Efficiency Limits for Value-Deviation-Bounded Approximate Communication

Phillip Stanley-Marbell and Martin Rinard

***Abstract*—Transferring data between integrated circuits accounts for a growing proportion of system power in wearable and mobile systems. The dynamic component of power dissipated in this data transfer can be reduced by reducing signal transitions. Techniques for reducing signal transitions on communication links have traditionally been targeted at parallel buses and can therefore not be applied when the transfer interfaces are serial buses.**

In this article, we address the issue of the best-case effectiveness of techniques to reduce signal transitions on serial buses, if these techniques also allow some error in the numeric interpretation of transmitted data. For many embedded applications, exchanging numeric accuracy for power reduction is a worthwhile tradeoff. We present a study of the efficiency of these *value-deviation-bounded approximate serial data encoders* **(VDBS data encoders) and proofs of their properties.**

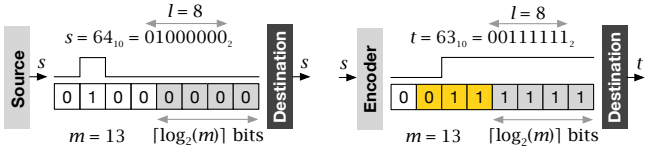
The bounds and proofs we present yield new insights into the best possible tradeoffs between dynamic power reduction and approximation error that can be achieved in practice. The insights are important regardless of whether actual practical VDBS data encoders are implemented in software or in hardware.

***Index Terms*—Approximate Computing, Approximate Communication, Bounds, Data Encoding.**

I. Introduction

WEARABLE and health-tracking devices dissipate ever-larger fractions of their power on sensor activation and on data transfer between processors and sensor integrated circuits. Since package and circuit board capacitances do not improve with semiconductor process advances, the fraction will likely continue to grow relative to components such as processors. For reasons of space and cost, the data transfer happens over serial interfaces, and not over parallel buses. This precludes the use of traditional low-power bus data encodings [1, 2]. To address this challenge, *value-deviation-bounded approximate serial data encoders* (VDBS data encoders) [3] reduce the dynamic power dissipation of serial buses when deviations in the values being transmitted are tolerable.

This article presents a study of the efficiency of VDBS data encoders, as well as proofs of properties of VDBS data encoders. Our analyses provide an essential yardstick for evaluating practical VDBS data encoding algorithms that may be proposed in the future. The proofs of properties of VDBS data encoders on the other hand provide important new insights into how VDBS data encoders fit into the existing body of research on reducing dynamic power of both serial and parallel buses.



(a) Without VDBS encoding. (b) With VDBS: fewer transitions.

Fig. 1. In this example, assume the tolerable deviation, m, is 13 (i.e., 5 % of 255). VDBS encoding halves the number of transitions while incurring a value deviation, $|s-t|$, of just 0.39 % of the full-scale range. All bits except the most-significant bit are modified, not just the lower $\lceil \log_2(m) \rceil$.

II. Definitions

Two essential components in the formulation of VDBS encoders are:

❶ **The number of serial transitions** that occur when a single value s is transmitted over a serial link (two transitions in Figure 1(a)). We call this the *serial transition count* (STC) of the word or value s.

❷ **The difference in serial transition counts** between two words (a difference of one between s and t in Figure 1).

Throughout this work, the values considered will be unsigned.

Definition 1 (*Serial transition count function, $\#_\delta(s)$*)**.** Let s be an l-bit unsigned integer with bits $s_0, s_1, \ldots, s_{l-1}$, from least- to most-significant bit. Then, we define $\#_\delta(s)$, the number of signal transitions in the serialization of s, as

$$\#_\delta(s) = \sum_{i=0}^{l-2} s_i \oplus s_{i+1}. \qquad \blacksquare$$

Definition 2 (*Serial transition count difference, $\Delta_{s,t}$*)**.** Let s and t be two l-bit words. Then, we define $\Delta_{s,t}$, as the absolute value of their difference in serial transition counts:

$$\Delta_{s,t} = |\#_\delta(s) - \#_\delta(t)|. \qquad \blacksquare$$

III. Properties of Function $\#_\delta(n)$

For an l-bit value n, the properties of the serial transition count (STC) function $\#_\delta(n)$, which we explore next, give insights into the efficiency limits of VDBS encoders.

