + 1$ sub-binary radices is given below.

```

Algorithm: Control sub-binary radix DAC
c   Test(D,Q) = too high or too low
%is only measurement information
c   D is desired, Q is actual
c   a(i) is i'th control bit
Initialize: a(i)=0 for all i
For i=k to 0 by -1
  set a(i)=1
  if Test(D,Q)= too_high
    then a(i)=0,
  end if
Return
End

```

This control algorithm first enables (turns on) the most significant control bit and checks to see whether the tuning test (synthesized analog output) is too high or not. After setting this bit, the next most significant bit is enabled and the tuning test is checked. This process continues until all of the control bits have been set. A dual algorithm initializes the DAC (Q) as all ones and tunes down. In the next section we examine the statistical behavior of the sub-binary radix system using this simple control algorithm.

4 Tolerance of sub-binary radix systems to random radix variations

We now consider the effect of random perturbations of the designed radices by imperfect processing. First consider the worst

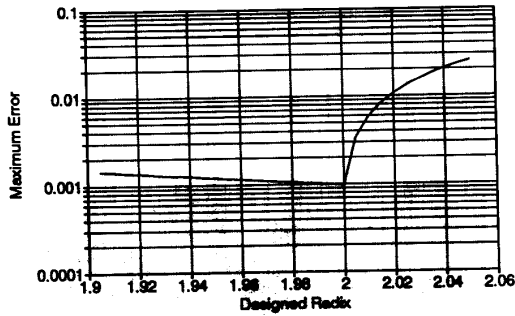


Figure 4: Maximum relative step size as a function of the designed radix.

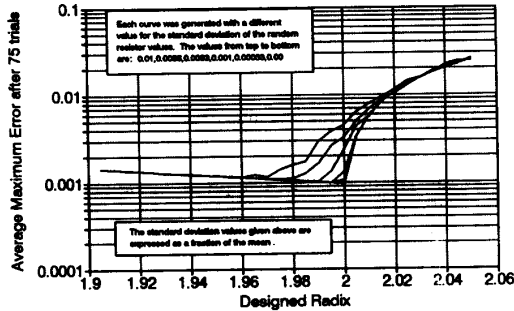


Figure 5: Effect of random radix variations on resolution error

case approximation error of constant radix systems. Fig. 4 shows how the maximum relative step size, defined as the worst case step size normalized by full scale, varies with the chosen radix. Notice that the error climbs very fast when the radix is increased above two. A simple interpretation of Fig. 4 is that radices that randomly fluctuate above two will cause severe loss of resolution.

Random processes inherent in manufacturing will create a mixed radix number system. For simplicity, assume a zero mean Gaussian distribution of radices. Fig. 5 shows the effect of random radix variation on a simple R2R type DAC structure with ten control-bits. The bottom curve corresponds to no variation, while the variance steadily increases for the upper curves. The y axis shows the maximum step size averaged over 75 experiments. The 'optimal' choice of design radix occurs where the curve for a given variance) drops to its lowest error value. This point is highly dependent on the variance of the process. Fundamentally, the radix must be chosen low enough so that random perturbations are not likely to increase some radix above two. The sub-binary radix system can be thought of as design for statistical yield optimization. Note that the error only increases slowly with decreasing radix below two. Thus, a conservative design approach can achieve both good resolution and effectively 100% yield (due to accuracy). We have proven that a sufficient condition for effective resolution equal to one-half of the least significant radix is:

$$R_i \leq R_{i+1} \quad \text{for all } i \quad (5)$$

In practical applications, some number G of the least significant radices, R_0, R_1, \dots, R_{G-1} can be designed as normal powers of two. The number G of such radices is the largest number of bits that can be implemented *easily* in a given technology and circuit style without resorting to tuning, sub-binary radices, etc. The higher significant digits can then be implemented with decreasing sub-binary radices.

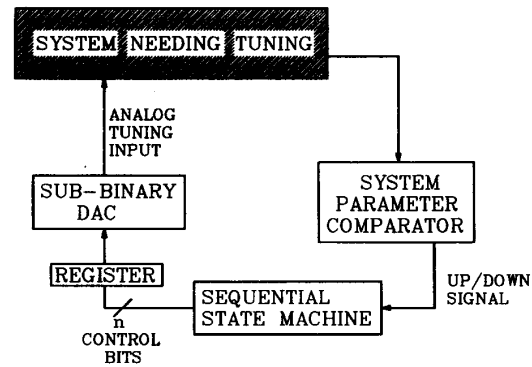


Figure 6: A sub-binary DAC tuning system.

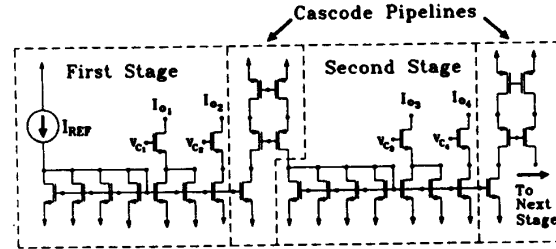


Figure 7: Two stages of a sub-binary radix DAC, with two control bits per stage.

5 A binary weighted sub-binary radix DAC for tuning analog circuits

Consider the generic analog circuit with associated tuning system of Fig. 6. We propose to set the control bits of the sub-binary DAC by using the iterative algorithm of section 3. The only information needed at each iteration is simply: is the present DAC output too high (or too low). Note that the analog circuit being tuned could itself be a DAC or Analog to Digital Converter (ADC), where the ADC has its radices tuned to exact powers of two. Here the sub-binary DAC supplies the analog tuning input to the system needing tuning. A test system comparator measures the status of the system being tuned. The comparator function will vary greatly depending on what type of parameters need to be measured. For example the test comparison block could measure the frequency, magnitude, or phase of sine waves. The necessary output of this comparator is an up/down signal used in the approximation algorithm for the sub-binary DAC. The approximation algorithm is encoded in the state machine. The output of the state machine is stored in a static register holding the DAC control bits.

6 Design of a sub-binary radix DAC for tuning

The design goal for the DAC was to produce a small digitally controlled circuit with over ten-bits resolution to tune typical OTA-C filters or data converters. The designed sub-binary DAC uses 12 control bits. OTA based circuits are typically tuned by adjusting their transconductance gain via the differential pair tail current. Since the DAC is to supply a bias current, its output should be constant over time. A building block of this DAC is shown in Fig. 7. This implementation uses simple current mirrors to scale down the reference current within each block. If this scaling were used for the entire structure, then the largest transistor would be 2^{12} times larger than the smallest. This would consume far

too much area. To solve this problem, the tuning DAC was implemented in stages similar to a R-2-R DAC structure. In each stage the current is twice divided by two and then passed to the next stage. This procedure reduces the area because the ratio of transistor sizes is compared against the first transistor in each stage, not to the first one in the DAC. Therefore the maximum ratio between transistors in each stage is 4. The choice of two divisions per stage was a compromise between having large area used for large transistor size ratios and having large area overhead for transistors connecting the stages. While it appears each stage might be identical, each stage must be sized according to the current it handles. The first stages must be wider to keep the transistors in saturation, while the last stages must be longer to keep the transistors out of subthreshold.

To make this tuning circuit a sub-binary radix structure, the gain of the cascode pipe between stages is made greater than one, while the ratios of the transistors inside each stage were kept at ratios of two. This made it easier to size the transistors in each stage. Only integer sizes of transistor lengths and widths are allowed, and changing one unit in a small device causes a large change in the ratio. It was therefore easier to implement the fractional gain ratios in the less numerous cascode connecting pipes. Each current can be controlled by the MOS switch at the drain of the output current devices. The final size of a 12-bit tuning DAC was approximately 40 mil^2 in $2/\mu\text{m}$ MOSIS CMOS. Note that approximately 100 such DACs can be fit on a Tiny MOSIS chip, and over 10,000 could be put on a modern large chip. Maintaining the effective theoretical resolution indicated may require that superposition of the selected current sources hold at the current adder. If the iterative control algorithm is used, this non-superposition problem is greatly lessened. Spice simulations show it isn't a problem in this application.

Many other circuit techniques can be used to achieve sub-binary radices. For example, we can use power of two division chains, and then apply a small amount of feedback from output to input. While this sub-binary DAC was implemented using MOS current sources, the basic idea of sub-binary radix systems can be applied, for example, to capacitance [5], resistance, or charge selection circuits. Each circuit technique will have its own dynamic range and integer sizing tradeoffs, so different variants of sub-binary radices may be found to best match these tradeoffs.