Proposition 1 (Maximum serial transition count pattern)**.** When l is even, the maximum serial transition count occurs when the l-bit word has $\frac{l}{2}$ 0s and the same number of 1s. □

Proof (Maximum serial transition count pattern)**.** To maximize the serial transition count, there should be a transition in moving between every neighboring pair of bit

P. Stanley-Marbell and M. Rinard are with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 02139. E-mail: psm@mit.edu, rinard@csail.mit.edu

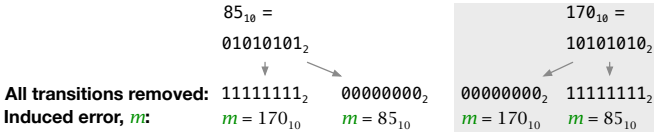


Fig. 2. The maximum serial transition counts for l-bit values occur when they have alternating 0s and 1s in their binary representations.

positions. Thus, when l is even, words with maximum serial transition count have $\frac{l}{2}$ 0s and the same number of 1s. ■

Corollary 1 (Maximum serial transition count basis values)**.** There are two values with maximum serial transition count. When l is even, these values are

$$\hat{b}_1 = \sum_{i=0}^{\frac{l}{2}-1} 2^{2i} = \tfrac{1}{3}(2^l - 1) \tag{1}$$

and

$$\hat{b}_2 = 2^l - 1 - \sum_{i=0}^{\frac{l}{2}-1} 2^{2i} = \tfrac{2}{3}(2^l - 1).$$

□

This follows directly from Proposition 1. For example, Figure 2 illustrates how, for $l = 8$, the maximal-serial-transition-count words are 85 and 170.

Lemma 1 (Maximum serial transition count)**.** For every l-bit word n, $\#_\delta(n) \leq l - 1$. □

Proof (Maximum serial transition count)**.** The number of bits (l) in a word is a natural number. When l is 1, there are no transitions in the word, by definition of the serial transition count. For all other l, the maximum serial transition count occurs when all adjacent bits of the word differ. There are four cases in which this could happen, corresponding to whether l is even or odd, and whether the least-significant bit (LSB) is a 1 or a 0.

First, consider the cases when l is even. When l is even, there are $\frac{l}{2}$ ones and $\frac{l}{2}$ zeros. If the LSB is 0, there will be one transition in moving from the LSB towards the most-significant bit (MSB), and each of the remaining $\frac{l}{2} - 1$ bits which are 0 will have two associated transitions. There will therefore be a total of $l - 1$ transitions. A similar argument applies if the LSB is 1.

Next, consider the cases when l is odd. When l is odd, there are either $\lfloor \frac{l}{2} \rfloor$ bits which are 1 and $\lceil \frac{l}{2} \rceil$ bits which are 0, or vice versa. The bit polarity appearing in the LSB will occur $\frac{l-1}{2} + 1$ times, and the opposite polarity to the LSB will occur $\frac{l-1}{2}$ times.

There will be one transition moving out of the LSB towards the MSB, followed by transitions in the remaining $l - 1$ bits. Since l is odd, it follows that $l - 1$ is even. But we showed above that such an even number of bits could contain at most $(l - 1) - 1$ transitions. Thus, when l is odd, the maximum number of transitions is also $1 + (l - 1) - 1$. That is, the maximum number of transitions is $l - 1$. ■

Theorem 1 (Serial transitions and Gray code)**.** Let s be an l-bit integer, let $\mathrm{GrayCode}(s)$ denote the sth value in Gray code order for l-bit values, and let $\#_1(n)$ denote the count of 1s in an l-bit integer n. Then, $\#_\delta(s) = \#_1(\mathrm{GrayCode}(s))$. □

$\mathcal{L}_0 = \{\}$	$l = 0$
$\mathcal{L}_0^0 = \{0\}$ $\overline{\mathcal{L}_0}^1 = \{1\}$ $\mathcal{L}_1 = \{0, 1\}$	$l = 1$
$\mathcal{L}_1^0 = \{00, 01\}$ $\overline{\mathcal{L}_1}^1 = \{11, 10\}$ $\mathcal{L}_2 = \{00, 01, 11, 10\}$	$l = 2$
$\mathcal{L}_2^0 = \{000, 001, 011, 010\}$ $\overline{\mathcal{L}_2}^1 = \{110, 111, 101, 100\}$ $\mathcal{L}_3 = \{000, 001, 011, 010, 110, 111, 101, 100\}$	$l = 3$

Fig. 3. Illustration of the construction of the list $\mathcal{L}_l$ of l-bit strings in Gray code order, for $l = 1$, $l = 2$, and $l = 3$.

For example, for $l = 8$ and $s = 30$ (00011110_2), $\mathrm{GrayCode}(s) = 17$ (00010001_2) and $\#_1(\mathrm{GrayCode}(s)) = 2$.

We will use the following, the Gray code theorem of Wilf [4], in the proof of Theorem 1. We include a self-contained adaptation of Wilf's original proof here so that our discussion stands on its own.

Theorem 2 (Wilf's Gray Code Theorem)**.** Let s be an l-bit integer with bits $s_0, s_1, \ldots, s_{l-1}$, from least- to most-significant bit. Let g be the sth l-bit integer in Gray code order, with bits $g_0, g_1, \ldots, g_{l-1}$, from least- to most-significant bit. For $l = 1$ we have $g_0 = s_0$. In general, for $l \geq 2$,

$$g_i \equiv s_i + s_{i+1} \pmod 2 \quad (i = 0, \ldots, l-2)$$

and

$$g_{l-1} = s_{l-1}.$$ □

For example, consider the 8-bit value 63. The string of rank 63 in the 8-bit Gray code, that is, the 63rd Gray code value, can be constructed as follows: For the ith bit, simply take the ith and $i + 1$th bits of 63, and add them modulo 2.

Proof (Wilf's Gray Code Theorem)**.** Let $\mathcal{L}_l$ be the list of l-bit strings in Gray code order. $\mathcal{L}_0$ is the empty list. The list $\mathcal{L}_l$ can be constructed recursively as follows:

- Let $\mathcal{L}_{l-1}^0$ be the list obtained by prefixing every element of $\mathcal{L}_{l-1}$ with an additional 0.
- Let $\overline{\mathcal{L}_{l-1}}^1$ be the list obtained by prefixing every element of the list $\mathcal{L}_{l-1}$, in reverse order, with an additional 1.
- $\mathcal{L}_l$ is the concatenation of $\mathcal{L}_{l-1}^0$ and $\overline{\mathcal{L}_{l-1}}^1$.

The construction of the list $\mathcal{L}_l$ is illustrated in Figure 3 for $l = 1$, $l = 2$, and $l = 3$. By construction therefore, the 2^l entries for an l-bit Gray code will be identical to the first 2^l entries for an $(l+1)$-bit Gray code; we use this property below.

We prove by induction on l that the property of Theorem 2 holds for all l-bit integers s. When $l = 0$, $\mathcal{L}_l$ is the empty list, and the property we seek to prove is vacuously true. Suppose

the property of Theorem 2 holds for all strings on the list $\mathcal{L}_{l-1}$. By construction of $\mathcal{L}_l$, we know the property must also hold for the first 2^{l-1} items on $\mathcal{L}_l$. Suppose then, that $s \geq 2^{l-1}$. Let $s' = 2^l - 1 - s$. Then the property of Theorem 2 holds for the string that has Gray code rank s', since it is by its definition less than 2^{l-1}.

Again by construction of the Gray code lists $\mathcal{L}_l$ from $\mathcal{L}_{l-1}$, it is the case that for any given value $0 \leq k < 2^{l-1}$, the first $l-1$ bits of the Gray code strings g and g' with ranks $s = k$ and $s' = k$ are identical. Furthermore, the most-significant bits, g_{l-1} and g'_{l-1}, of these corresponding strings, have the relation

$$g_{l-1} \equiv 1 + g'_{l-1} \pmod 2.$$

At the same time, the binary representations of the integers s and s' have the relation

$$s_i \equiv 1 + s'_i \pmod 2 \quad (i = 0, \ldots, l-1),$$

and the property of Theorem 2 continues to hold for all strings on the list $\mathcal{L}_l$. ■

We now use Wilf's Gray code theorem to prove the property of Theorem 1, which relates properties of transitions within a single word, s, when serialized, to properties of the rank-s Gray code.