An important practical issue is the area required by digital memory to drive the tuning DACs. In other words, the sub-binary DACs are so small that the digital latches holding control bits may dominate area. In typical analog monolithic IC applications the large parameter variation of manufacture roughly tracks across chip. Thus, a coarse tune representing, e.g., tuning from 50% to 3% can be performed for the whole chip using only 4 digital storage latches. The fine tuning of some N components to, e.g., 0.1% can then be accomplished with four fine tuning bits for each of the N components. For example, a 7th order leapfrog filter requires about 32 bits of on chip storage, which is reasonable.

7 Results and Conclusions

Our design was fabricated in 2μ MOSIS CMOS process and tested. The measured response was highly non-linear and non-monotonic as expected. The measured step size between adjacent values is shown in Fig. 8. Notice that the worst case step size is very close to 2^{-10} , which was the design resolution (i.e., the design radix was decreased the maximal amount to enhance yield while still achieving ten bit resolution).

We have shown that achieving high resolution in analog values for monolithic ICs can simply be extended beyond component matching if simple tuning tests can be accommodated at the system level. Achieving this high accuracy simply and in small area is most naturally done by using a redundant fractional radix number system, which we call sub-binary radix. While using such a number system for typical DAC applications might be very cumbersome, in the tuning environment it is not a hindrance and is

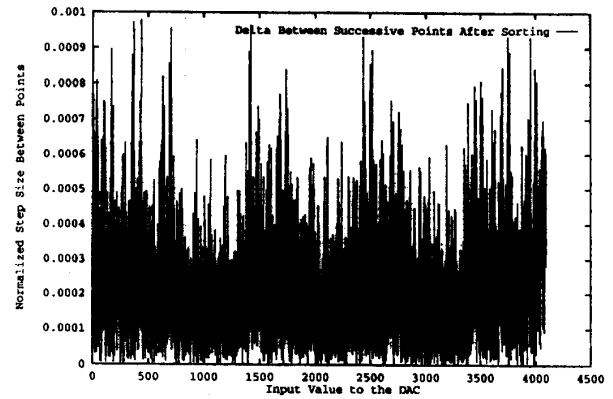


Figure 8: Measured response of fabricated 12-bit DAC with sub-binary radix.

simple to implement. We have demonstrated the technique with the design of a small DAC with over ten bits of effective resolution using twelve control bits. The design extends simply and easily to greater resolution, but our prototype application only required ten bits of resolution [1]. The use of sub-binary DACs extends naturally to the design of writable (pseudo) analog static memory. As long as write access time is long enough to allow the iterative algorithm to converge, then no other special circuitry is required.

References

- [1] W.G. Bliss and R. Sadkowski. High precision in situ tuning of monolithic continuous time leapfrog filters. *to appear in the Proc. IEEE Int. Conf. Circuits and Systems*, 1992.
- [2] Z. G. Boyacigiller, B. Weir, and P. D. Bradshaw. An error-correcting 14b/20us cmos A/D converter. *ISSCC Dig. Tech. Papers*, pages 62–63, February 1981.
- [3] J. J. Connolly, T. O. Redfern, S. W. Chin, and T. M. Fredrikson. A monolithic 12b+sign successive approximation A/D converters. *ISSCC Dig. Tech. Papers*, pages 12–13, February 1980.
- [4] H. Lee, D. Hodges, and P. Gray. A self-calibrating 15 bit cmos A/D converter. *IEEE Journal of Solid-State Circuits*, pages 813–819, 1984.
- [5] C.A. Leme and J.E. Franca. Efficient calibration of binary-weighted networks using a mixed analogue-digital ram. *Proc. IEEE Int. Conf. Circuits and Systems, Singapore*, pages 1545–1548, 1991.
- [6] K. Maio, *et al.* An untrimmed D/A converter with 14-bit resolution. *IEEE Journal of Solid-State Circuits*, SC-16:616–621, December 1981.
- [7] Y. Manoli. A self-calibration method for fast high-resolution A/D and D/A converters. *IEEE Journal of Solid-State Circuits*, pages 603–608, 1989.
- [8] H. Onodera, T. Tateishi, and K. Tamaru. A cyclic A/D converter that does not require ratio-matched components. *IEEE Journal of Solid-State Circuits*, pages 152–158, 1988.
- [9] R. J. van de Plassche. Dynamic element matching for high-accuracy monolithic D/A converters. *IEEE Journal of Solid-State Circuits*, SC-11:795–800, December 1976.
- [10] D. Wouter, *et al.* A self-calibration technique for monolithic high-resolution D/A converters. *IEEE Journal of Solid-State Circuits*, pages 1517–1522, 1989.