Proof (Serial transitions and Gray code)**.**
The proof is a direct result of Theorem 2. Let g be the Gray code representation for l-bit integer s. That is, g is the rank-s l-bit Gray code. The number of 1s in g, $\#_1(g)$, is

$$\begin{aligned}\#_1(g) &= \sum_{i=0}^{l-1} g_i \\ &= \sum_{i=0}^{l-2} \left(s_i + s_{i+1} \pmod 2\right), \quad \text{from Theorem 2} \\ &= \sum_{i=0}^{l-2} \left(s_i \oplus s_{i+1}\right).\end{aligned}$$

But this is exactly the $\#_\delta(s)$ from Definition 1. ■

IV. Bounds on Serial Transition Count Reduction

We can reduce the number of serial transitions in words without changing the word size, by introducing errors into the values represented by words. The maximum number of serial transitions we can remove by doing so, is limited:

Property 1 (Bound on serial transition count difference)**.**
For any two l-bit words s and t, the serial transition count difference, $\Delta_{s,t}$ is less than or equal to $l-1$. □

Proof (Bound on serial transition count difference)**.**
By construction, the serial transition count, $\#_\delta(s)$ for a non-negative integer s, is a natural number. Therefore, the largest serial transition count difference, will occur when either $\#_\delta(s)$ is zero and $\#_\delta(t)$ takes on the maximum value in the codomain of $\#_\delta(t)$, or vice versa. From Lemma 1, this maximum value is $l-1$. Thus the maximum serial transition count difference, $\Delta_{s,t}$ is $l-1$. ■

Across all possible l-bit words, the deviation induced when transitions are reduced by the maximum of $l-1$, is bounded:

Property 2 (Minimum and maximum deviation at maximum serial transition count difference)**.**
Let s and t be two l-bit words with l even. If s and t differ in serial transition count by the maximum possible amount $(l-1)$, then their difference in numeric value is bounded by:

$$\min_{\Delta_{s,t}=l-1} \{|s-t|\} = \tfrac{1}{3}\left(2^{\Delta_{s,t}+1} - 1\right), \tag{2}$$

and

$$\max_{\Delta_{s,t}=l-1} \{|s-t|\} = \tfrac{2}{3}\left(2^{\Delta_{s,t}+1} - 1\right). \tag{3}$$

□

Proof (Minimum and maximum deviation at maximum serial transition count difference)**.**
Follows directly from Corollary 1. ■

For example, for $l = 8$, we have from Lemma 1 that the maximal serial transition count difference is $l - 1 = 7$. The minimum deviation between two words which have this maximum serial transition count difference, from Property 2, is 85. Therefore, to reduce the serial transition count of an 8-bit word by 7 transitions, one cannot do so with a replacement word that deviates from it by less than 85.

The bounds of Property 2 are only specified for the case of maximal changes in serial transition count, not for any arbitrary reduction in serial transition count. General bounds across all possible values of serial transition count reduction are desirable, because they would enable us to answer questions such as:

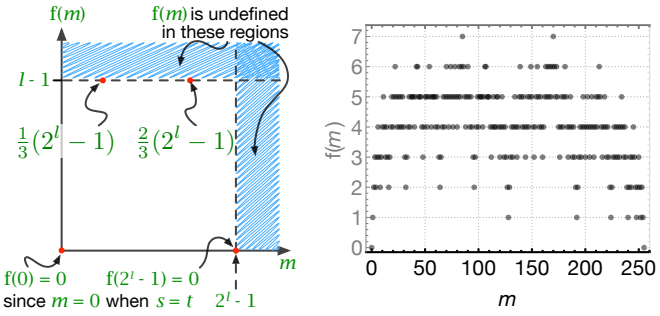
- **By how much can serial transition counts differ** for a given value deviation? This will be captured by Definition 3 and Theorem 3 below.
- **By how much can values differ** for a given difference in serial transition count? Property 2 answers this question for the restricted case of a serial transition count difference of $l-1$. The answer for the general case will be captured by Definition 4 below.

Definition 3 (Serial transition difference bound function)**.**
Given an l-bit integer m, let $\mathrm{f}(m)$ be a function yielding the amount by which the serial transition counts of two unsigned l-bit words s and t can differ if $|s-t| = m$. That is,

$$\mathrm{f}(m) = \max_{|s-t|=m} \{\Delta_{s,t}\}. \quad ■$$

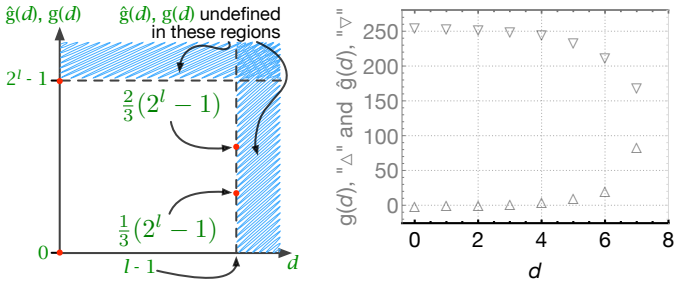
Why $\mathrm{f}(m)$ is important: The function $\mathrm{f}(m)$ is interesting because, if one had an exact expression or tight bounds for $\mathrm{f}(m)$, then an algorithm that searched for the serial-transition-reducing encoding for a value s could terminate as soon as it found a value t such that $\Delta_{s,t} = \mathrm{f}(m)$, since no better value than t is possible.

Theorem 3 (Bound on $\mathrm{f}(m)$)**.**
The function $\mathrm{f}(m)$ of Definition 3, for any l-bit value, m (with l even), is not monotone. The best linear monotone bound on $\mathrm{f}(m)$ is $\mathrm{f}(m) \leq l - 1$. □



(a) Illustration of $\mathrm{f}(m)$. (b) Numerical evaluation of $\mathrm{f}(m)$.

Fig. 4. The function $\mathrm{f}(m)$ yielding the amount by which the serial transition counts of two words s and t can differ if $|s-t|=m$, is not monotone.



(a) Illustration of $\mathrm{g}(d)$ and $\hat{\mathrm{g}}(d)$. (b) Numerical evaluation: $\mathrm{g}(d)$, $\hat{\mathrm{g}}(d)$.

Fig. 5. At minimum serial transition count (STC) difference, $d=0$, either s and t are identical, or they are different but take on values $s=0$ and $t=2^l-1$ or vice versa.

Proof (Bound on $\mathrm{f}(m)$)**.**
Let s and t be two unsigned l-bit words, and let m be $|s-t|$, a value in the domain of f. If m is 0, then s is identical to t, and must have identical serial transition count, thus $\#_\delta(s)=\#_\delta(t)$ and therefore $\mathrm{f}(0)=0$. If m is 2^l-1, then either s is 2^l-1 and t is zero, or vice versa. In both cases, their serial transition counts are 0 by definition, that is $\#_\delta(s)=\#_\delta(t)=0$. Thus, when m is 2^l-1, $\mathrm{f}(m)=0$.

From Corollary 1 and Lemma 1, the maximum value of $\mathrm{f}(m)$ is $l-1$, and it occurs at two values, $\hat{b}_1$ and $\hat{b}_2$ from Equation 1. Both $\hat{b}_1$ and $\hat{b}_2$ are greater than 0 and less than 2^l-1. Since $\mathrm{f}(0)$ is 0, $\mathrm{f}(\hat{b}_1)$ is $l-1$, $\mathrm{f}(\hat{b}_2)$ is $l-1$, and $\mathrm{f}(2^l-1)$ is 0, it follows that $\mathrm{f}(m)$ is not monotone.

From Corollary 1 and Lemma 1, since there are two values of m for which $\mathrm{f}(m)$ takes on its maximum value of $l-1$, it follows that the tightest linear bound on $\mathrm{f}(m)$ must pass through these points. Thus the tightest linear bound on $\mathrm{f}(m)$ is $l-1$. ■

Figure 4(a) illustrates several properties of $\mathrm{f}(m)$, and Figure 4(b) shows an empirical exact enumeration of $\mathrm{f}(m)$ across all possible unsigned 8-bit values. The maximum value of m is 2^l-1 and the maximum value of $\mathrm{f}(m)$ is $l-1$, as indicated by the shaded region in Figure 4(a). There can be no reduction in serial transition count when the accompanying deviation in value is 0, and thus $\mathrm{f}(0)=0$. Similarly, when the deviation induced by encoding is 2^l-1 (i.e., the original and encoded values are 0 and 2^l-1 or vice versa), there can be no reduction in serial transition count, and thus $\mathrm{f}(2^l-1)=0$. The maxima of $\mathrm{f}(m)$ occur at $m=\frac{1}{3}(2^l-1)$ and $m=\frac{2}{3}(2^l-1)$.

Definition 4 (Value deviation bound functions)**.**
Let $\mathrm{g}(d)$ be the minimum amount by which two integers s and t can differ if their difference in serial transition count, $\Delta_{s,t}$, is d. Similarly, let $\hat{\mathrm{g}}(d)$ be the maximum amount by which two integers s and t can differ if their difference in serial transition count, $\Delta_{s,t}$, is d. That is,

$$\mathrm{g}(d)=\min_{\Delta_{s,t}=d}\{|s-t|\},\quad \text{and}\quad \hat{\mathrm{g}}(d)=\max_{\Delta_{s,t}=d}\{|s-t|\}. \ ■$$

Figure 5(a) illustrates several properties of $\mathrm{g}(d)$ and $\hat{\mathrm{g}}(d)$, and Figure 5(b) shows an empirical exact enumeration of $\mathrm{g}(d)$ and $\hat{\mathrm{g}}(d)$ for unsigned 8-bit values. When there is no difference in serial transition count ($d=0$ in the figures) the original and encoded values may be identical ($\mathrm{g}(0)=0$) or may be 0 and 2^l-1 or vice versa ($\hat{\mathrm{g}}(0)=2^l-1$). At the maximum possible serial transition count reduction ($d=l-1$), the incurred deviation cannot be reduced below $\frac{1}{3}(2^l-1)$; the worst-case deviation at this maximal-transition-reduction point is however also limited, at $\frac{2}{3}(2^l-1)$.

V. Summary and Discussion

Value-deviation-bounded approximate serial data encoders (VDBS data encoders) reduce the power for transmission of a word over a serial communication interface. They achieve this by reducing signal level transitions in the transmitted word at the expense of numeric deviations in the values transmitted. This article provided insight into:

❶ The reduction in serial transitions (and hence dynamic power) that can be achieved at the cost of induced error (Proposition 1, Lemma 1).
❷ The relation between the *count of serial transitions within a single word*, and Gray codes, which minimize *transitions between consecutive words* (Theorem 1).
❸ Definition of the bound on transition reduction that can be achieved for a given value deviation (Definition 3). The relation between transition reduction and value deviation is not monotone, and Theorem 3 provides the tightest linear monotone bound.
❹ Definition of bounds on the maximum and minimum numeric value deviation that any VDBS data encoder will induce for a given reduction in serial transition counts (Definition 4).

The properties, proofs, and bounds are important regardless of whether actual practical VDBS data encoders are implemented in software or in hardware.

References

[1] W.-C. Cheng and M. Pedram. Memory bus encoding for low power: a tutorial. ISQED '01, pages 199–204, 2001.
[2] M. R. Stan and W. P. Burleson. Bus-invert coding for low-power i/o. *IEEE TVLSI*, 3(1):49–58, Mar. 1995.
[3] P. Stanley-Marbell and M. Rinard. Value-deviation-bounded serial data encoding for energy-efficient approximate communication. Technical Report MIT-CSAIL-TR-2015-022, MIT Computer Science and Artificial Intelligence Laboratory (CSAIL), June 2015.
[4] H. S. Wilf and A. Nijenhuis. *Combinatorial algorithms: an update*. SIAM, 1989.